TPM-FAIL

Description: This week we look back at November's Patch Tuesday while we count down to the impending end of patches for Windows 7 and Server 2008. We check in with CheckM8 and Checkra.in as the iOS bootrom exploit continues to mature. We look at GitHub's announcement launch of "GitHub Security Lab" to bring bounties and much stronger security focus to the open source community. We discuss a recent court ruling regarding U.S. border entry device searches. We cover yet another bad WhatsApp remote code execution vulnerability. We examine the impact of version 2 of ZombieLoad, the formation of the Bytecode Alliance, and a bit of media miscellany. Then we examine the impact of two Trusted Platform Module (TPM) failings, one which allows local key extraction, and a second that can be exploited remotely over a network.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-741.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-741-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. I am back in the saddle. Well, the bouncy ball, anyway. We've got a lot to talk about, including those Patch Tuesday updates last week. We'll talk about CheckM8, the iOS jailbreak you don't have to worry about, and why the Trusted Platform Module is just a little bit less trusted these days. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 741, recorded Tuesday, November 19th, 2019: TPM-FAIL.

It's time for Security Now! the show where we cover your privacy, your security, everything you need to know before you go out there in the Wild West of the Internet with this guy right here, Steve Gibson of the GRC.com. Hi, Steve.

Steve Gibson: Leo, welcome back. Everyone recognizes your voice after four weeks with Jason.

Leo: Yes. I have returned. Boy, I was gone a long time, wasn't I.

Steve: How is the recovery? It always seems to be like the jet lag coming back is the tough one. You're with us?

Leo: Well, I thank you and your Healthy Sleep Formula because I used it both coming and going. And the jetlag going there, even though it's as bad as you can get, by the way, because we were 12 hours difference...

Steve: Ooh, completely on the other side.

Leo: Yeah. It's 2:00 in the afternoon now. It was 2:00 in the morning there. And that's very bad. But we were able to - we flew to Athens. Of course we try to sleep on the flight. We try to get in shape. But your Healthy Sleep Formula meant when we arrived in the afternoon I walked around, I stared into the sun, you know, I got a lot of photons into my brain to let it know it's daytime, it's daytime, it's daytime. And then at bedtime, stayed awake till bedtime, used the Healthy Sleep Formula. And I used it the whole trip, and it really helped. And though I have to say coming back, you're right, it's worse. I'm a little, I don't know, loopy. So I upped the dosage on the Healthy Sleep Formula because I had the nicotinamide into half, just half a pill. But now I'm using the full horse pill. And last night, great night's sleep, didn't wake up in the middle. So I credit you. I think that that is the secret cure...

Steve: Cool, for jet lag.

Leo: ...for jet lag, yeah.

Steve: Maybe I'll take my own advice next time.

Leo: You didn't do it last time?

Steve: It didn't even occur to me, no. And I've heard you mention it many times. And I just, you know, I was...

Leo: Melatonin's in it, and that's a part - a lot of people have recommended that for a long time in jet lag.

Steve: Yes. Although a lot of people overdose on that.

Leo: They take too much.

Steve: We only make a tiny little couple milligrams is all that you want.

Leo: Yeah. I've been taking 300 micrograms, and that's what I took on the whole trip.

Steve: Yes, good.

Leo: But I did the one milligram time release that you recommend, and that's what I did last night. And, man, I slept great.

Steve: So I've been holding off on our Picture of the Week because it's actually a Video of the Week.

Leo: Are you saving it for me?

Steve: I've been waiting for you because you and I are both programming language people.

Leo: Oh, yeah.

Steve: And I didn't know about that with Jason. So I thought, well, I'll just hold this one till Leo gets back. But we're going to talk about a potentially significant new piece of research which has found fault in two popular Trusted Platform Module solutions.

Leo: Ooh.

Steve: Yeah, one is part of Intel firmware. The good news is it's pervasive, but it can be patched because it's firmware. The other one is an STMicroelectronics extremely popular TPM. And because it's in hardware, there's nothing you can do. So we're going to talk about that stuff. But there's a lot of good, fun, interesting news this week. We've got November's Patch Tuesday, which occurred last Tuesday. And we're counting down with not a very large count to the impending end of patches for Windows 7 and Server 2008, with some caveats for enterprise.

We're also going to check in with the CheckM8 and Checkra.in bootrom exploit to look at its maturation. It's continuing to mature. We saw the first public release last week. It's come a long way in one week. We're also going to look at GitHub's announcement launch of the GitHub Security Lab, which aims to bring bounties and much stronger security focus to the open source community. So no longer will it just be the proprietary software, where you're saying, Microsoft, something seems wrong over here. Or Apple, or whomever.

We also discuss a recent court ruling regarding U.S. border entry device searches. We'll cover, yet again, another remote code execution vulnerability in WhatsApp. Those just keep on coming. We also have now ZombieLoad v2, which we'll take a look at, as well as the formation of the Bytecode Alliance, which is interesting. Then a little bit of media miscellany that you and I are going to have fun talking about.

And then we'll wrap up, as I said, by looking at the impact of two Trusted Platform Module failings, which of course is important because it's very much like - it's related, actually, sort of to the Apple bootrom chain, boot chain problem because the only way we can secure our systems is if there is an anchor of trust which cannot be violated. And that anchor then starts a chain of trust where each link is verified before it is trusted. And so, as the name sounds, the Trusted Platform Module is supposed to be the thing on your motherboard which you can absolutely trust. Except you can't, in a couple instances.

Leo: Okay. The video. I've got it queued up here. Tell us about it.

Steve: So I think because this is an audio podcast, you should probably narrate this like a horse race.

Leo: Yeah. You're bad.

Steve: Because that's what this is. What this is, this is so cool. Now, many people have seen it. It's on YouTube, 2.6 million views.

Leo: So what this is, and we've seen these before, these graphs that show over time the popularity or unpopularity or whatever of something.

Steve: Right.

Leo: And how are they deriving the popularity? Because this is programming languages.

Steve: Good question. I didn't dig into it to figure out.

Leo: I think the Stack Exchange, or there are some sites that do this. GitHub does it. So I'll check and see where the source is. But here we are in 1966, and you probably guess, Fortran has almost half the market, followed by COBOL, ALGOL, Assembler.

Steve: Woohoo.

Leo: Then down under 5% you've got APL, you got BASIC...

Steve: Yeah, I didn't even know anyone actually programmed in APL.

Leo: Because you have to have a special keyboard. It's crazy.

Steve: It's got widgets and squirlicues and things, and yeah.

Leo: But remember, this is 1966. If you were a programmer...

Steve: And of course you need some really strong parentheses keys if you're going to do Lisp. And that was at 1.48%.

Leo: And that's because Lisp actually is one of the oldest languages, only one year younger than Fortran. But it's going to - I have a feeling, I'm just going to guess, it's going to diminish a little bit as we go through the quarters. Here we are in '67. APL is starting to beat Assembler. APL going over 5%. Fortran, COBOL, ALGOL still in the lead. APL fourth, Assembler fifth. The growth in Assembler is starting to take over. Assembler now becoming very popular. Oh, wait a minute, here comes a newcomer, Pascal in 1970. LISP is up to 3%. It's growing, now 4%. But Pascal's growing, too.

Fortran's still on top at only 24% of the total. COBOL, BASIC, Lisp. Lisp is up and coming. This is what I like to see. In 1973 the number third most popular language is Fortran. COBOL, Lisp, then BASIC, ALGOL, Pascal now taking over for ALGOL as it well should. And in 1975 for BASIC, wait a minute, it's beating Lisp now. Pascal in the third position. Fortran, COBOL, Pascal. Coming around the clubhouse turn we've got - I can't keep doing this. In 1977...

Steve: You're doing a fantastic job.

Leo: It's Fortran, followed by Pascal in the second position. Then COBOL drifting back behind BASIC, which is coming on strong. Lisp is hanging in there at almost 10%. But that's not going to last long as Pascal becomes the number one programming language in 1980. Shrinking market share with Fortran. BASIC hanging on at 20%. Then there's C, a newer language, started in 1970, coming on. It's now number two. Pascal, C, Ada, brand new in 1983. It's number three, then Lisp, Fortran, BASIC, Assembler, COBOL. But here we go, 1984 and here comes C, coming on strong. It's now the number one language. But Ada, the language from the Pentagon, is suddenly taking over. That won't last.

C takes over again in 1987, and Ada's going to shrink away to nothing. Pascal's shrinking, too. So is Lisp. In 1988 it's C, Ada, Pascal, Lisp, Fortran, C++, a young, up-and-coming language that is soon to take over, I suspect. Oh, yeah, watch C++ coming on strong. Here we are in 1990. C++ in fourth position after C, Ada, and Pascal. Lisp, you know, you just can't kill Lisp. It's still in there at about 10%. C and C++, they're neck and neck. C has 71% market share, but C++ has 20% in 1993, and on and on and on. And we're only halfway through, and I'm going to completely lose it. But I think, you know, it's funny is you can watch this, and you kind of know what's going to happen. I mean, this is a story we know the ending of.

Steve: Yeah, I just, for me, for those who are listening and haven't seen it yet, I did...

Leo: Here comes Paul Thurrott's Delphi, which took over for Pascal, is beating Pascal, because it is Turbo Pascal.

Steve: Yes, it is.

Leo: So a lot of those Pascal users are kind of giving in to Delphi. Oh, my god, Java just came out in the mid-'90s, and it is coming on strong. C++ is actually dropping. Here's PHP in the fourth position. And JavaScript. By 2000 it's Java, C, JavaScript, PHP, and C++. You could see the impact that the web has had on programming languages because both JavaScript and PHP are really designed for web pages. Java, too, really, and it's at 34% in 2004. JavaScript 24%. 24% for PHP. C and C++ kind

of hanging in strong. Lisp has completely fallen off, as Python, Perl, and Ruby start to climb the charts. Visual Basic even has 6% of the market. By 2007 it's Java, JavaScript, PHP, C, and C++. MATLAB, which I don't know where MATLAB came from. And Objective C comes up as the Apple starts to get popular.

And in 2010 Objective C actually is beating Visual Basic. It's now Java in 2011; JavaScript; PHP; Python; C#, Microsoft's version of C++; followed by C++. I'm surprised actually to see that C++ has such scant market share in 2013. It's only 8%. I would have thought it'd be easily second or third. Python has come all the way up to third position now in 2015. Growing fast, too. In fact, I think we're getting close to the end of the line here in 2016 with Java, JavaScript, Python, and C#. I wonder if Java's going to plummet, though, as we get closer to 2019. Is it going to be a sudden loss for Java? Or, no, it's hanging in there. Well, it dropped a couple of places.

Steve: You were right, yeah.

Leo: Python, JavaScript at the end in Q3 2019, 24% Python. Wow. Who woulda thunk it?

Steve: Yeah, yeah.

Leo: Followed by...

Steve: Well, you did a beautiful job, yup.

Leo: Sorry about that. I hope people fast-forwarded.

Steve: Well, so grc.sc/languages. Because I just think this is - it's so cool...

Leo: It's worth seeing, yeah.

Steve: ...for anybody, yes. And what our listeners couldn't appreciate is that the bars were changing their relative lengths and also swapping places. So you really got to see the languages, like, move to the top, and then hang on for a while, and then lose out to somebody else, and then slip back down. So again, I created a little - I used the GRC link shortener, grc.sc/languages. And it's, what, it's like a few minutes of YouTube that I think our listeners will probably get a kick out of.

Leo: This is a new - I'm seeing a lot of these on Reddit, this format. It's a new way of showing data is these moving bar charts over time. And I really like - for some things it's really useful.

Steve: Yeah, I think it's really compelling because, you know, you just get - and for those of us who are little bit long in the tooth, Leo, we've seen all of that happen. So it's

a little bit of a blast from the past. It's like, oh, yes, I remember Fortran. That's what I studied in college.

Leo: Yeah. Yeah.

Steve: And then, what? For who?

Leo: Yeah. I'm sad to see Lisp drop off to nothing. It's held its own, though, for quite a while, for three decades. That's pretty good. It'll be back.

Steve: You think?

Leo: Along with APL, yeah. Yeah, yeah.

Steve: That's right. Just find an APL keyboard, if you can. So last Tuesday was our Patch Tuesday for November. It brought us 73 vulnerabilities resolved across Microsoft's whole product spectrum; 13 of the 73 were rated critical. One of them was a zero-day. That zero-day was found to be actively exploited in the wild. It was a scripting engine vulnerability in IE, which was, oddly enough, independently reported by four different researchers. So somehow it was sort of in the wind.

The vulnerability in IE allows an attacker to execute their own code if a victim were to be coaxed into visiting a malicious web page, or if they're tricked into opening an Office document. There is a way of invoking this through Office. But Microsoft wrote of this, they said: "An attacker who successfully exploits the vulnerability could gain the same user rights as the current user. If the current user is logged in with admin rights, an attacker could take control of an affected system."

And in the Office document scenario, an adversary might embed an ActiveX control marked as safe for initialization into an Office document. If the ActiveX control were initialized, it could arrange to execute code from a malicious website, essentially to create a booby-trapped Office document. So even though, like so even for people who are not using IE, you're using Edge or Chrome or Firefox, even so the IE library is still in Windows, and it can be invoked through Office.

So anyway, so that's worth having patched for last week certainly because it was found being used in the wild. I don't remember if they said in targeted attacks, but that's what I would expect since sending somebody a Word doc and say, you know, here's the receipt for your traffic ticket or something, and people go, huh?, and then unfortunately make the mistake of opening it up because they're curious, and they get themselves owned.

So the other interesting thing that I had noted was that Intel, like Adobe, have now started synchronizing their own patches to coincide with our Patch Tuesday. So, I mean, everybody's going to be synchronized here on the second Tuesday of the month. In the same way that Adobe often, but not always, releases their updates on the second Tuesday of the month, so too now will Intel. Also there were nine patches for Microsoft's Hyper-V virtualization. And four of those, of the nine, were rated critical because they could potentially allow for remote code execution. You don't want remote code execution in your virtualization system because of course we know that that's how bad guys get into a shared hosting machine and can get up to some mischief. And then there were

three critical flaws in the soon-to-be-retired original Edge browser, one also affected Exchange Server, and the remaining four were in various Windows components.

And then the final issue, which we'll be getting to at the end of the show, that Microsoft mentioned in this Patch Tuesday update, was the weakness discovered in the way STMicroelectronics Trusted Platform Modules implement the Elliptic Curve Digital Signature Algorithm, ECDSA, in version 2 of their hardware. However, Microsoft did note that that's not one of the algorithms in the TPM that they make use of. So that mitigates the concern. Although other things running on the platform might well be making use of it. So you definitely would want to know one way or the other.

But speaking of Patch Tuesday, we should note that only two more Patch Tuesdays remain for Windows 7 and Server 2008 systems. So that's next month's December 2019, and then January 14th of 2020 is the final Patch Tuesday. We're already in the extended whatever they call it, release phase of Windows 7. I don't know why they even bother doing that. They officially stopped releases years ago. But of course everyone was still using it so they said, well, okay, we'll just call this the "extended service phase" or something. But they promise that they're actually going to stop after January, so there will be no February Patch Tuesday unless you're an enterprise customer who's able to get into the extended security updates for eligible versions of Windows.

I found a graph that I thought was interesting, which we had talked previously about how we suspected that most end users had probably moved over to Windows 10. For whatever reason they didn't stay on 7. This is a chart - shoot, and I meant to note from where it was. I know that it was in August, so it's relatively recent. And it shows three classes of users: customers, very small businesses, and small and medium-sized businesses and enterprises. And what was interesting is, as of late summer, Windows 7 and 10, those red bars, they're neck and neck. They're at 47% each.

So it's true that, overall, Windows 10 has taken a lead as we'd covered at the beginning of this year. It finally pulled ahead. But not surprisingly, the bias is in the end user who said, "Oh, new Windows? Yay." But the enterprise said, "New Windows? Nay. No, thank you." So it's really going to be interesting to see now what enterprises are going to do. Why is it that they haven't made the move? And of course one of the problems I think is that Microsoft has demonstrated that they're not going to let Windows ever settle down. I heard you, Leo, I think it had to have been since you were back, maybe it was on Sunday, just sort of bemoaning the fact that Microsoft is really having problems with Windows, with Windows 10.

Leo: With 1909. The latest version is causing all sorts of problems. And I thought that this would not be happening after a while. But no.

Steve: Well, right. And so once upon a time it made sense for a rational Windows 7 user to hold off before adopting a new OS. First of all, that would allow its newness to settle down and for most of the problems to be found and fixed. But of course anyone who looked at Windows 8 and then 8.1 would have probably, who had any sense, would have thought, okay, I don't want that. I'm going to keep waiting. And so it was sort of roundly regarded as a bit of a catastrophe. And what's now fully apparent, having been watching Windows 10 for some years, is that Microsoft deliberately has no plans to allow Windows 10 to ever, in air quotes, "settle down."

Leo: Yeah.

Steve: I mean, it's just not going to happen. So it's going to continue to be an unsettled work in progress, which is I think turning out to be not a wise move. It's one thing to do this to end users. But enterprise doesn't want a constantly moving target like this. So it's going to be interesting to see. I mean, what's odd to me is that I looked into - I have a link in the show notes, techcommunity.microsoft.com, here at the end of this story. This is from Windows IT Pro blog. And the link is how to get extended security updates for eligible Windows.

And I was curious; so I scanned through it. What I saw was that you will be able to register your instance of Windows to continue to operate the way - that is, of Windows 7 Pro - to continue to operate the way it has been. Which is to say it'll get automatic updates for, what is it, I think it's up to three years; right? And so you pay a certain amount for the first year, then you pay more for the second year, and you pay more for the third year.

And Microsoft makes a point of saying you can't purchase partial years. You have got to commit to the whole thing. Which says their entire automatic security patch process is automated, and nothing but policy is causing them to refuse to provide fixes for things that are wrong with Windows 7. That is, these are, I mean, this is the new bait-and-hook is that, oh, you don't want to continue running Windows 7, which works just fine, because it won't be getting any new security updates. Except they're going to have them, and they're going to be selling them and offering them to their enterprise people. But not to everybody else who still wants to be using Windows 7.

So, I mean, to me this feels like an ethical dilemma, that it's not like new features in Windows 7 that Microsoft isn't going to make available. In fact, it's because people don't want new features that they've stayed with Windows 7. No, it's the fixes to the security vulnerabilities due to the flaws in Windows 7 that Microsoft is now going to refuse to continue to provide to people, even though the entire system is in place for them to do so. I don't know. That just, to me, this seems like they've painted themselves into a corner. They've produced an OS in Windows 10 that a lot of people have opted out from accepting despite the fact that it's free. And now they're going to say, well, for three more years we'll provide people who insist on remaining with Windows 7 security updates. But everybody else, sorry, good luck. You know? You're going to have vulnerable versions of Windows because we're not going to fix the bugs that we've left behind in it.

Leo: They don't charge an insignificant amount to buy those updates, too. And it doubles every year. So they really don't - I think they just don't want people to do it. And they acknowledge that some businesses have to. But they just don't want to. They want everyone on 10. If they could get everybody on 10 with a snap of a finger, they would do it.

Steve: And I'm sympathetic to the idea that they don't want to continue to be responsible for an old code base. I mean, I get that. But if they're making it available to anybody, how do they deny it to people who need it and aren't enterprises? I mean, an end user can't buy it. You have to have the volume licensing agreement thing with Microsoft. So it's not like everyone who's chosen to stay with Windows 7 could purchase the additional security patches if they wanted to. They can't.

Leo: Yeah, but that's the point is those security patches are there only because they have no choice. There are some banks and X-ray machines and whatever.

Steve: Ah, okay.

Leo: They want everybody on 10. And by the way, there's a lot of good reasons. I mean, there's businesses reasons for Microsoft. But there's also developers want to have a platform that's consistent and uniform so they know what the features are. And of course there's the security issue, as well. So I understand there's a lot of reason why Microsoft would want to do this. I'm not defending them. I know what you're saying, Steve, and I don't disagree with you. I don't. I've just given up on Windows, to be honest. Honestly, I have.

Steve: Please, I can't wait to join you. I feel the same way. It just feels like, boy, wrong.

Leo: The good news is desktop Linux has gotten so good that, if people would just open their minds a little bit, there is a route. And almost every PC that runs Windows will run Linux just fine. There is a good exit route. But people are very reluctant. And I, you know, in a business I'm not going to make my employees use Linux. I can't do it. Maybe should charge them double if they don't.

Steve: So last week was the first emergence of Checkra.in. And we noted that the official website, which as you and I talked about I think in the beginning, it was initially - it was Checkra1n.com. And there was no security certificate there. And bad guys grabbed Checkrain, R-A-I-N, dot com and put some...

Leo: Oh, god.

Steve: Yeah, some malware there.

Leo: Of course they did.

Steve: So it was like, okay, the hackers sort of did the wrong thing by doing a hackeresque domain name. Well, now we know the final name is Checkra, C-H-E-C-K-R-A dot I-N. So that, since there is a TLD .in, Checkra.in. That's the official site. So this has continued to evolve. And of course aside from the fact that, as we covered last week, the exploit for the Windows BlueKeep vulnerability has now been perfected so that it no longer has a high incidence of BSODs, I mean, and that's something I'm sure we're going to be talking about in weeks to come.

Arguably the most significant thing going on at the moment is the continuing work to perfect Checkra.in. Which as we know is the iOS CheckM8 bootrom vulnerability exploit. And what makes it significant is, because it is actually a ROM bug, it's never going to get fixed. It's going to persist in all the devices that Apple needs to continue supporting until they finally go out of support, which is probably going to be years.

So anyway, since it's a big deal, I did want to sort of check in with where it stands today. When we talked about it last week, there were three betas: 0.9, 0.9.1. 0.9.2. Now we are at 0.9.3. We had also 0.9.3.2 and 0.9.5, which as of last night is where we are. I didn't check yet this morning, I meant to, to see if we were yet at another one past that. They are continuing to find and fix and squash bugs, and it is becoming increasingly stable.

Since I don't have any of my own first-hand experience with jailbreaking or with this particular emerging jailbreaking system, I went looking for some interesting and pertinent feedback from those who have been playing with it so far. Dan Goodin, who covers security topics for Ars Technica, had done some digging. And I've excerpted, and I paraphrase a little bit from what he found and reported. It turns out those who have looked at it are finding that this Checkra.in exploit is surprisingly reliable and robust. There are still platforms for which there is no support. There are some platforms, which is to say, some iOS devices for which there is no support. There are some for which it's still a little bit flakier.

Ryan Stortz, who's an iOS security expert and the principal security researcher at Trail of Bits, he said in his interview with Dan, he said: "I expected it to be a little rougher around the edges for the first release." And of course it's not even there. It's at beta. He says: "It's really nice to be able to install a new developer beta on your development iPhone and have all your tooling work out of the box." He says: "It makes testing Apple's updates much, much easier."

And of course it promises, once it's finished, to work reliably on a wide array of hardware, meaning everything from the A5 through the A11 chip. The iPhone 4S was an A5, but I'm not seeing it being discussed. So it might be the 5S which is the first one. It may be going no further back. It may have been that the A5 is using on the iPhone 4 an earlier version of iOS, and they're not going that far back with iOS. But in general, it is surprising people for its stability and the speed with which it's moving forward.

So there's a lot of controversy in the industry about its presence. Not that that's going to do anything to dissuade it from happening. The hackers who are putting it together are clearly having the time of their lives. They recognize, unlike previous jailbreaks, that they're investing in something that has legs. Inevitably, if a jailbreak ever becomes public, and it's only when it becomes public that it has an opportunity to acquire some user base, Apple will of course grab it and immediately close it, I mean, fix it in an emergency patch in order to keep this from happening.

So what we've historically seen is that Apple just squashes jailbreaks instantly, you know, at the first chance they get, demonstrating how unhappy they must be with the fact that there is now one that they're never going to be able to fix. And in reading into the industry's reaction to this, there are people who consider it a bad thing because I guess they're concerned that end users will use this in order to do things like we've seen back in the Android era or in other closed ecosystems where users wanted to, like, use an unauthorized launcher for their apps or something.

I don't think that's going to happen because, first of all, as you were mentioning on MacBreak Weekly, our devices are being rebooted from time to time. I'm often picking up my phone after I had used it the night before and being told I need to reenter my passcode, presumably because it did a self-restart overnight, and so it rebooted. So this cannot persist across a reboot. So I regard that as a good thing because I would never promote the use of this for persistent jailbreak on an end-user device.

I think this is interesting for the security community because it will give the security community, the white hat hackers, visibility into what Apple has been explicitly trying to keep away from everyone. Of course the flipside is that the black hat hackers will also get visibility into what Apple is doing, and in both cases, if new problems are found, those could potentially be exploited in non-CheckM8 jailbroken phones in the same way that any new vulnerabilities found in iOS are being exploited during the brief window of opportunity that exists.

So anyway, I think this has been an interesting event. There's nothing Apple can do about it. We are soon going to have a robust, tetherable, USB jailbreak which has

transient persistence, which will allow the security community to take a look into what Apple is doing in a way that they have never been able to before. And I think that's cool.

Leo: Just to be clear - because actually there was a little conversation about this in the TWiT Forums.

Steve: Oh, good.

Leo: The jailbreak is transient, but you could do things to the phone that would not be transient.

Steve: It's not clear because...

Leo: Okay. Because couldn't you install some - so one of the reasons people do jailbreaks is so they can install software, sideload software.

Steve: Yes, but that would not be signed by Apple.

Leo: Oh, and it would then be removed?

Steve: Yes, yes.

Leo: Ah, okay.

Steve: So on a reboot, the first moment that Apple sniffed an app that did not have their signature, meaning it did not come from the Apple App Store, it's gone.

Leo: Interesting. I mean, in theory somebody could be clever and figure out a way around that, perhaps, or rootkit or something.

Steve: Well, I don't think there is a way around that because you're going to then have the whole secure boot chain all the way up.

Leo: Right. And that's intact after reboot.

Steve: It is absolutely intact after a reboot. So the vulnerability would be, if you lost control of your phone at a border - going into China, for example, or Russia, or pick on somebody - and they gave it back to you, it could have had Checkra.in used to install something transiently which would then have residence until you restarted your phone. The comments have been made that the official Checkra.in app does place a Checkra.in app, and it puts a couple things on your home page, making it obvious that it's there. But it's not clear that those could not be removed by a variant of Checkra.in, which certainly somebody else could create.

So what this does on older hardware is create a window of exploit between the time you lose your phone and somebody might compromise it until you next restart it. So the advice has been twofold. First, if you are someone who might be targeted, and your phone is ever out of your control, power it down completely, and then power it back up again, which will again let Apple purge anything that might have been done from the phone. Or use a newer device. Because, remember, nothing that's A12 or A13 is vulnerable to this. So, like Leo, every time I see the back of your phone I think, oh, yup, Leo's got the latest and greatest. He's got the three-lens monster.

Leo: It's a giveaway, yeah.

Steve: So, yeah, not a problem for you ever. And so anybody who's concerned, this would be a reason to update to something that is at least A12. And if you're doing that, you might as well go to A13.

Leo: Right.

Steve: And if, for whatever reason, you have an older device, if it's ever out of your possession, and you have any reason to suspect that you might be a target, just power cycle it, and then you're going to be okay.

Leo: Good to know.

Steve: You know, is tethered, is USB. So it's just in some ways it's sort of like a perfect flaw because it can't be fixed. Apple needs to be providing updates for a long time to the A11 devices. I have one, an iPhone 10. And so I could do that to it. So it's, well, I have an iPhone 6, for that matter, which is still being supported. So I think a really, really interesting exploit. Not the kind of thing...

Leo: And relatively harmless. Relatively benign.

Steve: Yes. Yes. And that's why I'm not, like, worried about it. It is relatively benign. Yet it does lead, I mean, I understand Apple's interest in not letting people poke around. A year from now the poking around that white hats will have been able to do will probably have fixed a whole bunch of things that will benefit all iOS users. At the same time, the people who want to see inside for nefarious purposes, well, they've been able to do that, too. Although it might be that they were able to get in anyway, you know, through various ways. So anyway...

Leo: So you could make a strong case it's really only to the good.

Steve: I think so. I just think, you know, what we've seen is the more people look at software, the more problems are found and fixed. And Apple will fix them, if they are responsibly disclosed. And so there's a huge community of people who see this as their first opportunity ever to roll up their sleeves and see what Apple has done. And there will be improvements to iOS that arise from that.

Leo: Yeah. That's good. Steve?

Steve: So GitHub.

Leo: GitHub.

Steve: Has decided to launch a proactive Security Lab aimed at boosting open source software security.

Leo: Oh. Nice.

Steve: This was a, yeah, this is a very cool effort. This was announced last Thursday at the GitHub Universe developer conference. And in years past we've touched upon a few audits of important open source software. Early on, Matthew Green was largely driving the effort to audit TrueCrypt, back before it became VeraCrypt. There have been a couple of audits of OpenSSL. I just noted that the audit of the new Unbound DNS server has recently been fully funded.

But I think we would all agree that it's probably long past time for the creation of a public effort to formally incentivize and reward those who discover and report problems in open source software. All of the, you know, like the Zimperium that we've talked about, sort of the quasi-gray, we're not sure who they're selling their exploits to, but we do know that they're paying a lot for them, you know, those are all in proprietary closed systems. So what GitHub is doing is saying, hey, let's put together, let's formalize an effort to fix open source software, as they put it, before it's able to do any harm.

Jamie Cool explained at the announcement, actually he wrote of GitHub Security Lab. He said: "We all share a collective responsibility to keep open source software secure. None of us can do it alone. Today at GitHub Universe we announced GitHub Security Lab to bring together security researchers, maintainers, and companies across the industry who share our belief that the security of open source is important for everyone. We are excited to have an initial set of partners" - oh, and wait till you hear who they are. I'll get to them in a second - "who have all committed to achieving this goal. Together we're contributing tools, resources, bounties, and thousands of hours of security research to help secure the open source ecosystem. As part of today's announcement, GitHub Security Lab is making CodeQL, which is like their main flagship gizmo, freely available for anyone to find vulnerabilities in open source code." And of course, frankly, you could do it in your own source code. Maybe that's violating the license, I don't know.

"CodeQL," he says, "is a tool many security research teams around the world use to perform semantic analysis of code, and we've used it ourselves to find over 100 reported CVEs in some of the most popular open source projects." And I'll just note as an aside that GitHub obtained CodeQL from their purchase of Semmle, S-E-M-M-L-E. Semmle are the people who developed this. He said: "We're also launching GitHub Advisory Database, a public database of advisories created on GitHub, plus additional data curated and mapped to packages tracked by the GitHub dependency graph."

He said: "GitHub's approach to security addresses the whole open source security lifecycle. GitHub Security Lab will help identify and report vulnerabilities in open source software, while maintainers and developers use GitHub to create fixes, coordinate disclosure, and update dependent projects to a fixed version."

He said: "GitHub Security Lab's mission is to inspire and enable the global security research community to secure the world's code. Our team will lead by example, dedicating full-time resources to finding and reporting vulnerabilities in critical open source projects. The team has already had over 100 CVEs issued for security vulnerabilities it has found." He noted that: "Securing the world's open source software is a daunting task," he said. "First, there's scale. The JavaScript ecosystem alone has over one million open source packages."

Leo: What? How could that even be?

Steve: Yeah. JavaScript, one million open source packages. Then, oh, and of course it would be useful looking at the usage graph, you know, the distribution, because I'm sure Node.js is, like, way high. It probably, you know, there are probably, what, maybe 50 or 60 that are in heavy use, and then it falls off quickly. But if someone uses one of these obscure ones, and there's a problem in it, they can get their site compromised, and their users. But then, he says: "Then there's the shortage of security expertise. Security professionals are outnumbered 500 to one by developers."

Leo: [Laughing] Wow.

Steve: Uh-huh. So, you know, everyone listening to this podcast...

Leo: Everybody wants to be a coder. Nobody wants to be a security pro, yeah.

Steve: That's right. Who wants to fix those bugs?

Leo: IT, boring.

Steve: "Finally," he says, "there's coordination. The world's security experts are spread across thousands of companies. GitHub Security Lab and CodeQL will help level the playing field." He says: "Joining us in this effort are the following companies, donating their time and expertise to find and report vulnerabilities in open source software. Each have committed to contribute in a different way, and we hope others will join us in the future." So those are, at launch, F5, Google, HackerOne, Intel, IOActive, J.P. Morgan, LinkedIn, Microsoft, Mozilla, the NCC Group, Okta, Trail of Bits, Uber, and VMware. So an interesting group.

Leo: Could be good.

Steve: Yeah. And he says: "To empower the research community, we're also making our state-of-the art code analysis engine, CodeQL, free to use on open source. CodeQL" - get this, Leo, you'll find this interesting - "lets you query code as though it were data. If you know of a coding mistake that caused a vulnerability, you can write a query to find all variants of that code, eradicating a whole class of vulnerabilities forever." Which is really interesting, I think. I have a link in the show notes. To get started with CodeQL, it's https://securitylab.github.com/tools/codeql.

He says: "If you're a security researcher or work in a security team, we want your help. Securing the world's open source software will require the whole community to work together. GitHub Security Lab will run events and share best practices to help everyone participate. Follow" - and they have a Twitter account - "@GHSecurityLab account on Twitter for more details."

They said: "As the world's security researchers uncover more vulnerabilities, maintainers and end users need better tools to handle them. Today the process for addressing a new vulnerability is often ad hoc." Get this: "Forty percent of new vulnerabilities in open source don't have a CVE identifier when they're announced, meaning they're not included in any public database. Seventy percent of critical vulnerabilities remain unpatched 30 days after developers have been notified." And of course we've talked about this, how it's embarrassing, especially when a package hasn't been updated months after the maintainers of it were told of a problem. And then of course the you-know-what hits the fan when it starts getting exploited, even though it's been known for months.

So here he said: "Seventy percent of critical vulnerabilities remain unpatched 30 days after developers have been notified." And of course we've talked about why, too. The maintainers, this is a hen o at the usage graph, you know, the distribvoluntary effort for many of them. They've got a day job. They've got a bunch of stuff to get to, but they haven't gotten to it yet. So one of the things that the Security Lab is working towards is bringing much more automation to this process.

He said: "We're fixing that. Maintainers and developers can now work together directly on GitHub to help ensure new vulnerabilities are only disclosed when maintainers are ready, and that developers can update to fixed versions quickly and easily." He said: "This will be accomplished through GitHub Security Advisories. With Security Advisories, maintainers can work with security researchers on security fixes in a private space, apply for a CVE directly from GitHub, and specify structured details about the vulnerability. Then, when they're ready to publish the Security Advisory, GitHub will send security alerts to all affected projects."

Anyway, this just sounds like a huge step forward for the open source community in general and for the security aspect of it specifically. He notes that receiving a notification about vulnerable dependencies is helpful, but getting a pull request with a fix is even better. So to help developers respond quickly to new vulnerabilities, GitHub creates automated security updates, which will be pull requests that update a vulnerable dependency to a fixed version automatically. So automated security updates were launched in beta at GitHub Satellite 2019 and are now generally available and rolled out to every active repository with security alerts enabled.

And then finally he said: "We've made all of the data that maintainers create in GitHub Security Advisories, plus additional data curated and mapped to packages tracked by the GitHub dependency graph, available for free. You're able to explore the GitHub Advisory Database in your browser, link directly to records with CVE identifiers in comments, or access the data programmatically using the Security Advisory API."

So, for example, github.com/advisories is where you would go with a browser if you wanted to browse around and see what was going on. And I have the link in the show notes to the API endpoint, if you wanted to be able to query it through your own automation. So essentially what this looks like is we're bringing to what has traditionally been sort of a very ad hoc, you know, you could use the term "loosey-goosey" sort of process, you know, the kind of structure and automation which various corporate entities have had to bring to their own processes, but which had been lacking from the open source effort.

So I think this is just 100% good because it really seems clear to me. You know, here's Microsoft scrapping years of development effort on a complete rewrite of their web browser. They're like, eh. We're just going to use Chromium. It makes more sense to do that. So, and Leo, you were just saying, Linux is ready for the desktop. Well, all of this is open source. All of the pieces of it are open source.

So we need the same sort of focus on security moving forward that we're used to receiving from Apple and from Microsoft and other commercial vendors in whose own interest keeping their product secure is. There hasn't really been the same sort of economic drive, which is why it's been necessary to do fundraising campaigns to produce the money to then commission audits of important open source packages. This is the next step in that maturity. And so I think it's just - it's wonderful.

Speaking of wonderful, we've talked in the past about a number of different, really unfortunate instances of people being harassed at the border, and this whole issue of do you use a biometric, or do you use a password, when the border guards challenge you and demand to have access to your smartphone or laptop or whatever.

There was the case of a natural born U.S. citizen, Sidd Bikkannavar, who was at the time, and probably still is because it's only two years ago, a NASA engineer who was detained by U.S. Customs and Border Protection in 2017, pressured to hand over his NASA-issued phone and PIN so that the Border Protection agents could get into it. This was for no cause and in spite of the fact that the work-issued phone could have contained sensitive information relating to his employment at the space agency, and in spite of the fact that NASA employees are obligated to protect all work-related information. The CBP officer returned his phone half an hour later, saying it had been searched using "algorithms." There was another instance of...

Leo: Oh, well, those algorithms, you know how they are.

Steve: Yeah. And I was thinking, well, what is that? Is that supposed to mean that no people...

Leo: Black magic.

Steve: ...people looked at it, you know, so we weren't looking at your photos, but algorithms were? I don't know.

Leo: We don't know.

Steve: There was an artist, Aaron Gach, who - he's another natural born U.S. citizen who was forced to unlock his phone after returning from putting on a gallery installation in Brussels. That particular installation focused on "mass incarceration, government control, and political dissent." Is that why he had to turn his device over? We don't know. But he did, for no cause.

Diane Maye, a college professor and retired U.S. Air Force officer, was detained for two hours at Miami International Airport upon returning home from a vacation in Europe. At the time that the lawsuit was filed in 2017, Maye said that the encounter left her feeling "humiliated and violated." She explained that she worried that border officers, again without cause or reason, would read her email messages and texts, look through her

photos and so forth. She said that this was her life, and a border officer held it in the palm of his hand. She joined the lawsuit, which was brought by the ACLU and the EFF, because she strongly believed the government should not have the unfettered power to invade our privacy without probable cause.

Okay. So since this podcast has been underway, because it goes back to 2009, so for the past 10 years, that's when U.S. border agents obtained the right to legally search electronic devices of travelers at borders without any specific cause or suspicion, in the interest of protecting our borders and enforcing U.S. border security. Then in 2016, three years ago, that law was revised to require "reasonable suspicion," unquote, for anything beyond basic searches. But that still allowed agents to require travelers to unlock phones and gave them free rein to read messages and other basic information. Again, without a probable cause.

So what is significant is that last Tuesday a federal court in Boston ruled that suspicion-free warrantless searches of travelers' electronic devices at U.S. border entry points are unconstitutional. The decision comes from a lawsuit, which was Alasaad v. McAleenan, filed against the Department of Homeland Security in 2017 by the ACLU, the American Civil Liberties Union, and the EFF, on behalf of 11 travelers. Ten of them were legal residents of the U.S., and one was a lawful permanent resident, all of whom were forced into warrantless searches of their mobile phones and laptops at the border, including the three that I just mentioned.

Sophia Cope, a senior staff attorney with the EFF, the Electronic Frontier Foundation, said of this ruling: "This is a great day for travelers who can now cross the international border without fear that the government will, in any absence of suspicion, ransack the extraordinarily sensitive information we all now carry in our electronic devices." And reading into this a little bit more, and doing some digging, I noted that it does remain legal for border agents to look through the devices of travelers who get referred for a secondary inspection. So during the primary inspection, travel documents and passports are reviewed. That's what most of us experience. I'm sure you just did, Leo, and I just did.

Leo: Oh, yeah. Oh, yeah.

Steve: Coming back into the U.S., where they just asked, like, where have you been, why were you gone, and so forth.

Leo: What did you buy?

Steve: Yeah. And then just kind of wave you through.

Leo: Right.

Steve: If a secondary inspection is deemed to be needed, then it's the case that officers may still search phones, thumb drives, computers, and other electronic devices to determine whether they should let somebody into the country or to identify potential legal violations. So having read that, I'm not quite sure what everyone's jumping up and down about because it looks like you could still be pulled off to the side and given that. But, you know, this does seem like it's movement in the right direction.

According to the ACLU, the Boston district court's order puts an end to the authority that CBP and ICE had been granting themselves to search and seize travelers' devices for purposes beyond enforcing immigration and customs laws, which was the umbrella under which they were doing that. At this point, border officers are required to demonstrate "individualized suspicion of contraband" before they can search a traveler's device. So that does sound like they've raised the bar to a useful level.

And when I think back about 10 years ago, or before the era of smartphones, when you think about it, Leo, we used to just, before smartphones, we didn't have something that did contain a virtual record of our lives. You know, if you're using Instagram, as I was when I was traveling, in order to sort of create a little travelogue for my friends and family who were here, you know, that's recording everything we do. All of my text messages, I mean, they're boring, in my case, about when I'll be home for dinner. But it is, I mean, if you read through the contents of someone's phone, you pretty much know a huge amount about them. So the fact that we're now carrying a record of our life, why does that suddenly mean that border entry agents suddenly should have a right to what amounts to an electronic frisking? And so it's looking like maybe we're going to, you know, we will be putting an end to that. So that seems like a good thing.

Leo: Yes.

Steve: I did want everyone to know that WhatsApp has been found to have a problem again. And just so we don't consider these problems to be just another in a long line of theoretical vulnerabilities, which we are often talking about here, let's remember that Israel's NSO Group was recently found to have leveraged another recent WhatsApp vulnerability, this case in the VoIP calling, to successfully install Pegasus spyware on nearly 1,400 targeted Android and iOS devices worldwide. So remote code execution flaws, especially in WhatsApp, which now is able to claim the title of the world's number one supposedly secure messaging app...

Leo: Oh, everybody uses it. Everywhere outside the world uses it. It's universal.

Steve: So, you know, it won't go unexploited for long because of its high use. So in October, a few months ago, we noted that WhatsApp was using an open source GIF image-parsing library which contained a double-free memory corruption bug, which could be leveraged for remote code execution in the context of the WhatsApp application. So there was another instance of a problem. That's two. Now we have the third one recently. This is just this last month, WhatsApp quietly patched another critical vulnerability that would, had it been used, and we don't know one way or the other, attackers to remotely compromise targeted devices to steal secure chat messages and the files stored on the devices, or whatever else they wanted.

The vulnerability, which was tracked as CVE-2019-11931, is a stack-based buffer overflow which resided in the way WhatsApp's parsing of the MP4 video stream metadata was being done, which would either crash WhatsApp or allow remote code execution attacks. Of course we've often seen parsing stream metadata, especially for a compressed file format, is very difficult to get absolutely correct and to make robust. It's so easy when decoding metadata to make the assumption that the sender was a valid codec rather than something malicious.

So to remotely exploit this vulnerability, all an attacker needs is the phone number of a targeted user and then to send them a maliciously crafted MP4 over WhatsApp. It can then silently install a malicious backdoor or other spyware app on what is now a

compromised device. The vulnerability affects both consumers and enterprise apps on all major platforms. So this was very widespread, both on Android, on iOS, and on Windows. So this was patched.

So the takeaway is update your instance of WhatsApp. Make sure that you're running the most recent version. I won't enumerate them. There's six of them that I have in the show notes, which are the ones which were affected. And in every case it was versions before, up to, and including the one that I've listed. So just make sure that you're current.

The attacks would have been targeted. If you're somebody who might be a target of an attack, and if you can recall in the last few months having received an unsolicited WhatsApp message that crashed your WhatsApp, or tried to play a video and didn't, then you might want to dig a little further and see whether anything might have crawled into your device. Again, it's unlikely that anyone would have been affected. Facebook has claimed that no one was a victim of this. But how would they know...

Leo: No one, no one's a victim. No one.

Steve: ...if somebody was, if it were a targeted attack, yeah.

Leo: Oh, absolutely, no one.

Steve: And of course Facebook's reputation continues to decline. So, yeah, just trust us, trust us.

Leo: It's probably, though, a targeted attack because they'd have to have your phone number.

Steve: Yes, yes, yes.

Leo: They have to know where they're going.

Steve: And so, you know, it's not something that would be sprayed. Well, if it had been sprayed, then lots of people would have been affected by it. And so Facebook would not be able to claim, oh, no, nobody we know.

Leo: Nobody we know. That's a better way to put it. Nobody we know.

Steve: That's right. So we had fun, Leo, months ago, with ZombieLoad.

Leo: Oh, yeah.

Steve: Just saying it, saying it was fun. And I won't spend much time on this. I'll just note that there is version 2 of ZombieLoad. And like all of these, these have ended up, I

mean, and we thought this from the beginning. They were interesting from an intellectual standpoint. It's been two years now. When we get to next January, it will be two years. The first podcast that we did in 2018 was Spectre and Meltdown.

Leo: Yeah. It doesn't seem like it's been that long. But geez.

Steve: I know.

Leo: It's almost two years.

Steve: I know, two years. And significantly, in all that time, as far as anyone knows, none of this stuff has ever been exploited. So yes, it's a problem in theory. If you were on a shared hosted VM in a cloud environment, if something was able to arrange to share the same hardware processor as you, they could reach across the VM boundary and maybe steal some information. That's what got everybody up in arms and worried.

And as a consequence we had all these firmware updates. We've had Windows now patching the firmware on the fly for some versions of Windows. I mean, you know, basically two years of running around in circles because of Meltdown and Spectre and Foreshadow and Fallout and RIDL, you know, and we've got the so-called MDS, the Microarchitectural Data Sampling flaws. Intel continues to be dogged by this. I've seen reports of as much as 40% performance hit when all of this stuff is turned off. And why? Basically for what is nothing, it really has never been shown to be more than a theoretical attack.

Intel believed, and this is what's significant about ZombieLoad v2, they believed and claimed that their most recent architecture, the Cascade Lake architecture, released just in April, was completely protected against side-channel and speculative execution attacks in the hardware. It is victim of ZombieLoad 2. It turns out that this particular type of attack was always known by the attackers. It wasn't until Intel produced a patch for it, which they have just made available, that the researchers behind this updated their original report in order to also disclose ZombieLoad v2.

Microsoft wrote: "On November 12, 2019, Intel published a technical advisory around Intel Processor Machine Check Error vulnerability" - that's the official name - "that is assigned CVE" - get this - "2018-12207." Meaning, uh-huh, they've known about it for quite a while. I'm about to sneeze, I think. Nope, maybe not.

Leo: I hate that.

Steve: "Microsoft has released updates to help mitigate this vulnerability for guest virtual machines, but the protection is disabled by default. Enabling this protection requires an action on the Hyper-V hosts running untrusted VMs. Follow the guidance in the Registry Settings section to enable this protection on the Hyper-V hosts running untrusted VMs."

Anyway, so I didn't even put it in the show notes. Somebody for whom this could have some effect can certainly track it down. Given that not a single instance of any one of these low-yield theoretical attacks has ever made a usefully practical appearance in the wild, that's never been seen, and given that the only true vulnerability is in a shared hosting environment where a bad guy could arrange to co-reside on the same hardware, this is just not a concern for any end user, who certainly make up the bulk of this

podcast's audience. If something gets into your computer, you're already hosed. I mean, Windows uses this messaging system which is incredibly insecure, and so a random app can query the global clipboard that all apps share. So the instant you use it to copy a password that you've just created in order to paste it somewhere else, you're owned.

So anyway, as we know, personal workstations are not secure. It's certainly the case that cloud environments want more security, need more security, and Intel's just going to have to continue to figure out a way to provide the kind of isolation that we need among processes without having to roll back the clock and give up on a decade of performance benefit because unfortunately the tricks that they were using were leaving, as we know, a footprint behind in the so-called microarchitecture of these chips that clever researchers were able to leverage into theoretical cross-process boundary information leakage. Some of it worse than others, but none of it ever actually happened in practice.

And I think the last piece of news - is it the last piece of news before - yeah, it is, because we touch on a little bit of fun miscellany, Leo. And this is important, I think, the announcement of something known as the Bytecode Alliance. Mozilla, Fastly, Intel, and Red Hat are the cofounding launch members. And the short take of this is WebAssembly is outgrowing its browser. So as we know, and we've covered through the years, there's been some early competition among competing JavaScript replacements, with the general goal of improving client-side web application performance, come up with a way of making this stuff faster. The winner is WebAssembly.

So the good news is we're not going to have some fractured environment where there are multiple solutions to the same problem. WebAssembly is the standard, and it's now supported by all major web browsers. And as it's continuing to mature, it has proven to be the right choice. So the Bytecode Alliance is the next effort to generalize WebAssembly into a generic platform and to make it architecture-agnostic, creating a common runtime suitable for use on everything from extremely lean little IoT widgets through high-end CDN server farms. In other words, completely outside the browser.

And if this sounds familiar, it's sort of the role that Sun had always hoped that Java might obtain; right? Java, as we know, compiles to Java bytecode, and then you have the JVM, the Java Virtual Machine, as a runtime environment and interpreter of the bytecode. What's different from Java is that WebAssembly has as its Internet-facing origin the web browser, where strong attack hardening had to be explicit from the beginning. So for WebAssembly, security was built in from the start. And I think that's what gives us hope, actually.

Lin Clark, who was writing from Mozilla, explained it this way. He said: "We have a vision of a WebAssembly ecosystem that is secure by default, fixing cracks in today's software foundations. And based on advances rapidly emerging in the WebAssembly community, we believe we can make this vision real. We're already putting these solutions to work on real world problems, and those are moving towards production. But as an alliance, we're aiming for something even bigger. As an industry, we're putting our users at risk more and more every day. We're building massively modular applications, where 80% of the code base comes from package registries like npm, PyPI, and crates.io. Making use of these flourishing ecosystems isn't bad; it's good.

"The problem is" - and we talk about instances of this all the time on the podcast - "the problem is that current software architectures are not built to make this safe. And bad guys are taking advantage of that at a dramatically increasing rate. What the bad guys are exploiting is that we've gotten our users to trust us. When the user starts up your application, it's like the user's giving your code the keys to their house. They're saying, 'I trust you.'

"But then you invite all your dependencies, giving each one of them that full set of keys to the house. These dependencies are written by people you don't know and have no reason to trust. As a community," he writes, "we have a choice. The WebAssembly ecosystem could provide a solution here, at least if we choose to design it in a way that's secure by default. If we don't, WebAssembly could make the problem even worse. As the WebAssembly ecosystem grows, we need to solve this problem. And it's a problem that's too big to solve alone.

"That's where the alliance comes in. The Bytecode Alliance is a group of companies and individuals coming together to form an industry partnership. Together, we're putting in solid, secure foundations that can make it safe to use untrusted code, no matter where you're running it, whether on the cloud, natively on someone's desktop, or even on a tiny IoT device. With this, developers can be as productive as they are today, using open source in the same way, but without putting their users at risk." He finishes: "This common, reusable set of foundations can then be used on their own, or embedded in other libraries and applications."

So, I mean, I couldn't be more excited about the potential. That's all it is at this point. But they have - I think what happened is they recognized right now they have an opportunity. WebAssembly is demonstrating to run very fast. And due to the way it's been built, and because it had to be Internet-facing in the highest attack surface we have in the world, the web browser, and it is surviving, they said, okay, let's generalize it. Let's broaden its application, but let's hold onto what we have at this point, which is this secure-by-default approach.

And of course how many times have our listeners heard me bemoan the way code is still being created today? It is the Wild West. It's buggy automatically because it is so difficult to create really secure code. I mean, it's just near to impossible. And so we'll just cross our fingers and keep our eye on this effort, the Bytecode Alliance, and see maybe if this time we can do it. You know, Leo, I think it's been a matter of power. We haven't had the processing power to spare because we're always pushing it to the limit. We haven't had the bandwidth to spare and the storage space to spare. But we've got gobs of all that now.

Leo: Yup.

Steve: So if we can just restrain ourselves from writing everything in C and say, wait, let's finally prioritize security, let's make that first, even if there's a little bit of cost in performance, isn't it worth it, now that we have more performance, finally have more than we need?

Leo: Yeah, yeah. I think WebAssem is amazing. But, you know, you've got to be aware of the potential risks, obviously.

Steve: Yeah, yeah. And I'm glad these guys are going to take it seriously.

Leo: All right, Steve. TPM.

Steve: Well, not quite. As our listeners know, I reserve the right from time to time to share some things that I'm enthused about.

Leo: Oh, I always look forward to this, actually.

Steve: Sometimes it's sci-fi books. Actually most of the time it's sci-fi books.

Leo: The new Peter Hamilton Salvation Part 2. Have you read it? Are you reading it?

Steve: Oh, I know. No. I'm rereading the first book because it's been several years.

Leo: This is why I'm going to wait for Book 3.

Steve: You're right, you're right. And he always does this to us.

Leo: I know.

Steve: Remember "Pandora's Star," and how we were like, just dying?

Leo: Oh, it was painful because he wrote one, and then we had to wait a year or two for the next one. Oh, it was painful.

Steve: But I really like his writing. And John's reread the first one.

Leo: John's raving over it.

Steve: Yeah, and he says...

Leo: I trust him and you, yeah.

Steve: Yeah, he's been saying that the second one is like really, really good.

Leo: Oh, I can't wait.

Steve: So, okay. So two pieces of just random feedback. The second one, I couldn't wait for the podcast. The first one, I was disappointed. And I am fully prepared to have a minority opinion on this. I think I probably do based on its IMDB rating. But this is "The Mandalarian."

Leo: "The Mandalorian," yeah.

Steve: "Mandalorian," right. Oh, yeah, I misspelled it here. Yeah, "Mandalorian." I signed up for Disney+ on my Roku so that I could watch it. Lorrie and I sat down with some

excitement and anticipation. Lord knows the trailers look just fantastic. And about 50 minutes in I kind of was thinking, okay, this is a kiddie movie.

Leo: Oh.

Steve: I mean, something about it, I don't know what it is, but it just didn't have the serious gravitas of "Star Wars" at all. Okay, now, yes, Ewoks were a problem. I recognize that.

Leo: Oh, no. Oh, no. Oh, no.

Steve: But still, it just didn't do it for me. So Lorrie refuses to watch the second one, even though it's only 30 minutes.

Leo: Really.

Steve: I've heard it's action packed.

Leo: I like to give a show three episodes, at least.

Steve: Yeah. So anyway...

Leo: Because they often get better. The first one's the pilot.

Steve: Right. And they will be released on Fridays from now on. The first one came out, then the second one came out like a couple days later. So maybe I'll watch the first three. Anyway, I will be doing so solo. Although I should tell you...

Leo: Wow. Wow.

Steve: ...it's not because - it's just, well, just because she's got other things to do. I mean, she really did dislike it. She just thought, oh, this is really dumb. But we could not wait for, and we were not disappointed by, the movie we saw on opening night, which I rarely do, actually, last Friday. We'd been waiting for it for a while. And the reason I wanted to just mention it to our listeners is that it is very rare that I consider a movie to be perfect. I mean, it can be about anything. It doesn't have to be sci-fi. It's possible to have a non-sci-fi perfect movie. And I think "Ford v Ferrari" was a perfect movie.

Leo: Now, this is a little shocker. Now, it's funny because I just read an article, I think it was in The New York Times, saying it was the box office winner, even though there were movies everybody thought would easily best it. It made $30 million in the first weekend. So you're not alone. There's something about this movie.

Steve: You'll know when you see it. I mean, it is, I mean...

Leo: Christian Bale I love. Matt Damon...

Steve: Yeah. You're rooting for, like, for what goes on. There's maybe a little bit of nationalism because, you know, after all, "Ford v Ferrari." And so, you know, I'm a patriot.

Leo: And it's a true story. This was the story of Ford trying to design a racecar. It was going to acquire Ferrari, and it failed; right?

Steve: Well, yeah, in fact Ferrari was a little foxy. They used Ford's bid in order to get a better deal from Fiat.

Leo: Oh, that pissed Ford off.

Steve: And also insulted Ford, talking about, I mean, like, "Go back to your ugly factory and make your ugly little cars."

Leo: Ooh.

Steve: And so Ford Jr., Ford II, said, okay. Anyway, I've said enough. If any listeners have found my opinion to be useful in the past and think, I mean, and don't have some reason for thinking, oh, I'm never going to watch that, I will just say I don't think you will find your time to have been wasted. It was really...

Leo: Oh, interesting. And I don't think you're a gear head. I don't think you're a gear head, so...

Steve: Well, I was the top of my class at Bob Bondurant's Sears Point Racing.

Leo: Oh, you are a race driver. You're kidding.

Steve: And I very much liked my two - I had Fiat X1/9s because they were a mid-engine car and just handled. They were absolutely neutral steering beautiful little sports cars. So, yeah, I do love - I love to drive.

Leo: Oh, that's awesome.

Steve: But you don't have to. This isn't about that. This is about people and gumption and spirit and courage and, you know. And at one point I whispered to Lorrie, I leaned over, I said, "This is about art." I mean, it's just - it's like it's the pursuit...

Leo: The art of design and pursuit of perfection. And I understand...

Steve: The pursuit of perfection, yeah.

Leo: There isn't as much driving as you might think in a movie called "Ford v Ferrari," which doesn't bother me. But Christian Bale in the cockpit, it's pretty gripping, I hear. It's pretty amazing. I am dying to see it now.

Steve: Yeah. Don't miss it, Leo. I know you will love it. And I bet - I just wanted to make sure that it got onto our listeners' radar because, even if they don't watch it for two years until it streams on something the size of a walnut, it doesn't matter. Still. Although you'd really kind of miss it if you were looking at it on a small screen. But then again, we all have big screens at home, pretty much, now.

Leo: It was the Washington Post that was writing about it. And what they're saying is that, in this era of streaming, they really didn't expect a movie like "Ford v Ferrari" to do very well in the theaters because...

Steve: It's easy to wait.

Leo: It's easy to wait. The headline was "Streaming was supposed to kill original theatric movies. Don't tell 'Ford v Ferrari.'" Thirty-one million in ticket sales - beating "Charlie's Angels," which only had nine million.

Steve: Well, okay.

Leo: But, you know, you'd expect - that's a movie you'd expect people to go see in the theater. Apparently it's terrible, so.

Steve: I am not surprised.

Leo: Yeah, yeah. Come on. A reboot of "Charlie's Angels"? What's not to love?

Steve: Oh, I definitely loved the first series, the original one.

Leo: Yeah, with Farrah, yeah.

Steve: That was a lot of fun. But I was four at that time, so...

Leo: What did you know? What did you know? So anyway, you're not alone. The movie is doing quite well, which is - I think it's exciting.

Steve: I'm glad to know it. And we know that production companies look at the box office to decide what scripts they're going to pick up and what they're going to do.

Leo: That's right.

Steve: So it just...

Leo: If smart movies are back, I would love to see that.

Steve: And Lorrie is a gear head, so she was, I mean, she was down for this thing from the beginning. The first time we saw a preview when we were seeing something else months ago, we all three of us, because we also went with my best friend, and it's like, okay, that's it. Definitely going to see that.

Leo: Oh, how exciting.

Steve: And it was better, I mean, again, it was a perfect movie. It was perfect.

Leo: Great.

Steve: So what's not perfect, it turns out...

Leo: Uh-oh.

Steve: ...is a few implementations of something we need to trust, as you noted, the so-called Trusted Platform Module, the TPM. And I got a kick out of this. It has its own site, tpm.fail. And when I realized that there was now a top level domain .fail, that's going to clearly skew the naming patterns of all subsequent vulnerabilities.

Leo: Oh, yeah. Oh, yeah.

Steve: There'll be, you know, we.fail; you.fail; he, she, and we.fail. It's going to be...

Leo: It's also doing a brisk business in companies like twit.fail, buying it proactively; right? Leo.fail. I'm going to go out and buy those all right now.

Steve: I wouldn't be surprised if it's expensive. I know that .sucks is expensive.

Leo: Same thing; right?

Steve: So I wouldn't be surprised if .fail is expensive. But anyway, so we have tpm.fail. The best summary, I'll just read from their short abstract at the beginning of their paper. They said: "Trusted Platform Module serves as a hardware-based root of trust that protects cryptographic keys from privileged system and physical adversaries. In this work, we perform a black-box timing analysis" - and so all of our listeners who've been paying attention through the years know what this is about. It means there's a side-channel attack because it's non-constant time based on its secrets.

"We perform a black-box timing analysis of TPM 2.0 devices" - and, by the way, that's where we are, that's the standard currently - "deployed on commodity computers. Our analysis reveals that some of these devices feature secret-dependent execution times during signature generation based on elliptic curves. In particular, we discovered timing leakage on an Intel firmware-based TPM, as well as a hardware TPM. We show how this information allows an attacker to apply lattice techniques to recover 256-bit private keys for Elliptic Curve Digital Signature Algorithm (ECDSA) and EC-Schnorr (SCHNORR) signatures." Yeah, it is funny.

"On Intel fTPM" - that's their firmware TPM - "our key" - and this is scary - "recovery succeeds after about 1,300 observations in less than two minutes. Similarly, we extract the private ECDSA key from a hardware TPM manufactured by STMicroelectronics, which is certified at Common Criteria EAL 4+ level" - meaning that's the best you can get - "after fewer than 40,000 observations." That suggests that the timing difference is subtler there, but they were able to see a skew and use it.

They said: "We further highlight the impact of these vulnerabilities by demonstrating a remote attack against a strongSwan IPSec VPN that uses a TPM to generate the digital signatures for authentication. In this attack, the remote client recovers the server's private authentication key by timing only 45,000 authentication handshakes via a network connection." They said: "The vulnerabilities we have uncovered emphasize the difficulty of correctly implementing known constant-time techniques, and show the importance of evolutionary testing and transparent evaluation of cryptographic implementations. Even certified devices that claim resistance against attacks require additional scrutiny by the community and industry, as we learn more about these attacks."

So in the first place, unlike all of the previous Intel CPU leakage attacks that we were just talking about, these are practical to accomplish. And as they demonstrated, they can be successfully performed remotely over a relatively high-speed, thus low-jitter, network. And what I thought was interesting, in their discussion in their research paper they mentioned that Intel was initially not convinced of the attack's severity and gave it a rather low severity score.

Leo: As usual.

Steve: So the researchers then demonstrated a successful attack over the network, meaning that it was a remote key recovery attack which really got Intel's attention. Intel jacked the severity score up and then got busy fixing it. The good news is this is a firmware implementation of the Trusted Platform Module within what Intel calls their Platform Trust Technology (PTT). The bad news is that it's the easiest of the attacks to exploit.

The good news is that being implemented in firmware, it can be patched and updated, and Intel has last Tuesday released patches for the various chip firmware that supports PTT. And it goes way back, I think at least to 2015, maybe earlier. So on systems where that's the way the Trusted Platform Module, the TPM, is being used, it could be fixed and

patched and then - but of course that requires some means of updating the firmware; Intel releasing the patch. Unless Windows incorporates it into a security update, it would be incumbent upon the BIOS manufacturers of the affected motherboards to update their BIOSes.

And we know that there are companies that are maintaining their BIOSes going way back, like Dell has been very good at that. And I think HP is, too. So you'll want to be looking for a BIOS update for motherboards in the next couple months, and probably use something from Intel in order to determine whether you're vulnerable. Oh, actually there is going to be - the researchers are going to be making available for open source a tool to test it. So we'll see if Microsoft is going to patch this on the fly. Linux is normally really good at doing this. As we know, for all of the Spectre and Meltdown patches, shortly after Intel released the firmware, Linux had incorporated it into their boot. So I think Linux-based systems, and for that matter servers that are running Linux, will be able to get themselves secured probably sooner than anybody else.

They did indicate that they, as I mentioned, that they intend to publish tools they used to analyze the vulnerable TPMs, along with proof-of-concept code on GitHub. Of course, this will be a mixed blessing. It will allow system admins to determine which TPMs they are using on their probably many various systems to determine whether they may be vulnerable. But the proof of concept code, which should allow them to detect vulnerability, could also allow those devices to be attacked because the proof of concept would be able to be reverse engineered because it'll all be open source. So as we know, that would mean that bad guys could take advantage of it.

On the hardware side, there are many manufacturers of Trusted Platform Modules. The researchers tested all that they could find. And, for example, those from Infineon and Nuvoton were found not to have secret-based timing influences. But the very popular TPM by STMicroelectronics, those guys were found to alter its timing as a function of the secret it was manipulating. So it's not operating in constant time. The good news is the timing variations are so slight that many, many more tests were required. On the other hand, they can be performed very quickly on a local system. However, the network packet jitter inherent in any network communications renders this not attackable remotely.

So the good news there is it's not a remote attack. The bad news is it's in the hardware. So there's no fixing it. I did not dig in enough to see whether it might be possible to stop using an STMicro-based TPM on a motherboard that has it and switch over to using the Intel PTT solution if it's available. That might be an alternative, to use the patched Intel firmware-based solution, rather than the TPM that's on the motherboard. I don't know either way whether that would be effective. But it is important to note that it cannot be used remotely. So STMicro has updated their device to fix the problem; but for all those many, many, many, many millions of devices that are out there, there is a vulnerability.

Now, the one good piece of news is that this was in the Elliptic Curve Digital Signing Algorithm, which I'm sure I read somewhere, I don't have it in my notes, that Microsoft indicated they don't use. So Microsoft's normal use of the TPM does not invoke ECDSA. On the other hand, third parties can be storing their keys in the TPM, like the VPN that I noted earlier, which does make them vulnerable, if not the native operating system.

So that suggests that, for example, BitLocker, which relies on TPM to store its master secret, is probably safe from this. And that's probably the highest use of the Trusted Platform Module that any Windows users have is just using it to secure BitLocker. When I was traveling for the SQRL Tour a couple months ago, I encrypted my hard drive and used BitLocker in order to protect it and relied on the TPM to hide and keep its secrets secret.

So this is not a huge problem for Windows users. Probably a larger concern in an environment sort of like the Intel, the previous Intel vulnerabilities, where you might have shared hardware, and so you might have something trying to get at the secrets that are stored globally in the TPM hardware. But at least Intel has a patch ready. It's because the patch is ready that we are now being informed of this problem. And maybe it's possible to switch to a software-based solution, if you happen to have the hardware-based solution on your motherboard. So TPM-Fail.

Leo: Yeah. Oh, well. It doesn't, you know, I'm glad it doesn't sound like it's as bad as I was worried.

Steve: Yup. I'm thinking it's not.

Leo: Yeah.

Steve: So I wanted to put it into context for our listeners.

Leo: Thank you, yes. As always, the headlines are scary. But once we get down to it - that's why we count on you, Steve, to give us the inside scoop. You'll find Steve at GRC.com. That's where his most fabulous program lives, his bread and butter, SpinRite, the world's greatest hard drive maintenance and recovery utility, GRC.com. And I'm happy to say that soon work will commence, if it hasn't already - has it?

Steve: Work is commencing shortly. I have a few last bits to iron out. They're literally last bits, some status bit dialogue we've been having. And then I need to update the documentation. We had someone come into the group with a very critical eye and was very useful to the project because he attacked SQRL.

Leo: Good, good.

Steve: And from the attack we got a lot of good, useful feedback, mostly to - what it turned out is that I had done things in my implementation that I hadn't fully articulated in the documentation.

Leo: Of course.

Steve: So I will explain more clearly what it was, you know, how to do these things correctly where it is critical for them to be correct. And then it's on to SpinRite 6.1. And I am very excited.

Leo: Very excited. And we should mention that the SQRL, of course the unveiling has already occurred, but we're going to do our special November 30th, a Saturday afternoon, about 2:00 p.m. Pacific. Are we, yeah, I guess we're streaming it. So be a little after 2:00 p.m. Pacific, 5:00 p.m. Eastern time. That's 22:00 UTC on TWiT.tv/live. We will of course also package it up for distribution so you can all

watch it at your leisure. And we had mentioned that we were going to have a live studio audience. The response has been phenomenal. We are full.

However, we are creating a waiting list. So if people have tickets to see the event in Petaluma on the 30th and won't make it, let us know so we can release your tickets to another worthy person. It's no shame, of course. If you're not going to be able to make it, that's great. The shame would be not telling us because there are many people who'd like to go. I should also say, for those of you who will be there, we're going to do an after-party little event at the Lagunitas Brewery next door. We've got a room, and we're all going to be going over.

Steve: Gonna hang out.

Leo: It's going to be a lot of fun, Steve and I and all of you. And that's always been fun. We had such a good time in Boston. I'm looking forward to that. So November 30th. Don't forget to get your copy of SpinRite. If they buy a copy today, Steve, do they automatically get 6.1?

Steve: Oh, yeah, yeah, yeah, yeah, yeah, yeah, yeah. Everybody who has 6.0 and anybody who buys it before gets 6.1. And in fact, owners of 6.0 will be able to start playing with it way before it's released.

Leo: Oh, nice.

Steve: Very much like SQRL that has been working about a year and a half while all the final pieces got put together. That's the way we were doing it before. So owners will be able to use their serial number to look up their what we call the transaction code, and they'll be able to use that in order to download the versions well before its official launch.

Leo: Nice, very nice.

Steve: So there will be some early benefit to those early adopters.

Leo: Steve, while you're at the website, Steve has lots of great stuff. We mentioned the Healthy Sleep Formula. That's there. We also mentioned, well, we didn't, but we should have, ShieldsUP! and all the other free tools. They're all there, as well as SpinRite. So it's a great site to get to and browse around: GRC.com. This show is there, too. He has the only 16Kb versions of the show, as well as the 64Kb audio, and he has the only transcripts of the show. So if you like to read along while you're listening, that's a great place to go, GRC.com.

We have audio and video at our website, TWiT.tv/sn. Of course we always recommend you subscribe in your favorite podcast application. That way you'll get it automatically, the minute it's available. It helps us, too, because a lot of these apps note how many people subscribe and then in their discovery tab will show the podcast, and it helps us. So please subscribe in your favorite podcast application, if you haven't done so already. Steve, thanks so much. Have a great week.

Steve: My friend, talk to you next week.

Leo: See you next Tuesday on Security Now!.

Steve: Bye.