SECURITY NOW!

Transcript of Episode #668

Lazy FP State Restore

Description: This week we examine a rather "mega" patch Tuesday, a nifty hack of Win10's Cortana, Microsoft's official "when do we patch" guidelines, the continuing tweaking of web browser behavior for our sanity, a widespread Windows 10 rootkit, the resurgence of the Satori IoT botnet, clipboard monitoring malware, a forthcoming change in Chrome's extensions policy, hacking apparent download counts on the Android store, some miscellany, an update on the status of Spectre & Meltdown - and, yes, yet another brand new speculative execution vulnerability our OSes will be needing to patch against.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-668.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-668-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Lots to talk about this week, including Microsoft's Mega Patch Tuesday, a clipboard attack on cryptocurrency, and Steve has suffered a sad loss. No, it's not that bad. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 668, recorded Tuesday, June 19th, 2018: Lazy FPU State Restore.

It's time for Security Now!, the show where we cover your security, privacy, and safety online. Oh. I didn't press the record button.

Steve Gibson: Oh, I hate when that happens.

Leo: I hate it when that happens. Well, this recording is actually just for the reruns. But it's time for Security Now!, the show where we cover your privacy and security online with this cat right here, Mr. Steve Gibson of GRC Corporation. Hello, Steve.

Steve: Yo, Leo.

Leo: Welcome to the show.

Steve: Great to be with you again as we cross past the middle of June.

Leo: Yeah. How did that happen?

Steve: I don't know. It's, like, getting to be summertime already. So, well, early in January, when we first discussed speculative execution attacks, or the vulnerabilities, rather, that had come to light, and everyone ran around with their hair on fire, it seemed likely, and I said at the time on the podcast that I didn't think we had seen the end of this, that it was going to be, like, a real problem because so much work had gone into the engineering of shortcuts and, in some cases, surprising architectural complexity for the sake of squeezing every last possible bit of performance out of our systems. As we know, a couple weeks ago we talked about SpectreNG, also known as Variant 4 of the Spectre problems. Well, this week we have another one. This is the Lazy FP, or in some cases FPU, state restore. And patches will be flying.

Leo: Uh-oh.

Steve: Oh, yeah. So we're going to wrap up - sort of I want to do, as a consequence of last Tuesday's Patch Tuesday, take stock of where we are with what patches are available for which operating systems and which flavors and whether the mitigations are enabled and disabled by default. I've pulled all that together. And then also take a look at the latest gift from some German security researchers, who realized that another feature of the Intel core processors, so this only affects Intel core processor families, I think I read from - what was the bridge? Shady Bridge? That sounds wrong. Sandy Bridge. Sandy Bridge forward. No AMD problems. This one doesn't affect the ARM architectures and so forth. This is only Intel. But it's another way of leaking information.

But before that, we need to take a look now at last week's, what I would consider a Mega Patch Tuesday and what it means. We do the podcast as this is all happening, so it's just sort of with the timing of when the updates are happening and what's been released, there's just no chance to cover it. And normally I just sort of let it go because our listeners know it's important to patch. This time, well, let's see. Eleven critical vulnerabilities and 50, five oh, more than 50 updates. And several of them are really horrifying. So I think we're going to end up seeing some downstream consequences of what Microsoft has just fixed.

There was also, in the security news, a very clever and nifty hack of Cortana in Windows 10 from security researchers at McAfee. Microsoft put out an official "when we patch," or "when do we patch" guidelines summary which is sort of interesting because there's a little more discipline, or so it seems, to their decisions than was obvious before. We've also got some continuing tweakage of web browser behaviors to increase, and I guess maybe help, our sanity. A worrisomely widespread Windows 10 rootkit. We haven't talked about rootkits for a long time. They're really interesting things, of course. And the whole reason that Secure Boot exists is to prevent them. But it doesn't.

We also have the resurgence of the Satori IoT botnet with sort of a vengeance; some clipboard monitoring malware; a forthcoming change in Chrome's extensions policy; kind of a clever hack for apps in the Android store; a little bit of miscellany; and then, as I said, we'll wrap up talking about where we stand today with Spectre and Meltdown mitigations for Windows, and then what is this Lazy FPU State Restore. So it's been a busy week.

Leo: All right, Steve. I've got the Picture of the Week all queued up.

Steve: This is a particular poignant one for me, Leo.

Leo: Aw.

Steve: Now, of course, for those who don't have video, this is a picture of the famous 14-pin DIP.

Leo: I've been called worse.

Steve: DIP stands for Dual Inline Pins, or sometimes Dual Inline Plug.

Leo: This looks like the memory chips that I used to have to put in my motherboard.

Steve: Oh, they were ubiquitous. In fact, they were typically the 7400 series. The 7400 was a quad NAND. So you'd have 14 pins. Pin 14 is 5V VCC. Pin 7 on the exact opposite is ground. That left you 12 pins. And so a 2-input NAND gate needs three. So you'd have four 3-input NAND gates in one of these little 7400 chips.

Leo: So that was, what, 8K of RAM or 2K or 4K?

Steve: Oh, no. Well, no. In fact, probably what you're thinking of was the original Intel Dynamic RAM? That was 1K.

Leo: 1K.

Steve: They were, yeah, you needed a bunch of those to get anywhere.

Leo: But the motherboard had little sockets for these pins, and you'd just drop them in.

Steve: Yeah, yeah.

Leo: But this is solid-state memory, then. This is NAND memory.

Steve: Correct. Well, no. Oh, no, no. What I was talking about was a NAND gate, a Not AND. So it ANDed the two inputs, and then inverted the output.

Leo: So this is, what, a processor?

Steve: No, no, no. It's just...

Leo: It just does that? That's all it does?

Steve: That's all it did. That was how low-density...

Leo: What good is that? One NAND gate? That's it?

Steve: You do need a lot of them in order to get any work done. But I've had the picture sitting in my Security Now! pictures archive just because, first of all, the quote is famous Dr. McCoy, where all throughout Star Trek one of the recurring memes was Kirk would tell Bones to go fix some alien or some Horta that was tunneling through rock or something. And in some cases Bones was unable to revive this entity. And so he would say, you know, "It's dead, Jim," meaning that's it. Well, so is my Windows XP machine.

[Clip] BONES: It's dead, Jim.

Leo: What? Your Windows XP had that in it?

Steve: Well, no. Well, probably, actually. Probably old enough. But all of our listeners know that I have rather infamously been sticking with Windows XP because it works.

[Clip] BONES: Its brain is gone.

Steve: Because it works just fine. And I don't know when I set it up, but probably in 2003, when it would have happened. So it's 15 years old. And I had...

[Clip] BONES: It's medically impossible.

Leo: Sorry, I keep playing Bones.

Steve: I had VPNed into it Sunday evening in order to get something done.

Leo: To do what? You were working on that thing still?

Steve: It's my main workstation. It's everything. SQRL has been written there. All the podcasts have been produced on it.

Leo: No.

Steve: It's been running for 15 years, yeah. That's my main workstation. And Chrome began to complain that it didn't like the version of Windows I was still using, and Firefox began to complain. So it was becoming a little long in the tooth. Well, I got back to my office Monday morning, yesterday morning, and went into my little workstation area, and it was quiet.

Leo: Uh-oh.

Steve: And the machine was off. And in 15 years it had hung from time to time, so you'd just reset it, and up it would come. And it was getting kind of finicky where it wouldn't restart immediately. I had to let it cool off for about 30 minutes if I needed to do a reboot because it thought it was getting overheated. And so it would say, no, no, no, I can't boot. So it was giving me plenty of warning. But every day it just - it was there, and it kept working. And so I just kept using it. Until yesterday morning, when I was unable to bring it back to life. I thought maybe the power supply had died. So of course I had another system of similar age. Actually, it's the one that my bitcoins might be on. I had to dig that machine out from underneath the rubble.

Leo: Oh, that's good. So now you'll finally get that back.

Steve: I probably, well, I will at least have a chance to figure out whether or not I do in fact - I think what I did was I moved the bitcoin wallet and miner over to that other machine, planning to sort of just like, oh, see what else can happen, since I got 50 bitcoins overnight without trying. But I don't think I really - I don't know what happened. I got bored or didn't, I mean, they weren't anything back then. Anyway, so swapped the power supply, no change. So I think the motherboard just bit the dust.

So anyway, our listeners will remember when Microsoft had announced that they were not planning on supporting Windows 7 on newer hardware, and that was a couple years ago. And I said, what? Okay, wait, stop. So I immediately purchased the last motherboard family that they would promise to support Windows 7 on because nothing was going to make me go to 8 or 10. So I built that beautiful system, remember, 128GB of RAM and a state-of-the-art, superfast, direct-on-the-PCI-bus SSD, and a pair of multi-terabyte mirrored hard drives. I mean, it's a monster. Oh, and an 8-core processor.

Leo: This was the last computer you were ever going to build, I think; right?

Steve: Yes. That is. And it has sat patiently on the sidelines because...

Leo: We thought you were using that.

Steve: There's never a good time.

Leo: There's been no reason, yeah.

Steve: Yeah. There hasn't been a reason. And right now, I mean, now I'm facing it. I have nothing. I can't edit the audio, I mean, I had to, like, quickly put - normally I have a nice little stopwatch, like a timer, that shows me how far into the podcast we are so that I know it's time to take a break. I don't remember the name of it. So anyway, I didn't lose any data. I have multiple images and snapshots and things, although the very, very most recent stuff is on the RAID on the dead machine. So in the next day or two I will rehost that RAID on a different motherboard, bring the file system back up. Oh, it just - the system just blanked on me. See, for example, that. So anyway, I'm in the process of - oh, and my development environment had lots of 16-bit code. Brief, that I've still been using. I mean, I wrote - SpinRite 1 was written in Brief. Actually, FlickerFree before it was written in Brief.

Leo: It's really time. It really is time.

Steve: You know, in 1986 or something.

Leo: It's just really time.

Steve: So, yeah. And I was explaining to Lorrie last night that, well, you know, there is never a good time. I have not wanted to have any time off or out or pause because I'm always in a hurry trying to get as much done as I can. So it's like, okay, well, it works. So if it's not broke - anyway, so and finally it is broke. So this picture of the chip on its back with its legs up in the air, I just thought, okay, it's the perfect time...

Leo: It's appropriate.

Steve: ...for the "It's dead, Jim."

Leo: Do you need anything on that hard drive?

Steve: Yeah, everything.

Leo: Oh. Good luck recovering it, if it's a RAID. Did it have a dedicated RAID card?

Steve: Of course, yes. So I just...

Leo: He's no fool, ladies and gentlemen.

Steve: I moved it and its four drives over to another chassis. And I will then - I'll probably move everything, just camp it on the Drobo and then begin to pull things back from there.

Leo: Yeah. But you're going to use a more modern version of Windows; right?

Steve: I'm on 7.

Leo: Good, okay.

Steve: Yes, I'm now on 7. All updated, oh, like an 859MB update because of course it hadn't had any updates for two and a half years. So it's all current now. And I'm glad to be here.

Leo: I think you'll be faster.

Steve: Oh, boy, is it fast. Oh, you know, like I'm just not used to it.

Leo: Steve, it's normal. You've just been on a slow machine.

Steve: I used to think, boy, these pages have gotten so big since the Stone Age. They just take forever to load. And so I have, like, 300, what is it, megabits, yeah, 300Mb of download from Cox.

Leo: Oh, that's nice.

Steve: And it's great. Well, but then it hits the brick wall of XP that's got a stack from TCP-1. And so it says, what? And it just sort of - it's the large pipe hits the little straw. Anyway, oh, I'm having a great time. It's like, whoa, wait. That's downloaded already? Oh, okay.

Leo: Nice.

Steve: Yeah. Anyway, so I announced to the SQRL newsgroup that I was limping along at the moment. It'll take a few days for me to pull my - and, I mean, I have to rebuild a development environment because my whole tool chain was 16-bit. Everything, you know...

Leo: Steve. Steve. You're using MASM? Is there a 32-bit MASM?

Steve: Oh, I guess there must be somewhere.

Leo: Maybe.

Steve: Yeah, probably, yeah.

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

Steve: Actually, there's a 64 because I did for - what was it I did? Oh, for InSpectre I had to...

Leo: That's right, you wrote 64-bit, yeah.

Steve: ...create a little bit of 64-bit code for InSpectre. So I did my first 64-bit MASM for that.

Leo: You didn't do that on the XP machine. You probably did that on the new one; right?

Steve: Actually, I did that on my Lenovo laptop. I did that. That was a little project that I had in the evening. So, I mean, like everything else is Windows 7, and I have a bunch of Windows 10 machines because I have to make sure that SQRL works on them properly and everything. So, I mean, it's not like I'm stuck. It's just that - and Leo, over 15 years you just, like, there's so much tuning and tweaking and things done to a system. Like I just keep thinking of things. Ooh, I don't have that. Ooh, I don't have that. Ooh, you know.

Leo: Oh, yeah, I bet, yeah.

Steve: So anyway.

Leo: And a lot of those things won't run, I'm sure, in a modern machine.

Steve: No, no. In fact, it was Mark Thompson who made the jump before I did. And he made the comment, he said, "I didn't realize how much 16-bit code I was still using until things I used to do no longer worked any longer." Like my Gravity newsreader. I can't run the one I had on 32 bits because it was 16 bits. And, I mean, I do have the, what was it, there is a WinXP VM that you can run under Windows 7 if you absolutely must run some 16-bit code. But I'm not doing that. It's time. I mean, I agree with everybody it's time.

Leo: I had no idea, I mean, I guess I kind of knew, but I didn't realize that was your main machine. I had no idea.

Steve: Yeah, that's what I sit in front of every day. So I had to go out, and I got a bunch of display port cables because all of my monitors were DVI, and the new machine is an 8-head Matrox, two Quad Matrox cards, but they're all display port. So I cabled up all the existing monitors, and now I'm sitting in front of an array of monitors, but there's nothing on them.

Leo: Welcome to 2010, Steve. You're going to love this new decade.

Steve: And I've got no icons. I used to have all these icons. I just, like, went right off the screen. And now, like, I've got nothing. But anyway, yeah. So anyway, Microsoft's June Mega Patch Tuesday, it's the only thing we could call it. More than 50 vulnerabilities were patched affecting Windows, IE and Edge, MS Office, Exchange Server, the ChakraCore which is the core of the JavaScript engine in Edge, and even Adobe Flash Player. Microsoft said, oh, we're going to push that out, too. Among those 50 vulnerabilities, 11 were critical and 39 were important. And for a change there were no surprises. There were no Windows zero-day fixes this month.

What we had hoped would happen, I guess I was talking about it Tuesday. Maybe I was talking about it the Tuesday before because I think I would have checked. Maybe it was happening. And that is, remember that there was a very bad JavaScript flaw which

Microsoft dropped the ball on. It was discovered by Dmitri Kaslov of Telspace Systems, who responsibly disclosed it to Trend Micro's Zero Day Initiative because it was a zero day. He discovered it being exploited. That's how he found it. And it was a problem with the JavaScript engine.

And Trend Micro reported it to Microsoft in January, who let 90 days go by. They immediately acknowledged the receipt of the news, then let 90 days go by before they worried about it again. And something, maybe a reminder in their calendar said, you know, Trend Micro's going to go public with this any minute now. So they said, oh, we were unable to reproduce it without a proof of concept. And Trend Micro responded with, well, here's the proof of concept we already sent you 90 days ago. And then Microsoft asked for more time, and Trend said no. So Microsoft said, okay, well, we'll get this fixed as soon as we can.

So it is now fixed as of last Tuesday, so that's good. That was a zero day which Microsoft just didn't respond to in time. Trend did everything right, and I would argue also made the right call because this was already being exploited in the wild. So it's not like it was unknown. Dmitri found it in use. And without pressure on Microsoft, they just clearly aren't demonstrating a great need to fix things. So this was probably the right thing to do.

They did also fix that Adobe Flash zero-day which we talked about last week. That was the one where, because browsers are really locking down Flash now, even though Flash refuses to die until 2020, some clever malicious hackers figured out how to get Office to invoke the zero-day in order to get remote code execution capability. So Microsoft said, okay, we're going to take responsibility for that since it's leveraging our infrastructure. And they pushed that out, and Adobe had also updated their player.

Okay. But there are two things that were fixed Tuesday that should give us some pause. As we know, the worst vulnerabilities are those which are remotely executable, where no action is required from the user, which are fundamentally wormable and exist in a large number of systems. There are two that were fixed last week. The first one has the CVE number 2018-8225. That was a flaw, believe it or not, in the dnsapi.dll, which has affected all versions of Windows from 7 to current. And I don't know that it didn't affect XP because Microsoft doesn't talk about anything before because it's like, okay, those don't exist anymore. Maybe it's always been there.

But it turns out - oh, and this includes the Server editions of Windows. So all versions of Windows that Microsoft is talking about, 7 through 10, there has been a flaw, only just patched last week, in the way Windows was parsing DNS responses, which allowed it to be exploited by sending a specially malformed DNS response that would allow attacker-controlled malicious code to be run in the context of the local system account, which is the highest privileged account in the system, essentially in the kernel.

So what that says is that, until last week, and still for any systems including Windows servers that have not been patched with last week's updates, the receipt of a malicious DNS reply takes the system over. And as we know, DNS uses UDP, so it's trivially spoofable, and it's nonencrypted. So this has been hanging out there since Windows 7, at least, and until last week. So if you haven't updated your Windows systems yet, and hopefully everybody has, don't wait any longer.

And if you have any responsibility for servers, you know, most Windows machines today are behind a router. The problem is this doesn't even protect them because a bad guy monitoring outgoing DNS queries simply drops a UDP packet on the line heading back in, and that will cut right through the NAT traversal firewall and the Windows firewall to get into the machine. Servers are - they feel more vulnerable because they're deliberately exposed. But they're probably not going to care about an unsolicited DNS reply. They are going to care about one that looks like it's answering a question that they just asked. And servers are always performing DNS queries to a much greater degree. I mean, so are end-user machines.

But anyway, this is bad. So if you're involved in a company not directly responsible for IT, do make sure that the IT people responsible for any Windows servers, any Windows machines, really, that are in the network, got themselves updated. What will happen is, as we know, now everybody knows that a change was made to the dnsapi.dll to fix this very worrisome problem last week. The change will be reverse-engineered. The bad guys will figure out what it was that got changed and how to leverage this. And, I mean, it's low-hanging fruit to be able to send a DNS query into a system and take it over. So this is bad.

The other problem is not much less bad. It's CVE 2018-8231, which is another remote code execution flaw in another publicly exposed network component. In Windows, some time ago the various pieces got broken up. There's a kernel driver called http.sys which is the HTTP protocol stack that parses all incoming HTTP stuff. So many non-servers, non-Windows servers do run a service or a server. In some cases HTTP gets invoked. All Windows servers do.

So the good news here is that this problem only affects - well, only - Windows 10 and Windows Server 2016. So people using earlier versions of Windows, this one doesn't affect them, although any Windows Server 2016 there's another very worrisome, I would argue horrifying, remote code execution flaw in this component of Windows Server that is, again, directly exposed to the Internet. So I'm speechless. Last week's security update, which just sort of slipped under the radar, as I do, I went back and just sort of looked over it. And you normally don't hear me talking about these every Tuesday, or every second Tuesday of the month, because it's like, okay, fine. So a bunch of things got fixed. In this case they are worrisome.

And I guess I have to back off on saying it's wormable because at least in the DNS case I'm sure - I'm hoping - that you can't send a blind reply to a machine and have its dnsapi.dll cause a hacker-controlled remote execution. I'm sort of assuming you need to send a response to an outstanding request. But if not, then it's worse. So anyway, yeah, if you're responsible for any Windows machines, really last week's June Patch Tuesday was a biggie.

And everything else, you know, there were seven other critical memory corruption bugs, less horrible than those two. Both the ChakraCore and the Chakra scripting engine that drives the Core, they both had them. There were three in the Edge browser. There was another in Windows Media Foundation. And all of those could lead to remote code execution. So it was a busy month, and maybe that explains why Microsoft didn't fix the outstanding zero-day from Trend. Who knows?

I'm trying to think where they were - oh, it was McAfee. I mentioned that at the top of the show. Cedric Cochin in McAfee's Advanced Threat Research team came up with a clever hack. And we've seen things like this, Leo, with voice assistants before, and I'm sure that there have been some for iOS, where features are added without fully vetting all of the possible ways that they can be abused. Turns out that in Windows 10 Cortana can be abused. Until last Tuesday. This is another of the things that Microsoft locked down.

So this one is not a remote problem because it involves Cortana, but a local one. It turns out that, if you activate Cortana, and she comes alive, on a locked machine - so the machine is powered on, but locked so that it's not supposed to do anything. Turns out Cortana is still listening. So anyone can come along and activate Cortana. It turns out that you can either phonetically or use the keyboard to start typing. You can type while Cortana is listening for more verbal command.

And these guys came up with a multistage means, to make a long story short, of running an externally provided power shell script from a USB drive that's plugged into the machine. That ends up getting registered with a system. Cortana is able to see it. And when you ask for a search, its contents are indexed, which allows you to bring up a file which in this case is a power shell script which is the first stage of the attack. It's then possible to get that to execute code, your own hacker-provided code, from the USB device, and go as far as, for example, completely taking over the locked session, and even changing the system's password on a locked machine. The blog posting that McAfee put up was titled, "Want to break into a locked Windows 10 device? Ask Cortana."

So that's yet another thing that last Tuesday's patch round fixed, fortunately. And they noted that, because they felt that this was - now there's full details published. It takes no advanced hacking ability in order to abuse this. So they were suggesting, and I would absolutely concur, if you do not update, for whatever reason you have not updated to last Tuesday's patches that fixed this, by all means disable Cortana when the system is locked. And I would argue that's just sound security practice. There are just too many opportunities for this kind of exploit on a system which should just be shut down and not listening to anybody while it's locked.

So I would say, even after having patched the June updates, unless you have a compelling need to have Cortana listening while your system is locked, I think it makes a lot of sense to disable that. And had you done so already, this would have never been a vulnerability. I expect there will be some people getting up to some antics with this one because it's so easy to do. And if anyone is interested, I have a link in the show notes. They have a super-detailed, blow-by-blow, how do you pull this off explanation in their [crosstalk].

Leo: You'd have to have physical access to the machine, of course.

Steve: Yes. Yes, yes, yes, you absolutely have to have physical access. So it's not nearly as worrisome as the remote code execution problems that we talked about first.

And Microsoft also published a written policy which we've never encountered before on when things get patched versus when they feel comfortable deferring them to the next release of whatever it is they're doing. They called it, yeah, their Microsoft Security Servicing Commitments. And it's interesting enough that I'll share the beginning of this.

They wrote: "Our commitment to protecting customers from vulnerabilities in our products, services, and devices includes providing security updates that address these vulnerabilities when they're discovered." Okay. We wish that was true. That's generally true, although sometimes it takes many months. They said: "We also want to ensure we are transparent with our customers in our approach. This document helps to describe the criteria the Microsoft Security Response Center" - we've talked about it often, the MSRC - "uses to determine whether a reported vulnerability will be addressed through servicing, or in the next version of the product. For vulnerabilities in products, this 'servicing' takes the form of a security update, most commonly released as security updates on Update Tuesday." That's what they call it, "Update Tuesday." "The purpose of this document is to clarify the commitments as they pertain to Windows."

So they break down their security servicing criteria into two categories. Well, it's sort of a bit of a tree, sort of a decision tree. They said: "The criteria used by Microsoft when evaluating whether or not to provide a security update for a reported vulnerability involves answering two key questions. First, does the vulnerability violate a promise made by a security boundary or a security feature that Microsoft has committed to defend? And the second, does the severity of the vulnerability meet the bar for servicing?"

Okay. So security boundaries are also something that they then rigorously define. And, roughly, those are the things that we talk about all the time. For example, you could have a network boundary where you want to protect anything from having access to network traffic; a kernel boundary where application code is a pure client of the kernel and is unable to cross that boundary, unable to penetrate it. And in fact all of the Spectre and Meltdown problems, those are clearly security boundary violations, which is why the whole industry went crazy at the beginning of the year.

Process boundaries. Also processes are supposed to be contained. We can sort of think of these all as sandboxes, where the sandbox boundary is - the whole reason you have a sandbox of one form or another, network or OS or process, is for containment. Even the logon session boundary, Microsoft specifically says, if you've got two users logged onto the same system, they have a boundary that is dividing them and protecting them, and you have to preserve that commitment. Web sessions have boundaries, both the browsers' sandbox and also the same-origin policy that rigorously enforces a boundary about where the code came from and not allowing code from one origin to have anything, any visibility into resources from a different origin. And also finally their VSM, the Virtual Secure Mode is another boundary.

So they explain that all of those things, those security boundaries, receive Microsoft's, as they called it, a "servicing commitment," and warrant a bug bounty. So first of all, if you find something that is able to cross one of these security commitment promises from Microsoft, they take it seriously because it's a breach of the commitment they've made, and you can get remunerated for providing the information to them.

They then also have security features which they consider to be distinct from boundaries. And they define them: "A security feature provides protection against one or more threats. In some cases, a security feature may make a promise related to the threat they're protecting against, and there are not expected to be any design limitations that prohibit delivering on that promise," meaning it's a feature that's supposed to work under all circumstances.

And then they give some examples. BitLocker and Secure Boot are not boundaries. Those are features. Windows Defender is a platform security feature, and the Application Guard in Defender is an application security feature. Identity and access control - so like logging on, biometrics, and Windows management over what resources an individual has access to - those, again, are not boundaries. They consider those security features. And they also put the Cryptographic API as a feature, along with authentication protocols. So those are sort of services within Windows which they also treat as, like, take them seriously services. They call those commitments, and those also receive bug bounties.

The things that don't, sort of the second-class citizens, are what Microsoft calls "Defense-in-Depth" features. So those are features like UAC, AppLocker, Controlled Folder Access, and even Data Execution Prevention, Address Space Layout Randomization, Kernel Address Space Layout Randomization, DLL signing protection. Those are things that they consider important, but the breach of those does not cross a boundary. So some of those warrant, in Microsoft's opinion, bug bounties, and some don't. And so essentially their position is that, if the defense-in-depth things have a critical or important vulnerability level, they will be fixed in an upcoming Tuesday.

All of the boundary breaches and all of the security features - BitLocker, Windows Defender and so forth - all of those things that also warrant bug bounties, those also, if there's a breach in those, that gets their attention. That gets fixed as soon as possible. So essentially where they feel they've made a firm commitment in the architecture of Windows such that something should not be possible, but it's not about more security, it's more about absolute functional security, that gets fixed.

But apparently there is a large class of things that people report that are, well, you know, this isn't working the way it should, or we were able to get around it, but for example it doesn't rate critical or important, then Microsoft is taking the liberty here, and they've now spelled it out explicitly, of, okay, we should fix that, but we're not going to run around worrying about pushing it out immediately. When we do a next version of whatever it is, we'll probably fix it then.

So it was nice to have a formal enumeration of what their policy was. And also, I think, people were wondering what is subject to a bug bounty and what is not. And most of these things are. Microsoft certainly wants to encourage people to submit problems which are discovered. So anyway, I was glad to see that they laid it all out in detail.

Leo: Very useful. Okay, Steverino.

Steve: So Leo, you will be glad to know, I know you will be glad because I heard you talking about this elsewhere, that Microsoft has decided to join the group of browsers that will not be playing audio and video...

Leo: Well, hallelujah,

Steve: ...by default. Yes.

Leo: Everybody else does this already, though; right? They're the last one?

Steve: Yes. Well, Chrome actually was first. They've been blocking autoplay since Chrome 66. Mozilla has formally stated their intention of adding it to Firefox, but they're just saying sometime later this year. So they're definitely going to do it. Microsoft has stated that Edge would have this, and in fact it was expected to be in the most recent insider build, 17692. But apparently, in fact, some people thought it was in there and were surprised when it wasn't. So it just barely missed whatever closing deadline the insider builds have.

However, it will not be enabled by default. So under Edge Advanced Settings there will soon be "Allow sites to automatically play media." Users who, like, probably everybody, like who wants that, should just set that to Off. Microsoft is going to - so they'll give us the option. I don't know if maybe someday they'll flip it the other way so that it's enabled, that is, well, so that autoplay is disabled by default. They're taking it slowly. But at least the option will be there.

So for people who are not using Chrome, who are using Edge and who say, wait a minute, why am I getting these videos playing all the time, soon you'll be able to just turn that off. And I don't know about IE. There was no mention of that. I'm assuming they're just going to leave it alone and sort of use it as one more reason to push people away from IE and over to Chrome. Over to, well, yeah, Chrome. Over to Edge. So anyway, that's good.

Bitdefender Labs has been following for, oh, I think about five years now a very pernicious piece of adware which has, more recently, very sophisticated Windows 10-based rootkit behavior. We talked about rootkits, boy, a long time ago. And you'll remember this, Leo, because, I mean, it was a series of famous episodes for us. There was something from Sony. Was it a game? I don't remember what it was. It was some software which for - oh, I think it was maybe DRM. It might have been a DRM rootkit. Essentially a rootkit is something...

Leo: Oh, yeah, yeah, yeah. Installed a rootkit, yeah, yeah, yeah. It was [crosstalk], yeah.

Steve: Yeah, I think it was digital rights management was like their justification. And so what a rootkit does is - it's hard to think...

Leo: Oh, no, it was on music.

Steve: Oh, my god, you're right.

Leo: It was on CDs.

Steve: It was just Sony audio CD. Oh, wow.

Leo: BMG, Sony BMG distributed a copy protection scheme with music CDs that secretly installed a rootkit on computers. Geez Louise.

Steve: I know. Can you believe it? We've come a long way. So for those who don't know, who have not been listening forever to the podcast, it's sort of counterintuitive. But a rootkit is software which hides itself. And the way it hides itself is by modifying what the operating system shows when you do, like, a directory listing. And we're used to looking in Windows Explorer, or if you brought up a command prompt and you typed "dir," you're going to see - that's a directory command. You're going to see the files there.

Well, it's the operating system itself which is reading the file system and putting up in Windows Explorer the tree of files, or enumerating them, listing them on a command window. So if the operating system itself is subverted, then there can be files there that don't appear. And you can't unhide them. You can't show system files, show hidden files. They don't exist, like from any external view. The only way you could see them would be to boot like an external OS and have it look at the drive because then the drive essentially wouldn't be able to hide things from the person viewing.

Anyway, so as a consequence, rootkits have always made people very uncomfortable because they're hiding in plain sight. They're invisible. And so the way this is done is that, when you're doing a directory listing of something, when you're displaying the file contents, you're making calls to the operating system, asking it to return a list of things. Well, if the return runs through some malware, if there's something intercepting essentially a man-in-the-middle attack is what this is, a man-in-the-middle attack on the operating system API itself, then it's able to remove from that returning list of files references to anything it doesn't want to have appear. Things just vanish, and you can't see them.

So anyway, the idea that there is a Windows 10-based adware rootkit which does this is a little unsettling, especially because it's affecting the most recent version of Windows, which is supposed to have very proactive anti-rootkit technology. The whole Secure Boot system that we've talked about and detailed over several past podcasts is all about maintaining a firm chain of control, essentially a chain of trust, from the moment the motherboard powers up, and the BIOS is signed and protected. And through a series of modules which are loaded as the operating system boots, each one is verified for its not having been tampered with and having been published by a trusted source. So the idea that this has been and can be subverted is troubling.

BitDefender Labs has published a very detailed 104-page PDF whitepaper that I'm not going to go into in great detail. They called this Zacinlo, Z-A-C-I-N-L-O. Zacinlo is the name of this adware-based rootkit. They've taken a look at the population of exposure: 89% of the infections are Windows 10; 5.3% are Windows 7; and the rest is made up of 8 and 8.1. What's also interesting is that, unlike most of these or many of these things that we see, where it's like, okay, well, most of the infections are in Asia or over in Europe somewhere, or maybe in the Middle East, almost all of the infected systems are in the United States. So this is adware clearly focused upon and targeting U.S. users.

In the beginning of this long whitepaper they say: "Last year we came across a digitally signed rootkit capable of installing itself on most Windows operating systems, including the newest releases of Windows 10. Since rootkits," they write, "these days account for under 1% of the malware output we see worldwide, this immediately drew our attention and prompted us to carry out an extensive analysis of the payload, its origins, and its spread. We discovered an ample operation whose central component is a very sophisticated piece of adware with multiple functions."

And as I said, they have been tracking this for about five years, although it has recently evolved into this Windows 10-capable rootkit technology. So they've identified 25 different components and found almost 2,500 distinct samples. So this is making people money somewhere and doing everything it can to stay off of the radar, unlike very loud botnets like Satori that are attacking people and commandeering crypto mining and so forth. This just wants to operate quietly in the background. It has a rootkit driver.

And I did, I should mention, I looked through this extensive whitepaper. And I'm very impressed, unfortunately, with the technology that has been brought to bear. This thing has received a huge amount of engineering work in order to create something that burrows into people's Windows 10 machines and then hides and makes itself undetectable. They run through some of the things that it's able to do.

They said: "The presence of a rootkit driver that protects itself as well as its other components is the main feature. It can stop processes deemed dangerous to the functionality of its own adware, while also protecting the adware from being stopped or deleted." And of course that's what a rootkit is about. It hides itself, and then it makes sure that - and also defends itself. "The presence of man-in-the-browser capabilities which intercepts and decrypts SSL communications."

So this thing is buried so deeply that it is able to even intercept encrypted HTTPS web pages, which allows its adware component to inject custom JavaScript code into the web pages being visited by the user. And since the injection can appear to be first-party, we're sort of lucky that this thing is only wanting to be adware and not be more malicious. I mean, this is a very potent piece of malware.

It features an adware cleanup routine used to remove potential competition in the adware space. So apparently there are some other companies that they don't like, and they get rid of them as part of their coming in and taking over the system. They describe that portion as "generic and does not target a particular family or type of adware." It can uninstall and delete services based on instructions it receives from the command-and-control infrastructure. So it is also, in addition to adware, it is a botnet with a mature command-and-control infrastructure. It's able to report some information about the environment it's running in back to central command; whether antimalware solution is installed and, if so, which one; and which applications are running at startup.

So, I mean, it's also spyware, spying on the machines that it has infected. It takes screen captures of a desktop and sends them to the command-and-control center for analysis. And BitDefender notes that this functionality has a massive impact on privacy because of course, as we know, screen captures may contain sensitive information - email, instant messaging, or ebanking sessions. I guess it's one reason for blanking the password while you're typing it in. I've never, as we've talked about here, I've always felt that not displaying the password while you enter it causes more trouble than it's worth. But with the one exception being, if something is sitting around capturing, like snapshotting your screen, you'd rather have black dots showing than what it is you're typing.

It says it can accommodate the installation of virtually any piece of software on the fly to extend its functionality. So it is a full trojan with rootkit technology, and it has an automatic update mechanism. It's able to redirect pages in browsers. It's able to intercept and replace advertisements on the fly with those of its own. It supports many different platforms from which to pull advertising for revenue-generating, including Google AdSense.

So essentially it's creating its own advertising network. You go to websites that hope to be ad supported. You assume they are. But the ads you're seeing are coming from a rootkit in your system generating revenue for these cretins instead. It's able to rotate and obsolete and expire ads. It's also able to run a web page in the background in hidden windows and interact with them just as a normal user would - scrolling, clicking, and providing keyboard input. So what it's doing is it's also perpetrating advertising fraud. So it's pulling up pages, replacing the ads with its own, and generating clicks in order to simulate the user doing that, even if they're not, all in the background, and all completely invisible. It uses open source projects and libraries - chromium, cryptopop, jsoncpp, libcef, libcurl, zlib, and so forth - and uses Lua scripts to download several components, most likely as a way to fly under the radar of some antimalware solutions that detect suspicious downloads and block them. So, I mean, it is extremely configurable, highly modular, and hiding.

The good news is it looks like its point of entry is a bit limited. From everything that they have seen, the only place they have found this installing itself, although you want to make sure you're not a victim of this, it presents itself as a free and anonymous VPN service called s5Mark. It is distributed in the installer for this VPN, which actually is just sort of a weak proxy. It sort of is VPN-esque. But since that's not its intention at all, it's intended to install itself as a rootkit and set up a little adware ad fraud shop inside your system, the idea that it provides VPN services is just sort of to get itself into your system once.

Anyway, s5Mark. If that rings a bell, if anyone listening to this podcast has ever thought, oh, I'm just going to use this little free VPN, you need to maybe take some action about seeing if something like this has been installed in your system, and see about getting rid of it. s5Mark is a simple little GUI that says it establishes a VPN. And for whatever reason, it's like all in the U.S. In their 104-page whitepaper they have a complete map of where they have found infections, and it's the continental U.S. and Alaska is all red, and very low infection levels anywhere else. So just be sure you haven't done anything that would allow this thing to crawl into your machine.

Last week our old friend the Satori IoT botnet went into overdrive. A whole bunch of researchers who watch Internet activity started tweeting when they suddenly saw port

8000 scanning activity just explode on the Internet. Well, it turned out that this was the botnet itself scanning for new victims. Proof-of-concept code was published to exploit a buffer overrun vulnerability in the lightweight web server. So it's an HTTPD, which is used in many embedded routers and IoT equipment sold by some Chinese vendors. It's XiongMai, X-I-O-N-G-M-A-I, and it's the uc-httpd which is this little web server built in. Turns out it opens up port 8000. It's a web server. And we've seen, for example, that as know, the default HTTP port is 80. And sometimes because you have to have root privilege or admin privileges to open up ports below 1024, some web servers will open port 8080 as a means of opening a port up in user space if a user wants to host a web server for whatever purpose. In this case, they're using 8000 as theirs.

Well, it turns out that sending a malformed packet to this little HTTPD server, which is listening on port 8000, can take it over. So the people behind the Satori IoT botnet, a few days after the proof-of-concept code went public, had incorporated that into Satori and turned their botnet loose, looking for other victims. And of course we know that there were the GPON routers, the optical network routers, largely located in Brazil. They got quickly taken over by Satori. This spike in port 8000 traffic wasn't long-lasting, apparently because in rather short order they had captured all the available devices that had port 8000 exposed. They then switched to scanning port 80 and 8080, the standard HTTP ports and the alternative, for instances of the D-Link DSL 2750B devices. Turns out similarly a proof of concept had been posted. They incorporated that into their network and went about capturing all of those.

So what we're seeing here with Satori is a botnet which is not going away. There was an attempt to take it down in December. It has survived that. And it is, as I mentioned earlier, unlike this rootkit adware, which is trying to stay under the radar and basically not be seen and just be generating revenue for the people who put a lot of time and effort into its development, Satori is just sort of blatant. It's scanning like crazy, taking over things as they become available. What it looks like is that its developers are actively monitoring the so-called "dark web" and hacker forums, looking for any published proof-of-concept code. It doesn't look like they are interested in developing their own. But the moment some proof-of-concept code appears, they send it off to their existing botnet and tell all their bots to go exploit it and find everybody else who is vulnerable to this.

So, I mean, if you wanted to have a botnet that was going to stay current and be aggressive and essentially actively work to supplant other botnets, this is the way you do it. As soon as something appears that offers an opportunity to access another chunk of exposed devices, you have an existing delivery system which is able to exploit that new discovery and jump in and take over before anybody can compete with you. And that's what we're seeing now on an ongoing basis with Satori.

So I have a feeling we're going to have it for some time to come. And remember it's also been responsible for some very powerful DDoS attacks. So it's not staying hidden. It's hijacking cryptocurrency mining. It's stealing funds from people and launching attacks at will against targets of their choosing.

And speaking of stealing cryptocurrency, there's something we talked about a couple months ago. I just wanted to keep it on our listeners' radar. And that is the malware which watches people's clipboards for the presence of bitcoin addresses, either bitcoin or ethereum. The guys at 360 Total Security found 300,000 machines infected with something that they call "ClipboardWalletHijacker." It's malware which takes up residence in a Windows machine and just sits in the background very quietly, polling the user's clipboard.

If it sees something that looks like a bitcoin address, which is a properly sized number beginning with a 1 or a 3, and depending upon actually the date in the month because it uses several different ones, it assumes that's a bitcoin address which a user has temporarily copied from some payment request, put into their clipboard, getting ready to submit it to their wallet. So on the fly they simply replace that clipboard, that bitcoin or ethereum, with their own. And so when you then paste it into your wallet in order to make a payment, they get the money instead of the intended recipient.

Now, despite being in 300,000 machines - and again, this is not going to be nearly as lucrative as the adware rootkit. A rootkit adware system is going to be generating money to the degree it can on every machine that it infects. This thing is only going to ever generate money if a user is using cryptocurrency, if they're using bitcoin or ethereum, and if while infected they do use the clipboard in order to move a bitcoin or ethereum address from one app to another.

Now, the fact is I don't know how you would do it without using the clipboard. I mean, any Windows user it's just second nature to copy something on the clipboard. But as a consequence of knowing, having reverse-engineered these 300,000 instances of this on machines, the 360 Total Security guys know the wallets to which the payments are being made. So they're able to state that, as of the time of their report, this trojan has successfully hijacked five bitcoin transactions. So it's not making money hand over fist. However, the amount of the latest transaction was 0.069 bitcoins, which was about $500. So it's not nothing, but it's a lot of work to have gone through.

As I said, the problem is, from a standpoint of making money, there has to be a whole bunch of things come into alignment in order for anything to happen. So it's not generating a huge windfall of revenue for these people. I thought that I had some other information in my notes about - oh, I'm sorry, yeah. There was a different address. There have been 46 successful transactions in total. And those were ethereum. And there were also five bitcoin transactions, one of which was $500. So again, it's not making tons of money.

But I just did want to remind people that maybe the thing to do is double check, once you paste - if you're an active user of cryptocurrency and moving money around, when you copy the address of the payment recipient from the request for payment into your wallet, you can easily look at the first handful of digits and see if they're the same. Just verify they've not been changed because presumably you'd like to send payment where you intend to.

Google has announced a change of policy about where Chrome extensions can come from. At the moment, and I imagine that we have seen this as we cruise around the web, third-party websites often offer visitors who are using Chrome the option of installing a Chrome extension to enhance their experience in one way or another, typically on that site. The Chromium blog from last week was titled "Improving Extension Transparency for Users," which boils down to, uh, we're not going to let that happen any longer. Essentially, what Chromium has announced, or the Chromium team in their blog have announced, is that they're going to wind down to zero the ability to install Chrome extensions anywhere else. And it's probably worth quickly running through what they said here.

They said: "We strive to ensure choice and transparency for all Chrome users as they browse the web. Part of this choice is the ability to use the hundreds of thousands of extensions available in the Chrome Web Store to customize the browsing experience in useful and productivity-boosting ways. However," they said, "we continue to receive large volumes of complaints from users about unwanted extensions causing their Chrome experience to change unexpectedly" - yeah, things like bitcoin mining in the background - "and the majority of these complaints are attributed to confusing or deceptive uses of inline installation on websites." They call this "inline installation" when you're somewhere else, and you install an extension from some third-party site into Chrome.

They said: "As we've attempted to address this problem over the past few years, we've learned that the information displayed alongside extensions in the Chrome Web Store plays a critical role in ensuring that users can make informed decisions about whether to install an extension. When installed through the Chrome Web Store, extensions are significantly less likely to be uninstalled or cause user complaints, compared to extensions installed through inline installation. Later this summer, inline installation will be retired on all platforms. Going forward, users will only be able to install extensions from within the Chrome Web Store, where they can view all information about an extension's functionality prior to installing. This change will roll out in three phases."

So they wrote, and this was last - I think it was last Tuesday, exactly a week ago. Oh, yeah, June 12th. So they said: "Starting today, inline installation will be unavailable to all newly published extensions." So game over. Anything new coming online, it just will not work to install it inline. Those extensions will have to be registered with the Chrome Web Store.

They said: "Extensions first published on June 12, 2018 or later that attempt to call the chrome.webstore.install() function will automatically redirect the user to the Chrome Web Store in a new tab to complete the installation. Starting September 12, 2018" - so, what, July, August, September, so 90 days from then, September 12th - "inline installation will be disabled for existing extensions, and users will be automatically redirected to the Chrome Web Store to complete the installation."

So essentially, publishers of existing extensions have 90 days from the date of this policy publication last week to decide, to know that they're not going to be able to do them inline, and they're going to be redirected to the web store. And, they said, in early December 2018, the inline install API method will be removed from Chrome 71. So it goes away altogether.

They finish by saying: "If you distribute an extension using inline installation, you will need to update 'Install' buttons on your website to link to your extension's Chrome Web Store page prior to the stable release of Chrome 71. And if you haven't already, be sure to read up on how to create a high-quality store listing, and consider using our 'Install' badge on your site." So anyway, I think this represents sort of the inevitable evolution of extension management in Chrome.

And actually it's interesting because this ties into something else that I saw that I just thought I'd mention because I thought it was interesting. Speaking of misleading and deceptive, cheesy developers, or at least developers of cheesy, probably cheesy Android applications, have figured out a clever means of misleading people about the popularity of their applications on the Android store. They are using developer names which display under their applications' icon of, for example, "more than one million downloads" is the developer's name of some of these applications. And there's more than one of them.

So the unwitting user who is using the popularity of applications as a signal for whether or not it seems like a good thing, are actively being misled by just this renaming of the developer's name. Some developers have gone as far as to put that in the icon of the application itself, which again is probably not going to catch out any attentive listeners, probably not people of this podcast, but it's just interesting that here the whole social media side of, well, not social media, but the metadata surrounding how popular applications are as a signal for whether it seems like a good thing is being subverted.

So it does look like we're moving, of course, Microsoft has famously set up the Windows Store. We've got Google and Android Play Store. And now Chrome is also moving all of its extensions in-house and asking people to come there. And I'm sure that that's what the Chromium people were talking about when they talked about the metadata surrounding the extension being useful for helping people decide that this was an extension that made sense for them.

So I wanted to take a minute to tell people where I am with SQRL. I haven't talked about it for a number of weeks.

Leo: SQRL. We missed SQRL.

Steve: SQRL. It remains finished, as I had [crosstalk].

Leo: Well, that's a good - it would be sad if it became unfinished.

Steve: Well, you know, there have been some...

Leo: Regressions?

Steve: Well, somebody will do something that somehow got through all of our testing. And it's like, oh, okay, good, you know, thank you for reporting that. And in fact, I mean, as you remember, Leo, it's been running for quite a while as I and the group that I'm working with in the GRC newsgroups have been nailing down every last aspect of it. And so when I say it's remained finished, that's sort of the way something that's been this large a project happens. It's not done instantly. It's, okay, I think it's done. Whoops, okay, here are a few more problems. Okay, I've got those resolved. Now I think it's done again. Oops. Oops. Okay. Fewer this time, but, you know.

And so it's been done for a while. So much so that the SQRL public web forums exist and have been online for a while. We've got, I mean, the groups are populated. Everything is ready to handle the announcement except that I don't yet have SQRL login on the SQRL forums. And seems to me that being able to use SQRL to log into its own forums would be a good thing to have. I of course have a demo server at GRC, which is what we've been using for quite a while. But it's not the same as to actually log into a real website.

So where I am right now is in the process of bringing up the use of SQRL for logging into the SQRL forums. And at that point we're ready to release it to the world. So getting close. Again, I got a little bit of a hiccup here with my main machine having died on me early yesterday morning. But I'll get the system put back together, and we'll be off and running again.

And I got an interesting note from someone, a little bit more on the propeller-head side, for SpinRite. But I know that we have a lot of propeller-enabled listeners of this podcast, so I thought it would be interesting for people because it's not something I've ever talked about before. This was sent by Brian in Albany, New York. The subject was "SpinRite reboot after scan completed." And so he addressed you, Leo, saying "Longtime listener, sometime feedback provider." Then he said: "Steve, I'm not sure this will make the June 5th episode or not." And he sent it on the 5th, so that was two weeks ago.

He said: "Is there a reboot option in SpinRite?" He says: "I have some machines that I would love to run this on for maintenance. If only there is an option in Settings to reboot after a scan. This would allow the SpinRite operator to start a scan and walk away." He says: "So long as there are no issues, if the machine would reboot after 100% of scan, the SpinRite operator would not have to intercept the machine prior to the user coming back to work at 8:00 a.m." He said: "I'm using the ISO made from the EXE. Is this a possibility in a future release?" He says: "I searched the SpinRite manual, and there's no mention of this that I could find. All the best, Brian in Capitaland, New York."

And the answer is that could be done today. In fact, that could be done since SpinRite 2, I think. SpinRite is set up to be easiest to use for the non-techie. So when you create a bootable ISO or a bootable floppy, for those who still have floppy drives, it boots FreeDOS and uses a feature in the DOS config.sys to execute SpinRite as the DOS shell. So it just - it runs SpinRite. And when SpinRite ends, it just stays there so that the typical SpinRite user can browse through the screens, check out the SMART data, look at the final map of what SpinRite did, which often provides a lot of information, and then say, okay, I see how things went, and then terminate SpinRite.

There is, however, a mature command line vocabulary. And SpinRite can be run from the command line or from a batch file just as easily as from being the Windows shell in config.sys. So, Brian, and anybody who's interested, if you edit the image of the little DOS file system which is there, it's a little FAT file system, first of all, run SpinRite and do /help, and you'll get a help system which is built into SpinRite. There's a graphical user interface; a full little explanation of all the verbs. And there is an autoexit verb which tells SpinRite, don't remain in, but exit once you're finished.

And so it would be very possible to add the autoexit to the command line in SpinRite. And then what you would do is you would set up the system so that the boot priority was the main drive. That is, so that it doesn't normally boot from the USB. It boots from the main drive. But all BIOSes now allow you to do, typically it's F12, and so a one-time boot override where you say boot from the device I select. So the idea would be - oh, you also need to get a little reboot command. There are reboot commands around for DOS. They've been around since the dawn of time. And so the idea would be the batch file would run SpinRite with the autoexit verb, and then would reboot.

So you boot the system with the USB dongle plugged in, override the normal boot sequence, which would go to the system's main drive, tell it to boot just this one time from the USB. That fires up SpinRite. You get it started, and you walk away. The system will then, once SpinRite is finished, drop back out, issue the reboot, restart the system, ignore the USB drive, and go directly to the hard drive. So it's absolutely possible using existing technology. And it sounds like a good feature for - and what's interesting, too, is it's possible to have the reboot bypassed. So that would be a nice feature for a future version of SpinRite, sort of a "run once" capability where it would reboot the system and then not accept the boot the second time, but just allow it to bypass and go directly to the hard drive. So I'll keep that in mind. And Brian, thanks for the great idea.

Leo: FPU. PU. PU.

Steve: Yeah. I should have done that. That would have been good. I missed that one, Leo.

Leo: Oh, yes. You have to have a mind of an eight year old, that's all.

Steve: Well, no. Zippity Do or Don't, that was last week.

Leo: That was you, yeah, okay.

Steve: I could have done FPU. So, okay. So first of all, we had the much-needed second Tuesday of the month update last week. As of that, we have the availability of mitigations for the SpectreNG, the Spectre Next Generation we talked about a couple weeks ago. Okay. So here's where we are in terms of what's available and what's enabled by default and where and when. Windows 7, 8.1, and 10, so all three platforms. And you know, in this listing they didn't enumerate x86, that is, 32-bit versus 64. So I'm not sure about 32. This certainly is the case for 64, which is where Microsoft's focus is.

But Windows 7, 8.1, and 10 have Meltdown mitigations available and enabled by default. Those three platforms - 7, 8.1, and 10 - also have both of the first two Spectre variation mitigations available and enabled by default. But things are different with Spectre Next Generation: 7 and 10 are the only two versions of Windows with Spectre Next Generation mitigation available, but it is disabled in both cases. And 8.1 does not have Spectre Next Generation mitigation available at all. So it's not clear whether 8.1 will follow. I sort of think it's going to. Maybe 7 and 10 they focused on, just because those are the larger install bases, and then they'll catch up with 8.1.

But even so, even while Spectre Next Generation, that is basically v4, as it's often called, because we have 1, 2, and 3 with Meltdown and the two Spectres, and this is number 4, even though they're available, they're disabled in Windows 7 and 10, the reason being the performance hit, balanced off of the fact that there's no known exploit of them anywhere. So I think, to be responsible, Microsoft decided, okay, well, we've got to offer the mitigation, but we don't want to slow things down as much as turning this on by default would. So we'll have it there so that it's ready in case it's needed.

And what is really significant is although they are available for the server variations, that is, 7, 8.1 and 10 have the server versions, Server 2008 R2, Server 2012 R2, and Server 2016, those are the server variants of those desktop platforms. They all have the mitigations disabled by default. So again, I think this is Microsoft recognizing that, for many platforms, the cure is worse than the problem. And it's just not worth slowing systems down for what is even today, here we are, six months downstream - because this was the first thing we talked about at the beginning of the year. Even now, six months later, this is still a theoretical problem and is not practical.

There is guidance online for anyone who wants to turn this stuff on. Again, probably the most important ones are on by default. The v4 is not enabled, the Spectre Next Generation, again because there the performance hit is profound relative to nobody having demonstrated an active exploit of this. And remember, too, that for your typical user, I mean, the real danger is in a virtualized environment with multiple virtual machines, where one or more might be nefarious, being able to crack the VM boundary.

Again, it's a theoretical problem. It's never been seen. And maybe Microsoft has some inside information about how difficult it would actually be to pull off. But for the typical end-user it's just probably not worth worrying about until there's any sign that this has actually happened. I mean, we're routinely talking about stuff that's devastating in the wild, you know, ripping across the Internet, flooding the Internet with port probes from botnets, versus oh, darn, maybe there's a chance of information leaking inter-virtual machine in some situations that could cause a problem. I mean, yes, the industry did what it had to do because this was a theoretical problem. But here we are half a year later, nothing.

Leo: Not one exploit; right? Nothing.

Steve: Nothing, yeah.

Leo: Is there safety in numbers? I guess not because, if you're a target, it doesn't matter that there's a billion other targets.

Steve: Right. So I would say in a high-risk environment you want these things enabled. But the typical end-user, you know, there's really what information, I mean, here's the problem. That suggests that you've got malware on your machine; right? So if you've got something malicious on your machine, it's game over anyway. The reason it's interesting for virtualized environments is that bad guys could deliberately be running malicious virtual machines, trying to get across into other virtual machines on the same hardware.

Leo: That's the issue, yeah.

Steve: Yes.

Leo: If you're running a server, you need to protect yourself, yeah.

Steve: Yeah. And so here on a personal workstation, if you've got something, you know, if you've got something that is in your system, the last thing it's going to do is use Meltdown and Spectre. It's just going to look around and do whatever it wants to.

Leo: Maybe that's what killed your XP machine. Maybe that was the real reason. It melted down.

Steve: Okay. And here's the other thing is there are some weird bugs, Leo, some weird side effects with these that have been introduced...

Leo: With fixes, yeah.

Steve: Yes, with the Spectre and Meltdown patches. Get this. Some non-English Windows platforms may display the following string in English instead of the localized language. Your non-English Windows may say, "Reading scheduled jobs from file is not supported in this language mode." What?

Leo: Okay. Talk about side effects.

Steve: Exactly. It says this error appears when you try to read the scheduled jobs you've created and Device Guard is enabled. Doesn't come out. And they don't know why. Doesn't come out in the foreign language. Just that one sentence comes out in English. Oh. And then of course this makes them a little nervous, too, because they're thinking, well, what else is broken that we don't know about? Also, when Device Guard is enabled, some, again, non-English platforms may display the following strings in English instead of the localized language. So suddenly it'll say "Cannot use ampersand or dot" - you know, period - "operators to invoke a module scope command across language boundaries." That just comes out in English because, you know, why not?

Leo: Why not? No one knows what it means anyway, so who cares? Who cares what language it's in?

Steve: Exactly. Exactly. Or "Script resource from PS desired state configuration module is not supported when Device Guard is enabled. Please use script resource published by PSDSC resources module from the power shell gallery." So some bizarre power shell thing doesn't work with Device Guard. So again, it's like, okay, that just sort of, I mean, like, what? So anyway, yeah.

Also, some users, a little more on point, some users running Windows 10, the latest version, 1803, may receive an error: "An invalid argument was supplied when accessing files or running programs from a shared folder using SMBv1 protocol," which as we know has been long deprecated, but still generally enabled unless recently disabled.

Also, bizarrely enough, a stop error, meaning a blue screen, right, occurs on computers that don't support the streaming single instruction multiple data, the SIMD Extensions 2, the SSE2. Boom. Blue screen. Whoops. That's not good. And there's an issue with Windows and third-party software that's related to a missing file. And there's like an OEM and some number dot inf that causes the network interface controller, your LAN, essentially, controller to stop working. So you just go off the 'Net. Or, I mean, like completely offline. It's like, okay, that's not good.

So these things make Microsoft a little uncomfortable. And so I think they're thinking, okay, we can't turn these on on servers. Servers can't go offline all of a sudden. And so there's some stuff they haven't figured out yet and haven't fixed. Which seem like bizarre side effects from, like, turning off a performance optimization feature. But there it is.

Leo: Hmm.

Steve: Okay. And, finally, so that's where we are with Meltdown and Spectre. Basically everybody's been patched now. The important things are turned on. The more expensive things are turned off. And still six months later we've never seen this ever, ever used. And arguably there's zero real, like, obvious danger for any end-user. The only way to be infected is if something is running in your computer. There was some concern that maybe a malicious web page could do this. But web pages do not, you know, the timing information has already been fuzzed by other mitigations against other attacks. So web pages don't have the resolution - the code, the JavaScript running on a web page isn't a means into your system. You've got to have native code running on your machine. And if that's the case, you've got malware, buddy, so you've got bigger problems than something maybe trying to breach one of your boundaries.

Okay. Lazy Floating Point State Restore. I have to say I wasn't surprised when I saw this. It's like, okay, yeah, makes sense. In all of our multitasking systems, meaning all modern computers, it used to be a big deal that you could be doing two things at once. I'm unable to do more than one thing at once, but that's another...

Leo: But if you switch really fast, it'll feel like multitasking.

Steve: Exactly. So there's this sort of an abstraction that we've talked about on the podcast often known as a "thread," a threaded execution, which is where the program counter in the processor is as it steps along and executes instructions. And it reads things from memory, and it writes them back, and it moves data around the registers. At some point it's time for somebody else to get a chance to do something. Like if you've got multiple things going on, you've got your PDF reader is open, and your browser's open, and your clock is ticking, I mean, there's a lot of stuff going on in your system.

What's actually happening is the processor is being yanked around between different things to do, with varying scheduling algorithms and so forth. And if you have more cores, then you've got more actual execution threads. But if you even had a single core - before we had multiple cores we just had one processor. We still had multitasking. It's because that one processor was very busy jumping all over the place, working on something, and then having what's called a "context switch." A context switch is the entire execution state of that thread is saved somewhere, and the execution state of another thread which had been suspended is restored.

So what that actually means is the value of all the registers is written to memory so that they could be read from memory and restored when it's time for that thread to start up again. Now, context switching is an expensive thing to do because our modern processors have a lot more registers. There are larger registers. There are more registers. There's also what we were talking about before, the SIMD bank. There's XMM and EMM and just like, I mean, if you look at the context of a state-of-the-art processor, there is just a ton of stuff. And the problem is every time, since you only have some number of cores, and there are many more things that have to be done than the number of cores. So you have to put those registers, all of the registers, away somewhere, and you need to bring them back.

Well, what Intel in their somewhat less than, it turns out, infinite wisdom decided was, huh. If a thread doesn't use any floating point instructions, then we don't need to restore the context from what it was the last time we switched away from that thread. Now, they had to make, again, sort of a value judgment. You can't do that with the main registers because those are actively used all the time by everything. But, you know, floating points, kind of off to the side a little bit these days. Maybe, I mean, they must have done an analysis and saw that a substantial number of threads were never touching the floating point unit while they had processor, while they were in context. They just didn't ever get around to doing floating point. Maybe they would half an hour later, or 10 minutes later. But not this time.

And so the engineers said let's allow lazy floating point state restore. Because it's expensive, because we've got to, like, read all of this from main memory through the caches, flushing the cache and waiting for memory, because it's an expensive thing to do, for threads that don't touch the floating point registers, let's not bother. Let's be lazy about it.

And so there's two terms. There's "eager restore" and "lazy restore." And what Intel did was they created a bit in the CPU for whether the system would be supporting lazy floating point state restore. And you know where this is going, if you've been following along this year. Because it takes time to restore the floating point state, it's possible for an attacker thread to learn information about the previous state of other threads or processes in the system by leveraging whether the floating point state was restored or not. And this has been leveraged by some clever security researchers in Germany to exactly this end. This affects all of the Intel core-based microprocessors, none of the AMD, none of the ARM architectures, none other than Intel. Also, even recent versions of Linux from kernel v4.9 on, and 4.9 came out in 2016, so for the last two years Linux has not been affected by this.

And I don't know what the history is. I didn't have a chance to dig in and decide whether they just decided it wasn't worth the complexity to support it. But that kind of would be my feeling. They thought, eh, how much time is this really saving? Although older versions of Linux would be at risk if this ever became a thing. I mean, again, here we are just having said that Spectre and Meltdown really, like, okay, well, we're ready for them if they're a problem, but so far we haven't ever seen it. Recent versions of Windows, including Windows Server 2016, the latest editions of OpenBSD and DragonFly BSD are also not affected. And again, I don't know why Server 2016, what it was that got it out of there because Microsoft has published a security advisory offering guidance for this lazy floating point state restore vulnerability, explaining that they are already working on security updates to be released in July's Patch Tuesday.

So three weeks from now, basically lazy FP state restore goes away. It was sort of nice while Intel offered it. Probably didn't save that much time. Looks like other operating systems had already abandoned it. These German researchers figured out there was a way to abuse it in the same way that speculative execution can be abused. Basically, any of these things that are optimizing for performance that leave some state information behind can be used to leak information across thread and process and security boundaries. And so as an industry we're backing off of those things.

Leo: OpenBSD announced that they're disabling hyperthreading by default.

Steve: Interesting. Oh, that's interesting.

Leo: Yeah. Let's see. Simultaneous multithreading implementations - this is from the OpenBSD CVS list. "Since simultaneous multithreading implementations typically share TLBs and L1 caches between threads, this can make cache timing attacks a lot easier. We strongly suspect this will make several Spectre-class bugs exploitable," especially on Intel's SMT implementation, which they call "hyperthreading." We really should not run different security domains on different processor threads on the same core.

Steve: Yeah.

Leo: So basically, "Since we suspect there are serious risks, we disabled them by default."

Steve: And, you know, hyperthreading, it was something Intel did back in the mono core days.

Leo: Right, multithreads on a single core.

Steve: Yeah, exactly. And it gave, I don't know, was it like 10%? It wasn't like, oh, I got a second processor. No.

Leo: No, because there's a lot of overhead in the switching.

Steve: Yeah, you've got a little 10% kind of more processor. It just never really offered a huge amount of really that much leverage. Certainly not enough to justify the potential danger of two separate threads executing in different processes and sharing any sort of state. So that sounds like a good move from BSD. So anyway, that's really all we know at this point. Intel has said nothing more than that this problem exists. Red Hat has an advisory published. We're going to get patches in three weeks. I doubt anybody is going to notice any performance hit because it looks like a bunch of OSes have long since backed off from it and just decided it's not worth having one more thing that's exploitable.

So anyway, we're continuing to see instances where this kind of leveraging of performance that breaks our trust boundaries can be abused, and we're having to give it up. So we'll have to find more - actually, I think if the super fast nonvolatile RAM technology, the XPoint stuff...

Leo: Optane, yeah.

Steve: If Optane happens, I bet you we're going to see some real acceleration there because our processors are running at 3GB. They're not getting 3GB of data. They're starved. They're sitting around saying, okay, you know, can anybody do anything right now, or are we all waiting for memory?

Leo: Optane might be the next hyperthreading, though. You never know.

Steve: That's true. I was sorry also to see that it doesn't have infinite writability. I mean, it fatigues also over time.

Leo: Oh, interesting.

Steve: Yeah. And so there's, like, now they're talking about fatiguing in terms of how many times you can write the size of the whole storage per day and for how long it lasts. So if you had a 2TB memory, then it might be like you could write 20TB per day for some length of time before it dies. I think it's very robust, but it's not infinite. So that's one place that hard drives really seem to be out there is that the technology is inherently reversible. All the other things we've come up with so far are, eh, mostly reversible, but not quite. There's a little bit of fatigue that happens.

Leo: It's interesting because this could really in the long run just end Intel. Apple's talking about using ARM chips. Microsoft, I didn't realize this, but Microsoft Research has its own processor design they call E2. And they have successfully ported Windows 10 and Linux to E2. So "E2 is a radical departure from the computer chips in use today. It uses an instruction set architecture known as Explicit Data Graph Execution, or EDGE." I have no idea. I have no idea.

Steve: Cool.

Leo: Yeah. Yeah, I mean, there's no E2 processor. They're all running on FPGAs. But still I think it's no accident these companies are looking at alternatives to Intel.

Steve: Yeah.

Leo: I would be, too.

Steve: Well, Intel had a good run.

Leo: They had a good run. They had a good run, yup.

Steve: Yup. They had their heyday. And it was inevitable that no, I mean, look at OSes. So did Windows. And now there's been a lot of shred. Yup.

Leo: Yup. You know, that's the good thing about being in this business is when you're covering technology, you're never bored.

Steve: No.

Leo: Your computer might die.

[Clip] BONES: It's dead, Jim.

Leo: But you're never bored.

Steve: Were we to rename this podcast, "Never Bored" would be good.

Leo: Never bored. Stimulating security news every Tuesday, 1:30 Pacific, 4:30 Eastern, 20:30 UTC, right here at TWiT.tv/live. Join us in the chatroom if you do that, irc.twit.tv. And of course you can get on-demand versions of the show from Steve's site, GRC.com. He has not only audio, but also transcriptions. A few days after the fact you can read the transcriptions from all 668 shows. It's an easy way to search, too.

While you're there, check out SpinRite, the world's best hard drive recovery and maintenance utility, and all the other projects Steve's got his fingers in. But SpinRite's his bread and butter, so that's the one we want to make sure everybody knows about.

Steve: Yes, please.

Leo: Yes, please. Go to TWiT.tv/sn. We have audio and video there. Actually, really the best thing to do is find your favorite podcast app, there are lots of them now, and subscribe. That way you'll get Security Now! the minute it's available, in a hot second. We're going to push this out. We're going to edit it up, push it out, usually by the end of the day Tuesday.

Steve: Yeah.

Leo: Thank you, Steve.

Steve: Thank you, my friend. Next week who knows what will have happened. But we'll be here to cover it.

Leo: Never boring. See you then.

Steve: Thank you, my friend.