SECURITY NOW!

Transcript of Episode #662

Spectre - NextGen

Description: This week we begin by updating the status of several ongoing security stories: Russia vs. Telegram, Drupalgeddon2, and the return of Rowhammer. We will conclude with MAJOR new bad news related to Spectre. We also have a new cryptomalware, Twitter's in-the-clear passwords mistake, new Android "P" security features, a crazy service for GDPR compliance, Firefox's sponsored content plan, another million routers being attacked, more deliberately compromised JavaScript found in the wild, a new Microsoft Meltdown mistake, a comprehensive Windows command reference, and signs of future encrypted Twitter DMs.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-662.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-662-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here, and I'm back. Hello. Thanks to Robert Ballecer for filling in for the last couple of weeks. I came back just in the nick of time. Turns out Spectre's back, baby. Steve has the details next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 662, recorded Tuesday, May 8th, 2018: Spectre NextGen.

It's time for Security Now!, the show where we cover the latest in security and privacy with this guy right here, this cat, Mr. Steven Gibson of the GRC Corporation. Hi, Steven.

Steve Gibson: It's true, Leo. Welcome back. You were missed.

Leo: Nice, nice. I missed you guys. And thanks to Robert, Father Robert Ballecer, the Digital Jesuit, for filling in.

Steve: And I heard you discussing on This Week in Google, which just preceded this podcast as a consequence of the change of schedule, thanks to Google I/O, that you were mentioning Azure Sphere, which actually was the podcast topic week before last. So indeed, as you would expect, we covered it in great detail.

Leo: And do you concur that it is a good idea?

Steve: Yes. Absolute win, for exactly the reasons that you reiterated on This Week in Google, the idea that...

Leo: Cheap Chinese manufacturers can actually make something secure.

Steve: Yes. And all they want to do is pump out the silicon and sell consumer products. They don't want the downstream responsibility that comes from the fact that they're actually shipping Internet-connected computers.

Leo: I don't know all the details. But is Sphere cheap enough that, you know, somebody makes a cheap camera would still use it?

Steve: That is the question, whether the manufacturer decides that it is so inexpensive that it's better for reputation damage, I mean, really...

Leo: But, yeah, I'm sure there'll be a little icon they can put on the box or something like that.

Steve: Yeah. And so the licensing of the chip, it's open licensed chip. It's open architecture. It's open source for the Linux. So one thing that isn't is you do have to tie back to Microsoft Azure Cloud to get the monitoring and maintenance. So we just don't know yet what the pricing is going to be. But fingers are crossed.

Leo: Good, good, good.

Steve: Speaking of fingers being crossed, today's - I can't believe this. Well, yes, I can. Spectre NextGen.

Leo: What?

Steve: Yes.

Leo: More?

Steve: Yes. Eight new problems. Intel is scrambling once again. One of them is way worse than Spectre ever was. And I wish I were more surprised. But it really was the case that for decades we were blissfully, willingly not looking at the fact that an architecture that we were sharing was not secure in a shared environment. And as we've said, any time there is the residual effects of running code on a processor that then switches to a different environment, it's possible to very cleverly, I mean, these are not easy things to do. But you can figure out how the processor was changed by what it was just previously doing. And that leaks information.

So anyway, that's the topic of the podcast. We'll get into it in more detail at the end of the show. But we've also got a bunch of other stuff. We've got first sort of like continuing on monitoring some stories that we've been covering. Russia vs. Telegram, there's new activity there. Drupalgeddon2, new activity there. Including, you're going to love this, a Google spreadsheet of the 350-plus websites that have been now compromised. In fact, I made that the bit.ly shortcut link for the show. Anyone who's interested, bit.ly/SN-662, and that will take you to the rather distressing Google spreadsheet.

We've got, believe it or not, the return, better than ever, of Rowhammer that affects Android smartphones. We have major new bad news, as I said, related to Spectre. Also new cryptomalware. Twitter's mistake with their passwords, which has gotten a lot of coverage. We have the new Android P security features. And I don't think we heard, did we, what P stands for?

Leo: No, no. They won't announce that till the fall.

Steve: Oh, okay.

Leo: Pumpkin Pie, I don't know.

Steve: Or pudding or...

Leo: Pudding. I think they should change it before then because, frankly, it's a little unseemly to keep talking about their P.

Steve: Okay, yes. We also have a crazy service for GDPR compliance. This is the EU's privacy and security compliance, which we've touched on before. It comes into effect on May 25th, which is not far from now. And what catches everybody's attention is the apparent penalties for anyone whose services are global, meaning that an EU citizen might be your customer, are breathtaking, in millions of dollars. Anyway, we'll talk about that and this crazy service that has an answer for it.

Firefox has a plan for sponsored content. We've got another million routers being attacked through a new means. Another instance of JavaScript being deliberately subverted, that is, a JavaScript library. And I want to talk a little bit about what this really means in a bigger context because it's very troublesome. Believe it or not, a new Microsoft Meltdown mistake. And the good news of a comprehensive 924, I think it is, page Windows Command Reference PDF that Microsoft has finally published. And a little indication that Twitter may be experimenting with encrypted DMs, so like secure messaging coming to Twitter.

Leo: Oh, wow, interesting.

Steve: Yeah. And based on the screenshot I saw, it looks like they did it right. It looks like elliptic curve keys are being sent back and forth. So lots of news to talk about.

So our Picture of the Week I got a kick out of. This is from the coverage of a disturbing new cryptomalware. Unfortunately, even though cryptocurrency mining is the big thing, there's still people encrypting all the files on individuals' drives, trying to get ransom from them. And I got a huge kick out of the note, which I'll share when we get down to the story. But this is the Windows 10 login screen, which an unwitting user is greeted by after their system has fallen prey to this particular new SynAck version of cryptomalware, where it just sort of says - and in fact the reason I got a kick out of what was written was it sort of sounds like, oh, we didn't do this to you, but we'd like to help you get out of this pickle. And it's like, uh-huh.

Leo: Uh-huh.

Steve: Anyway, so here on the login screen your first experience of this is "Hello. Your files are encrypted. To restore files, please contact us by email." And then it's synack@scryptmail.com or synack@countermail.com. So it's like, oh, something encrypted your files. Oh, goodness.

Leo: Hello. Hello. Hello, file-encrypting person.

Steve: So anyway, we will discuss this a little bit later. It uses a technique which we've already talked about. But essentially these things that are effective never go away. And unfortunately, this is making some money for some people, and so it hasn't gone away.

Speaking of what's not going away, we're going to doubtless be covering, both in the U.S. here and everywhere else, this tension which is not going to go away between a state's, you know, a government entity's feeling of their right to and need to monitor communications within their boundaries, and the fundamental problem that cryptography robustly prevents that. So we've been - and you didn't miss any of this, Leo, because it's been quiet. Well, maybe - no, I don't think you did. Were you here when we talked about how the Russian government blocked about 20 million IPs?

Leo: Yeah, yeah, yeah.

Steve: Right, okay. So that was the last news we had. So this is Russia vs. Telegram. And first Russia...

Leo: Well, this was part of it, wasn't it, that they had to block all those IP addresses - was it for Telegram? Yeah.

Steve: Right. Yeah, yeah, exactly. So they first said, please give us the keys to all of the communications. And Telegram predictably said no. Then they blocked Telegram's IPs. So Telegram moved over to Amazon and Google Cloud services specifically because that would conflict with all these other services using the same things. And the Russian intelligence service said, we don't care. And so they just used like four massive /16s and /15s, so like huge blocks of cloud space got blacklisted. And of course it did, it killed thousands of other - of access to legitimate services. So that only lasted about a week and a half because there was so much back pressure, as you can imagine, I mean, this was a horrible thing to just black out, I think it was like 19 million IP addresses.

So then what happened is these IP restrictions got lifted, but Amazon and Google both started preventing what is technically an abuse of their service, which it's not clear whether they'd been allowing this deliberately or not. But they're not happy about it. And it's something that human rights organizations and people who wanted to prevent censorship have been using.

So here's the idea. It's known as "domain fronting." And it's been in the news a lot because Amazon did tell Moxie at Signal that they were going to have to stop using one of Amazon's domain names to sort of sneak their Signal communications through the Amazon Cloud. The idea is that communications happens when you establish a connection to a remote site on several levels. There's the transport level, where you use DNS to look up the domain name's IP. Then you connect to that IP. Then you do the TLS handshake where, among other things, you use the SNI, the Server Name Indication, to indicate which domain within a shared hosting environment you want to connect to. That allows the endpoint, the other endpoint that you're connecting to, the server, to select among all the certificates it might have as part of the TLS handshake to provide the proper one that matches the domain. Now you've established an encrypted connection. Then, when you make a query, you also in the query headers have the host header, which specifies which domain you want to connect to.

Well, get this. The way both Amazon and Google and Microsoft, the way all these big cloud providers have things set up is it doesn't really matter for the first couple steps who you say you're connecting to. You can say you're connecting to any service that that particular provider offers at the DNS lookup, and even at the TLS handshake level. And if you're really clever, once you've established that encrypted connection, which nobody observing, nobody doing deep packet inspection can see into, then you can use, or you have been able to - this is what just changed in the last week or two - you had been able to specify a different host name, an actual different, like, server you want to connect to in the tunnel, in the TLS-connected tunnel to the cloud, to get to anyone else.

And so the point is this was being - it turns out lots of people were doing this who were using cloud hosting in order to sneak their traffic through passive packet and even deep packet inspection. So somebody monitoring packets would see an individual connecting to a non-censored, no reason to worry domain. But once that connection was established to an Amazon property or a Google property or a Microsoft Azure Cloud property, then the actual query in the encrypted tunnel, which could no longer be observed, would be to a different domain.

And so the idea is that these cloud providers were terminating the encryption at one layer, but then servicing the actual requests at a different layer. And they weren't connected, these layers. So they didn't know or care that the original handshake was to a domain different from the following queries being made in that tunnel. Now, this is technically an abuse of the service. I mean, this is not the way it's supposed to work. And cybercriminals have been abusing this, too. So it hasn't just been people who we might want to be rooting for, wanting them to have freedom within repressive regimes or people trying to avoid censorship. Bad guys were actively doing this, too.

So what has happened is that this is being shut down. Microsoft and Amazon and Google have begun to shut this down. And in fact, in their letter to Moxie Marlinspike, of course, regarding Signal, Amazon wrote to Moxie saying: "Moxie. Yesterday AWS became aware of your Github and Hacker News/Y Combinator posts describing how Signal plans to make its traffic look like traffic from another site, popularly," they write, "known as 'domain fronting,' by using a domain owned by Amazon," and that's Souq.com. Amazon writes: "You do not have permission from Amazon to use" - I don't know how you pronounce that. S-O-U-Q dotcom.

Leo: Souq. It's, I mean, I don't know, but it is an Arab bazaar. They call them the "souqs."

Steve: So they say: "You do not have permission to use Souq.com for any purpose. Any use of Souq.com or any other domain to masquerade as another identity without express permission of the domain owner is in clear violation of the AWS Service Terms." Then they reference Amazon CloudFront, Section 2.1: "You must own or have all necessary rights to use any domain name or SSL certificate that you use in conjunction with Amazon CloudFront." It is a violation of our Acceptable Use Policy by falsifying the origin of traffic and the unauthorized use of a domain.

"We are happy for you to use" - meaning Signal, of course, for encrypted communications - "to use AWS Services, but you must comply with our Service Terms. We will immediately suspend your use of CloudFront if you use third-party domains without their permission to masquerade as that third party." So this has sort of been going on. It hasn't come up in our coverage because it's interesting, but it wasn't very controversial. Now basically this hack - which is what it is. It's a hack to establish a connection under one domain and then issue queries to another domain. It's only a consequence of the architectural split in the way the cloud providers have been operating until now that has allowed this to be possible. So it's going away.

And unfortunately my feeling, I think there's a - my feeling is this is all abuse of technology. I mean, while I'm sympathetic to what people are doing, it's bound to fail. Russia chasing after Telegram demonstrates that Telegram is going to lose this. And there's another one, it's not Ving, it's - I can't remember the name of it. There's another very popular encrypted application which the Russian intelligence organization has now notified they're next. So it's been nice while it's lasted, but it's not going to last. If a country like Russia decides that they absolutely refuse to allow encrypted communications that they cannot see into within their borders, they're going to win that battle.

So anyway, what Russia has now done, unfortunately, in pursuing telegram, is to block 50 VPN and proxy services completely because they were being used by users to avoid the Russian filters. And so once again it's like, okay, if this is what's going to happen. So VPN services, even when they were being used for benign reasons, they're all now suffering because some users are using them to still have access to encrypted communications with Telegram.

So again, it's not something which is going to survive in the long term. I don't know, I'm sorry for the people within these repressive regimes. But the technology which sort of permits it to happen can't withstand someone who refuses to allow it to happen. The nation that you're in is going to win this. And of course we're interested to see how this all comes down in the U.S., as well. But Telegram is already out of China. They're going to end up being out of Russia. And it will have just been a transient capability which ends up getting lost. There's just no way to avoid that kind of filtering.

We've also been following for the last, what, I guess about a month and a half, this drama with Drupal, the so-called Drupalgeddon2, where a very critical vulnerability was discovered. All Drupal sites were given one week's notice of the availability of an update. And we could paint the future in this case, and it's unfolded exactly as expected. Leo, in the show notes here I have a link to the Google Docs spreadsheet which has a list of more than 350 known Drupal sites that have been compromised and are currently serving a cryptocurrency miner into the browsers of everyone who visits them. Most notable among them is Lenovo. Every visitor to Lenovo is running, unless their own browser blocks it, a malicious cryptocurrency miner in their browser.

Leo: The main Lenovo site?

Steve: Yes.

Leo: That's crazy.

Steve: Yes. I noticed, I don't know why, a lot of religious sites were there. Boy, there was the National Labor Relations Board in the U.S. was breached and had this. It's "jquery.once.js?v=1.2" is the cryptocurrency miner which has been edited into those Drupal sites, which is then downloaded by visitors' browsers and run.

Leo: So on this spreadsheet some stuff is pink or red, and some stuff is yellow. Is the yellow less [crosstalk]?

Steve: I don't know why. I saw that, too, and I didn't see any key or indication of what that meant. But it's quite sobering to see the actual list of sites there. And cybersecurity firm Imperva also discovered another campaign, also against Drupal sites, which they named Kitty because the JavaScript which is being maliciously installed is M-E-0-W. So me0w.js. Thus it's the Kitty malware campaign. That one is running a Monero miner from Webminerpool.com. And in addition, they're installing a persistent PHP backdoor into any site that they compromise, giving them long-term access after the site's owner updates.

Of course, that follows onto the longstanding wisdom which is, after your site has been compromised, you can't trust anything about it. I mean, you really have to wipe and reinstall from scratch because, if bad guys get in, they can install all kinds of persistent mischief that you have no way of detecting in some cases. And they're also running a server-side miner. So this goes from, in our coverage, from here comes a problem. Everybody patch as soon as you can. Inevitably that will not be heard because Drupal is the second most popular content management system on the 'Net, WordPress being first.

Many sites will not patch. They will immediately, within a day - the patch was on a Wednesday. The next day, on the following Thursday, there was already sniffing traffic being picked up of sort of a precursor. That matured into attacks. Then we covered, yes, attacks are happening. And now we have lists of compromised sites. So unfortunately, no surprises here. This is going exactly the way we now know these things go, unfortunately.

So hopefully people will see the list of sites that are compromised, know somebody, you know, individuals, know somebody there and say, "Hey, do you realize your site's been compromised? You've got bad guys running around in your system." Because this is a remote execution attack, so a very, very serious and critical attack.

Okay. It's being called GLitch, G-L-I-T-C-H, and GL is in caps in the name of this, as in WebGL. A group of researchers at the Vrije University in Amsterdam have developed a new, startlingly effective remote attack against Android smartphones using the user's web browser and the device's built-in GPU via JavaScript and the WebGL API. So what this means is they have demonstrated that a user of Android visits a particular website. JavaScript is delivered and runs either under Chrome or Firefox. They tested it under both. And get this. In two minutes, the attack is so potent that - and I should mention this is Rowhammer. Within two minutes their phone is compromised.

So it's the first Rowhammer attack to leverage GPUs to leak the computer's memory contents and allow bad guys remotely to compromise a phone. We've been talking about Rowhammer for quite a while. Remember that it leverages essentially noise in RAM access to cause a bit to flip in the actual memory grid of dynamic RAM as a consequence of it sort of being a malicious access pattern. And it turns out that GPUs are far more effective than CPUs. The reason we haven't seen this on desktops before is that GPUs have their own memory. On lower end devices like smartphones and tablets, the GPU shares main memory with the CPU. So by leveraging WebGL, which gives the JavaScript extremely accurate timing information, they're able to perform a cache attack with GPU because the GPU has a much shallower cache than the CPU does in order to compromise the device within about two minutes.

So what's not clear is whether there has been some fuzzing in the WebGL support on those browsers. There had been some fuzzing in timing in response to Meltdown and Spectre earlier this year. Under the Cert.org page, I went looking to see specifically whether Mozilla and Google with Chrome had responded, and nothing was there, although in digging around I saw some reference to perhaps this already having been taken advantage of, but nothing yet posted publicly. So it's not clear. Certainly both companies know, and I'm sure they will be arranging to limit the timing accuracy of WebGL. And essentially that's the crucial thing, as we have been seeing a lot lately, to determine whether - essentially for Rowhammer to be used to penetrate caching is you have to be able to sense whether a particular access was cached or not. And for that you need to know with very good accuracy what the timing was.

So anyway, we do have, again, another instance of our famously often-quoted, and I think it was Bruce Schneier who said, "Attacks never get worse; they only get better." And so here's Rowhammer, which continues to cause us trouble. And hopefully downstream we will see some improvements at the hardware level.

What we know is that it's possible to more intelligently refresh DRAM. Either increase the refresh rate, which lowers the bandwidth that we can use the RAM, so it essentially slows the DRAM down, but hardens it against attack, or probably in the future we're going to see Rowhammer-resistant hardware design, which notices that there is an access pattern which may not be malicious because there's nothing inherently malicious about accessing DRAM. It just turns out you can leverage it.

So next-generation DRAM hardware will probably preferentially refresh rows adjacent to those being accessed heavily in order to prevent these bits from flipping and thus essentially over time adapt the hardware to the attacks. And of course downstream that's where we have to end up being with this continual flow of Spectre problems with the hardware that we'll be talking about a little bit more in a second.

I wanted to soften the blow to our listeners who are using Firefox as their preferred browser, as I am, with the news first of all that Firefox 60 will be returning to a sponsored content model for newly opened tabs. They purchased Pocket last year, and Pocket will be providing some sponsored content. So for people who want to support Mozilla, I think this is a good thing. First of all, it is important to know that Mozilla is absolutely protecting the privacy of their users who view this sponsored content. There's no age, gender, location, user data being sent back to the advertisers who are on these promoted stories, which will be appearing on Firefox's newly opened tabs by default. And if you really object to this, you can turn them off.

So first is maybe you want to support Firefox, as I certainly do. So know that Firefox is keeping the interaction privacy enforced. But if you do object to it, rather than abandoning Firefox, you can just, under the little gear icon on a newly opened tab, you can simply turn off - it's just a checkbox, "Show Sponsored Stories," and that could be turned off. So I hope that Firefox and Mozilla stay with us because it's good to have a solid second to Chrome, if only that, and I still prefer it. Until Chrome gives us much more manageable memory footprint and tabs on the side, I mean, integrated side tabs, which there's been some talk of, but we haven't seen it yet, Firefox is the only browser I can use because it just makes it possible to have lots of tabs open.

And so the news, Bleeping Computer had some good coverage about somebody trying to hide a backdoor in a popular JavaScript npm package. And the specific story itself is interesting. But the background and the trend we're seeing is I think deeply worrying, which is that there are an increasing number of instances of JavaScript packages being deliberately compromised. Not mistakes being made that are found, but backdoors being installed in packages, being snuck into packages on an ongoing basis. In Bleeping Computer's coverage they talked about, extensively, previous instances where this had been done. And if anyone wants the full story, I have the Bleeping Computer link in the show notes.

But my bigger concern is that the entire - and I have to put this in air quotes - "security model," because, I mean, it's an insecurity model, which has just sort of quietly and innocently evolved, is deeply flawed. As we know, the way it is today, websites you visit have links in their page headers which cause our browsers to go out and indiscriminately download JavaScript packages which their site developers, or sometimes their sites' packages developers, I mean, it's like multiple levels of, like, "it's not my problem" indirection caused these things, this executable code to be downloaded and run in our browsers.

In the case that Bleeping Computer talked about, it was a cookies package that was, like, three levels removed where a deliberate exploit had been inserted into it, which just happened to be found by people who noticed that you could use a certain format of query in a header to essentially cause the browser to execute a command that it wasn't authorized to by embedding this in a header. So, I mean, somebody deliberately embedded a backdoor into this JavaScript that was in this JavaScript package being downloaded.

So the problem is, I mean, I have no solution for this problem. I am profoundly unhappy with the way the world has evolved in this regard, this lack of responsibility and this tendency to just keep adding JavaScript packages from third-party sites for functions that the website uses in order to - and, I mean, we've looked at this. Sometimes it's like 40 different sites are having JavaScript pulled from into the user's browser. And of course we know that web security requires every single one of them to behave themselves and not to misbehave. Any one of them that does compromises our security.

So the good news is the browser vendors understand this, and they're doing everything they can to create a sandboxed environment. I don't blame anybody for running their own sandbox or running a browser in a VM, like adding belts and suspenders to the existing security model of the browser. But it's bad enough, I guess is my point, where mistakes happen like as mistakes. It really becomes worrisome when there's a strong incentive for packages to be deliberately backdoored and then leveraged against people who unwittingly download them. So anyway, in looking at this coverage in Bleeping Computer, I just thought, wow, you know, I guarantee you we will be talking in the future about this kind of problem. This is different than what got Drupal, but very worrisome.

So, okay. This is just - it's not really comic relief because it's...

Leo: I'm laughing.

Steve: Because if you're affected by this...

Leo: It's not funny, yeah.

Steve: ...it's just terminal. But the guys that did this crypto malware, which encrypts all of a victim's system, what you get is a "read this please" and then a unique string of eight characters dot txt file. And I'm going to read this exactly as-is because it's clear that English is not their first language. But it says, as if it's like a benefit of some sort...

Leo: It's not our fault. We're so sorry, yeah.

Steve: They have a nice acronym, too: FES. So it's SynAck FES, centered, Files Encryption Software. "Dear Client." Right? Client. "We apologize for inconvenience with your files." Oh, yes, uh-huh. It's a little inconvenient that they're all encrypted, and we can't get them. "So we make a business offer to order file recovery service from us. We do not extort money. Files restore is an optional service."

Leo: Absolutely.

Steve: Otherwise you could just give up right now and go home.

Leo: It's optional.

Steve: Well, yeah, exactly. Eat or die. "Also we will do auditing of your network FOR FREE [all caps]" - they've already been in, right, and rummaging around destroying everything. Oh, look, we found a bug - "for free if you order file recovery service. So some details about SynAck FES. This software uses" - and then it's absolutely true - "an ecies-secp192r1 algorithm."

Leo: Oh, I thought that was made up. That's real?

Steve: It's actually real, "...to create unique pair of private and public keys for the session," which basically means you're screwed. The encryption is good and you can't get it back. Each file is encrypted with random key using aes-ecb-256 algorithm. And, you know, if I were nasty, that's what I would do.

"We strongly recommend you not to use third-party decryptors because they can damage your files." Right, you know, we encrypted them, but don't try to decrypt them. "But if you want to try to restore your files by yourself, make sure you have made backup copies of encrypted files." So they're being really helpful here, Leo, because you might try to fix it yourself, but just make backup copies so that we can help you after you give up.

"And please do not remove files with text notes because they contain important information required for file restoring. If you want to order file recovery service" - which of course is not extortion, it's entirely optional, up to you if you ever want your files back - "please contact them" - I'm sorry, back to them - "please contact our support using one of the following email addresses," and then those are the two addresses I gave at the top of the show, synack@scryptmail.com and synack@countermail.com. "If you have not get a response in 24 hours, please do not panic." Because they're very busy decrypting everyone's files and offering this great service. "Please do not panic and write on Bitmessage [and then] using site https://bitmsg.me." And then they give you their crypto key handle. "Keep in mind that there are fake services offering decryption." Leo, how could anyone set up a fake service to offer decryption? "Do not believe them or you will lose your money."

Leo: [Laughing] We wouldn't want you to lose your money.

Steve: No, no.

Leo: We want you to send it to us.

Steve: Exactly, so we can properly decrypt the files that we have encrypted.

Leo: That's right. It's not a loss. No, no.

Steve: That's right. It's a recovery. Anyway, "There is one method you can use for proof. Ask to decrypt some files for free."

Leo: Oh, nice of them. Free trial.

Steve: "No one except us will be able to do that." And then we have, "Please include the following text in your message." And then they've got a six-line gibberish key thing. "Best regards, SynAck Team."

Leo: I didn't know about this Bitmessage thing. That's kind of cool.

Steve: Yeah, yeah.

Leo: So bitmsg.me, and then you of course have to have a key.

Steve: Yup.

Leo: I presume it's like a cryptocurrency key. Is it actually using...

Steve: Not cryptocurrency, but it is a...

Leo: It's like a miniLock or something.

Steve: Yeah, it's a cryptographically strong key, which you are then able to use to send them a message, and they're able to reply.

Leo: Good. Very, very thoughtful.

Steve: So for what it's worth, cryptomalware is still with us.

Leo: Oh, I think it's worse than ever, Steve. I think the numbers are actually up; aren't they?

Steve: Yeah, yeah, it is, unfortunately. And people click links, and that's what happens when you click the link.

Leo: They do, they do.

Steve: It's like, yeah, I'm an African prince, and I have all this money for you. I just need to talk to you a little bit. Click this link.

Leo: You got that email, too? Oh.

Steve: Well, you and I are both friends of his, Leo.

Leo: Yeah, yeah, yeah.

Steve: So, okay. Twitter discovers a mistake. 330 million people are asked to change their passwords. I got email.

Leo: That's everyone, by the way. That's everyone.

Steve: That's everyone, yes, everyone. The email from them said: "Hi, @SGgrc. When you set a password for your Twitter account, we use technology that masks it so no one at the company can see it," meaning no one at Twitter, and not to mention anybody else who obtains their database. "We recently identified a bug that stored passwords unmasked in an internal log. We have fixed the bug, and our investigation shows no indication of breach or misuse by anyone. Out of an abundance of caution, we ask that you consider changing your password on all services where you've used this password," meaning Twitter and everywhere else where you also used it. "You can change your Twitter password anytime by going to the password settings page." And then they just explain about the nature of this.

So I guess my feeling is everyone should decide. Hopefully, you don't use that password anywhere else. So this is the common advice now of not using the same password anywhere else, annoying as that is. If you're worried about somebody actually having obtained it and tweeting, basically impersonating you on Twitter, being able to log in as you, which could cause you to lose access to your Twitter account, then you may feel the need to change it. It's not hard to do. And then update your password manager in order to know what your new Twitter password is. I think they've done the right thing.

GitHub also just like a week ago had the same problem. They're completely unrelated. Essentially part of their developer pipeline was they were making some changes or doing some maintenance. They turned on a log at a point where incoming passwords were logged in the clear before being robustly encrypted with bcrypt, which is what both services, Twitter and GitHub are using, which is a good, state-of-the-art, password-based key derivation system. And so I think you should consider changing your Twitter password. I imagine everybody got that email. So you probably are aware of that. It doesn't mean that it was breached. They believe it wasn't. But again, as they said, out of an abundance of caution we recommend that people change their password.

Leo: Yeah. I mean, if you have two-factor you're probably not at huge risk, although the real problem is that a lot of people use the same password on multiple sites. So that's the one to watch out for.

Steve: Yes, yes, yes.

Leo: If you've reused that password, you have other places to change it.

Steve: Yup. Okay. So we don't know yet what the "P" of Android P stands for. "O" was Oreo, right, as we march through the Alphabet. So whatever it is, we know a few things about it. They've talked about some of the security and privacy features which are coming. A nice one is that there was some worrisome cross-app leakage of network use. Any app which declared that it needed network access, well, and really what app these days doesn't, would, just by getting that blanket permission, have access to the /proc/net process. This has long been known, and I don't quite get why it took so long for Google to restrict this, and they are now, but it wasn't a data leakage. There's no way that would have been allowed. But any app could see what any other app was doing from a standpoint of what things they were accessing.

So there was a clear privacy concern that all apps that had any access to networking features essentially were able to watch the global Android OS /proc/net process and see what other apps were doing. That's being restricted only to VPNs or apps which Google closely looks at to understand why they have a need to access that. So it is being removed from the way it has always been in P. Also the good news is Android P will block cleartext, that is, HTTP traffic from all apps. HTTPS will now be the only means of accessing the 'Net.

So, yay, we're finally at that point where that kind of enforcement is practical, which it hasn't been until now. There have been some instances where you had to have non-encrypted traffic. No more. So anybody who has an endpoint with an app in Android needs to be able to encrypt it so that their app will be able to get to their endpoint.

Also P will use a uniform UI when requesting fingerprint authentication across apps and devices. So there's been a concern that your fingerprint authentication could be spoofed. And so now they're going to lock that down, and there'll just be one UI that all apps share in order to get this. Also we were talking recently about apps accessing the phone's camera and microphone without any notification. That'll be blocked from all background apps by default. Also, P will encrypt backups on the device with a local secret key before sending it back up for storage on Google's Services. So yay to that.

And two last things. Android P will begin experimenting with support for MAC address randomization. And this is an important feature. We've talked about this before with regard to iOS. All WiFi is Ethernet, and Ethernet uses a 48-bit MAC, the Media Access Control ID. By design, that's a unique number so that you never have two devices that might be on the same Ethernet with the same MAC, or they will collide because the traffic is routed to the devices by their MAC address.

So if you were, in the case, for example, of iOS, which has resolved this, if you were just wandering around with your WiFi radio on, the MAC address of your device would, if steps were not taken, be available without associating to an access point, just because you're out looking, enumerating the WiFi in your neighborhood. As we've discussed before, Apple has long ago solved this by using a randomized MAC address which is not actually you until you associate with the access point. So once you're actually associated with the access point, then your real MAC address is used. But otherwise it's just random.

So that's a really nice tradeoff for staying compatible with the standards, yet not actually being a unique MAC that can wander around. So it's good news that Android P is going to begin rolling this feature. I don't know what it actually means. All I could find was, quote, "Begin experimental support of." So maybe they're just being cautious about this. We'll have to see.

And lastly, support for DNS over TLS, which is another great thing. Remember that DNS by default is over UDP, which is unencrypted. DNS over TLS means probably that Android will be using Google's DNS and set up an encrypted tunnel over which all DNS queries will flow, to hugely boost privacy. So that's all for the best.

And I talked at the top of the show about this upcoming GDPR compliance issue. I will be updating GRC and explicitly stating, in our privacy policy, explicitly stating why and how we are GDPR compliant. It's easy for us. And as a small organization, and you are, too, Leo, you're less than 250 people, so the requirements are substantially reduced. But both of us, GRC and TWiT, have services available to citizens in the EU. So the point is, any site that is offering services globally, and where global includes the EU as it does, needs to be GDPR compliant. And enforcement begins on the 25th of this month, 25th of May.

Anyway, I got the biggest kick out of this. This is truly a service: GDPR-Shield.io. GDPR-Shield.io. And the banner comes up with the good news: "You can save thousands on GDPR compliance." And then the little subheading here: "Simply paste our JavaScript snippet..."

Leo: Oh, boy.

Steve: "...into your..." Huh?

Leo: Okay. Sure. No problem.

Steve: "Simply paste our JavaScript snippet into your website's code. We'll check every visitor of your site and will block access to users located within the EU." Yay. Problem solved.

Leo: That is one way to do it. That's what Unroll.Me does. They don't let anybody from Europe - I think, based on what you were talking about earlier, probably wouldn't be a good idea if people just paste Java snippets into their websites.

Steve: Exactly. That's probably - exactly.

Leo: Who knows what it really does? And by the way, I think it's down because I just went to GDPR-Shield.io, and it's not responding.

Steve: Oh, no kidding. Interesting.

Leo: What a surprise.

Steve: Okay, well, who knows what's going on there. I do have a picture of the site from Bleeping Computer's coverage, where they grabbed a screenshot of the home page, so showing it. And, okay. So the ultimate goal of GDPR, and we've not covered it extensively here, we just mentioned it in passing that it was coming, but it was a long way off, and now it's here, is to give individuals more rights over their data and to restrict how companies process private information.

What really, as I mentioned before, caught everyone's attention is the "impossible to sanely ignore" fines which the legislation imposes upon anyone whose site or services are accessible by citizens in the EU. Under GDPR regulation, firms can be fined - I hope everyone is sitting down for this - up to 20 million euros, about 28 million U.S. dollars, or 4% of the company's worldwide turnover, and I guess that's profits, whichever is - or revenue, rather - whichever is greater, not the lesser of, the greater of $28 million U.S., or 20 million euros, or 4% of your revenue, whichever is greater. You have to have, for example, a Data Protection Officer, a DPO, which having one is mandatory.

But then it turns out, whoops, slow down, wait a minute. If you have fewer than 250 employees, things are a little better for us - me, Leo, by a lot, and you also fortunately. We do not need to - small firms do not need to have explicit documentation explaining why personal data is being collected and processed, which information is being stored or for how long, although I fully intend to explain, essentially the only stuff we have is what we collect for the purpose of selling SpinRite.

Leo: Well, wait a minute, though. You have server logs.

Steve: No, I don't.

Leo: You don't have any server logs?

Steve: None.

Leo: So when somebody visits your site, your servers do not log their IP addresses?

Steve: Nope, no record of it.

Leo: That's general practice for most websites.

Steve: Not for me.

Leo: You're smart. Now you don't have to worry about it. Do you actively delete them, or is there a setting you can turn it off?

Steve: I don't log. No, I have no need to log. No reason to. But if - and I'll explain. I'm going to update our privacy policy, as I mentioned. If someone said I want you to delete your record of my purchase, fine. But you won't be able to upgrade because we use your purchase record in order to, you know, previous proof of purchase. But fine. And that's essentially all that's necessary for compliance for a small firm is to be conscious of users' rights, to tell users what information you are storing about them clearly, and to give them the right to be forgotten, to give them the right to request deletion. So we have that capability, and I'm happy to make that available.

So anyway, so for what it's worth, I wanted to sort of plant the bug in our listeners' ears. I know we have a lot of listeners in enterprises of all sizes. And so just be aware that this May 25th deadline is approaching. Oh, and for what it's worth, it's not as if a single infraction immediately causes the levying of a 20 million euro fine. That's a ceiling. And historically these privacy fines have been, I mean, it's like only if after warnings, and you've been told, and blah blah blah, you don't fix things. So it's there, but it doesn't mean suddenly you're going to get a bill from the European Union for 20 million euros. Doesn't work that way.

But I'm glad that the EU is putting this stuff in place because this is what all sites should do. All websites and services should say, you know, "Here's the data we're collecting on you. And if you object to this, we'll delete it for you." And I guess, if that deletion capability is absent, that's something that you would need to look at, I mean, immediately because this GDPR regulation requires that people doing business in the EU have the ability to delete EU citizens' data completely from their services and sites and so forth. So if that's not there, you need to have that.

Leo: I don't know how we're going to do that. Geez.

Steve: What?

Leo: Well, I'm sure we have server logs. That I can turn off. But we log every download because that's how you count the downloads. And we log the IP address. Is logging an IP address considered to be a violation of privacy?

Steve: Good question. I don't know.

Leo: I mean, you keep track of an IP address for at least as long as the conversation is going on with your website. You wouldn't be able to have a conversation otherwise.

Steve: True. True.

Leo: But that's a transient log. Hmm. That's going to be interesting.

Steve: Be interesting to see whether IP address [indiscernible]. Now, I mean, certainly it only means something when you deanonymize the IP.

Leo: Which you can easily do. But you usually can't get to an individual. Usually only to their Internet service provider. Huh. I'll have to - now I have to find that out.

Steve: Yeah. It's worth - that's a really good question, whether an IP - I'm sure now that we've discussed this on the podcast, my Twitter feed will be full of information.

Leo: Yeah, and that's all we know because we don't ask for additional information. And we don't even sell anything. There's nothing. But we do, I mean, we use Google Analytics, which logs IP address and probably does geographic locationing. And we certainly do that for downloads, whenever you download a show. That's how we know how many people listen to the show. That's how we charge. So I don't know. Huh. We are well under 250 employees, but I don't think that lets us off the hook entirely.

Steve: No, it does not, no.

Leo: Oh, I hadn't even thought about that. Oy oy oy.

Steve: Well, yeah. It seems to me, I mean, in the case of the IP address, if someone said this is my IP, remove it from your logs, I mean, if you haven't deanonymized them, if you don't have a name associated with the IP, and neither of us do, I don't even have logs because I just don't need them, and I don't want them just for privacy, I mean, I just - I have no need for them. Then the only thing I could imagine is someone says, "This is my IP. Expunge it from your logs," which technically is possible. But I doubt...

Leo: Standard server logs with IP addresses must be disclosed in the privacy policy, but you don't have to get consent because they're being collected as part of a business's critical need to prevent fraud.

Steve: Nice. Good.

Leo: Yeah. So you can collect them, but you do have to disclose that you collect them. I think we do that already in our privacy policy. I don't know how we would delete them, though. You must routinely delete logs in 60 to 90 days, I think. I'll have to look at this.

Steve: Well, and in fact in your case keeping an IP longer than that doesn't serve any purpose because...

Leo: No, no, we don't need them, yeah.

Steve: ...they float around.

Leo: We probably do delete them. You typically delete logs because you don't want to have infinite - our logs are very large.

Steve: They do get big.

Leo: They are very big.

Steve: Yeah. In GRC's privacy policy I explain that we don't log by default. But if we need to track down a problem, we may turn logging on briefly while resolving a problem, and then...

Leo: That's typically why somebody logs; right? So you can...

Steve: Turn it off and shut down, yeah. Okay. So more than one million Dasan GPON routers are under attack. So Dasan is a South Korean manufacturer that produces routers known as GPON. That's Gigabit Passive Optical Network routers. There are big networks in Vietnam, Kazakhstan, and Mexico. And more than one million of these routers have been identified from scans of the 'Net. The 360 Net Lab Researchers have identified a botnet controlled from Vietnam currently scanning and exploiting these devices.

What's worrisome, there are two CVEs that have been created for these, 10561 and 10562. 561 reads: "An issue was discovered on Dasan GPON home routers. It is possible to bypass authentication" - get this, Leo - "by simply appending '?images' to any URL of the device that requires authentication" - like, you know, the login - "as demonstrated by /menu.html?images/." And then they give some more options. Anyway, one can then manage the device. And these devices have remote management exposed to the Internet, of course, by default, password protected, but it doesn't matter because you just put "?images" on, and that bypasses, essentially just skips over the check.

Then, as if that wasn't bad enough, the next sequentially numbered CVE, 10562: "An issue was discovered on Dasan GPON home routers. Command Injection can occur via the dest_host parameter in the diag_action=ping request" - and it goes on. Essentially there is a remote command injection vulnerability that makes it quite simple to execute commands and retrieve their output on the router. So we can bypass authentication and give them commands. As a result, they are being swept up right now in a botnet being controlled from Vietnam, so another instance of home routers with a vulnerability that is discovered, and then immediately exploited by bad guys. And it's good news to the individuals whose routers are exploited that people just want to use their bandwidth, probably mine crypto or use them to spread a worm, who knows. But, yeah, the world we live in today.

Speaking of the world we live in today, Alex Ionescu, he's a security researcher we've referred to often with Crowdstrike, tweeted on May 2nd, so what, last Wednesday, he said: "Welp [W-E-L-P], it turns out the #Meltdown patches for Windows 10 had a fatal flaw. Calling NtCallEnclave returned back to user space with the full kernel page table directory, completely undermining the mitigation." He writes: "This is now patched on RS4" - so that's I'm sure Redstone 4 - "but not earlier builds." And he asks: "No backport?"

Okay, now, what was funny was that, as I'm reading this tweet, I'm thinking, uh, wait a minute, that sounds eerily familiar. Yes. At the end of March, on the 28th of March, we discussed on this podcast the so-called Total Meltdown bug which exposed the system's memory, the virtual memory manager page tables for Windows 7 and Server 2008 R2 systems. And remember we talked about it at the time. Basically it was supposed to have the system privilege bit set, but it returned with the user privilege set, which allowed user access to the entire global memory map and read/write access to all of the system's memory. Whoops. So that was the Total Meltdown that was an emergency patch by Microsoft.

Now what was found was that the Meltdown patches for Windows 10, not including the most recent Redstone 4, which is the just-released-last-week version 1803 version of Windows 10, all previous ones apparently have, as Alex tweets, this very bad mistake made in the Meltdown mitigation. So I also, just as we were going to start recording the podcast, I saw a note from Brian Krebs talking about this is the second Tuesday of May. The podcast last week was May 1st, Mayday, the first Tuesday of the month. So this is Patch Tuesday. But I've been checking for patches all morning and getting nothing.

Brian just tweeted, I just saw him send actually email out to his email list, that there were a bunch of patches, 67 of them I think I saw him say. So I've not had a chance to look at them. Hopefully the fix for this is among them for Windows 10 users who have not moved up to 1803. And as we talked about last week, there have been a lot of problems with this Redstone update. Among them, apparently, the Spectre mitigations are not present, and there's no "go out and do them yourself" patch yet for those, although I have one report of it the Spectre mitigations being built in to 1803, which is what we would expect. So still unknown in detail about that.

Leo, Microsoft has published a 948-page PDF.

Leo: I was just reading that.

Steve: The long-awaited Windows Console Command Reference. Never before in one place...

Leo: It's never been documented, really?

Steve: No.

Leo: There's no help file?

Steve: Well, it's just sort of all scattered around, and you kind of have to know who to ask. Now here it is. I've got the link in the show notes for anyone who's interested. I also tweeted it, so @SGgrc. If you're following me on Twitter, you've got a link. It's the 250 Windows Console Commands on a just shy of 1,000-page PDF. So all kinds of stuff.

Leo: Wow.

Steve: Yeah, really cool.

Leo: You know, they're actually, I think, deprecating it. They want you to use PowerShell, not the Command Console.

Steve: Right, right.

Leo: Holy cow.

Steve: And just finally a note that Twitter has an unlaunched secret encrypted messages feature. A security researcher, rummaging around in the pre-release land of the Twitter Messenger app saw a reference to the ability to provide your keys and exchange keys and apparently do encrypted DMs. Shortly after the Snowden revelations, Edward tweeted to Twitter, to whoever the head honcho is or was at the time, that it would be nice to have encrypted Twitter messages. And the response from Twitter was, "That's interesting. We'll look into it."

And it appears that we may before long be talking about encrypted DMs, where you would arrange somehow to exchange - based on the length of the keys, they have to be elliptic curve keys, so they're manageable length. You would somehow exchange those with someone and then be able to use Twitter to send encrypted DMs to each other, which by the nature of the way good encryption, properly done encryption works, nobody would be able to access your tweets, which would be a nice feature.

Speaking of a nice feature, I got a fun nice piece of email from a Bree Duffy, who's located in Oregon. This was the 26th of April he sent this with the subject, he said: "A security concern." And then he said, comma, "SpinRite saves my sister's hard drive and me a headache." And he said: "When I heard about the virtual card numbers from Capital One" - that Leo, you were just talking about - "on your show I was highly intrigued." And I would argue, as he should be.

He said: "However, I'm also highly distrustful of apps from most companies due to concerns of snooping or downright sloppy programming causing issues. Do you have any thoughts on the safety of Capital One's Eno plugin for desktop, or the phone app?" And of course I have no, just to respond to that, I have no a priori explicit knowledge. But I would argue, I mean, we've talked about per-use credit card numbers, and I think that is a huge benefit. So I'm bullish on the service. And I imagine that...

Leo: I'm sure it's well coded.

Steve: ...this was well coded, yeah. I mean...

Leo: I mean, first of all, we should say they're a sponsor, as you know because you just heard their ad.

Steve: Right.

Leo: But you know how many different programs and services you're running on a typical PC?

Steve: Exactly. Exactly. Exactly.

Leo: You know, I agree, and it is kind of my motto, don't install anything you don't need.

Steve: But if it's useful...

Leo: But if it's useful, install it.

Steve: Yeah.

Leo: A Chrome plugin is probably not...

Steve: That's exactly what I was going to say was that I was talking about how well sandboxed our browsers are now as a consequence of the risk of third-party JavaScript being downloaded. These are one of the good guys, not one of the bad guys. So for what it's worth, the idea of not exposing my credit card is clearly there's a value proposition here that has it making sense.

Leo: Yay, thank you.

Steve: Yeah, I have no problem. And as for SpinRite, he says: "Now to SpinRite. I have the rather unfortunate distinction of being the go-to guy in the family whenever there is computer trouble." Leo, you and I both know about that, although I don't think it's unfortunate. I love helping people. But I think he does, too.

Oh, he says: "I love helping them, but figuring out obscure problems can be quite the headache. My sister brought me her computer which wouldn't boot into Windows. It wouldn't even see the boot record. I had fortuitously," he wrote, "just bought SpinRite" - thank you very much, Bree - "and decided to give it a go. As with many such stories, while no errors were reported, all of a sudden the computer booted properly, saving me the headache of either formatting or buying a new hard drive, and reinstalling Windows and all of my sister's applications, et cetera. Thank you very much for making this very useful tool. I look forward to the new versions when you get to it, and SQRL when it becomes available."

And on that note I should mention two things. I meant to note that, as for Twitter and their problem with storing their users' passwords in the clear or having logged them by mistake, that's something that SQRL completely eliminates. As I have mentioned before, the SQRL system gives websites no secrets to keep. That is, what websites store is the user's public key per site. So their per-site public key. It's first of all not valuable to anyone else or useful to any other site. And having their public key is not a secret. I mean, like their public key is not a secret. It's used to verify a challenge that they're able to sign a random challenge, and it's used as their identity, which is kind of cool because it's double function. But there's nothing to lose. The second thing I wanted to mention is it's done.

Leo: Woohoo!

Steve: Friday, Saturday, and Sunday...

Leo: Words I never thought I'd hear.

Steve: I know. I have begun building up the forums, the SQRL forum server. It's up and running. I've got PHP and MySQL and Zen 4.0 as the forum's system all up and running. I'm in the process of configuring it and customizing it and getting it set up. That's the last thing I needed to do because I need somewhere a public web place to host all of the various aspects of SQRL for people to ask questions and get answers. It's not something that I can ask either me or Sue or Greg to do. Everybody wants me back on SpinRite, which is what I intend to do as soon as I get this thing up and running. But it is ready. It is finished.

Leo: Nice.

Steve: And working on getting it public. Then I'm going to come up and spend some time with you and Father Robert, Leo, to do a TWiT Special and tell the world about it.

Leo: Fantastic.

Steve: So it'll be really fun, yes. And lastly, Spectre NextGen. I guess it shouldn't be a surprise. And it's really kind of not because, as I said, I was surprised that Intel was even able to fix this kind of problem in the microcode. And as we know, they were not able to fix it on all of their architectures. Their earlier architectures didn't have the flexibility necessary in microcode to essentially add unplanned-for instructions to give operating systems the control needed over branch prediction in order to mitigate the vulnerability.

But the greater problem is that forever, ever since we began trying to squeeze more processing power out of our chips, we've been willingly ignoring the fact that to do that, processors have been storing the history of code execution and sort of doing a mini version of on-the-fly compilation. That is, if you're watching branches that are taken and not, and remembering whether they're taken or not, and then using them to speculatively execute the branch you expect to be taken next time, then you're storing state based on what the code has done. And if you then switch the context back to a different process, it can cleverly figure out how the processor's behavior has been modified by the code that was previously running. That represents a cross-process boundary information leakage. And that's Spectre.

So what has come to light, the site Heise.de, H-E-I-S-E dot D-E, had the first coverage of this. There are eight new CVE numbers that have been allocated, bug identifiers, but their contents are currently being kept secret. Google's Project Zero has discovered one of eight new Spectre flaws. They're being called Spectre-NG for NextGen. Intel has internally classified, and this is all still - there's a complete cone of silence lid on this, nothing being said yet publicly. Heise got on the inside and had some dialogues. And under agreement of complete secrecy, there's no specifics yet. But they have verified with second and third parties all of their coverage of this.

Intel has internally classified four of the eight new Spectre-NG vulnerabilities as high risk. The remaining four are rated as medium. According to Heise's research, the risks and attack scenarios for the Spectre-NG flaws are similar to those for Spectre, with one important exception. One of the Spectre-NG flaws simplifies attacks across system boundaries to such an extent - and I can't wait to learn what this is because now I'm really curious. I mean, as we know, there was some early Meltdown proof of concept because it was easy to do, but it was also easy to mitigate. That was the caching attack, an attack against caching that was easier to fix. There were never really any Spectre proof of concepts. There was just the spectre of Spectre.

This looks like something easy. They said it simplifies attacks across system boundaries to such an extent that the threat estimate is significantly higher than it ever was with Spectre. Specifically, an attacker could launch exploit code in a virtual machine and attack the host system from there, the server of a cloud host, for example. Alternatively, it could attack the VMs for other customers running on the same server. And Intel's Software Guard Extensions, which are designed to protect sensitive data on cloud servers, are not Spectre-safe.

Heise believes that Intel is planning two rounds of patches. The first is scheduled to start in May. Apparently this Project Zero timed out yesterday. So maybe Google is going to punt on their rigid 90-day deadline just because of how bad this is. I don't know that the one they discovered is this one of the eight that is really the most worrisome. Nothing happened yesterday, so maybe Intel has said we need a little more time on this. We just don't know. And of course we also know that these microcode patches can be problematical themselves. So I'm sure we'll be talking about this in the weeks to come.

Anyway, the first round is scheduled in May, and the second currently planned for August, which, wait, June, July, August, another 90 days downstream. And remember also that Spectre was 60 days embargoed, not the normal 90. I mean, these are really, as we know, this was the big news at the beginning of this year, and now it's back for more. ARM systems are also believed to be affected by at least some of these eight new vulnerabilities. And we don't know about AMD. So stay tuned.

But we do now have another worry. Now, again, understand that this is cross-process and cross-VM information leakage. So it's a concern for hosting providers, much less concern for individuals. You would have to have malware in your machine that was then stealing your credentials or stealing passwords and things. I mean, that's not good. But it's anomalous for you to have malware in your machine. It's not anomalous in a hosting provider because a hosting provider is saying, here's a virtual system. Run anything you want. Well, bad guys could be running something, trying to break out of the VM to get to the host or to get to another VM which is legitimately running, sharing that server's hardware.

So end users, not such a problem. Never has been. Probably still won't be. You know, you don't want malware in your machine at all. So this does break process boundary. And so in time it's worth fixing it. But I would argue, given the problems we've seen with rushed out microcode patches, that it may be worth sitting back a little bit and not getting all hot and bothered over this until there's some reason to worry. So I'm sure we will have more information next week and in weeks to come on essentially a serious collapse in the security, the fundamental security of the architecture of our processors as a consequence of how much was done to make them go fast.

Leo: Wow.

Steve: Yeah.

Leo: And what's your sense? I mean, the next-generation Intel processor's probably not going to fix this. It's going to be a year?

Steve: Yes. The problem is that the processor development pipeline is years deep. I mean, the stuff they're working on now that they will definitely resolve this for won't come out the other end until 2020 at the earliest. So maybe they can fudge some improvements into what they've already got in the pipeline. But they've got an install base they need to worry about. They've got their already in the pipeline yet not released, I mean, it's a little difficult to buy a new Intel chip that has known bugs in it.

Leo: Yeah.

Steve: So, yeah, ouch. It's a mess.

Leo: But it's not like you could go use somebody else's processor. We kind of have to live with it.

Steve: We don't know about AMD. It'll be interesting to see. That could hurt Intel if their processors are vulnerable.

Leo: AMD doesn't have this problem, yeah.

Steve: Yeah, exactly.

Leo: But I wonder how likely that is, given that they have the other Spectre and Meltdown flaws.

Steve: Yes. And the fact that ARM systems are also vulnerable. So this sounds like another new type of bad problem. I can't wait to find out what this is that's like more exploitable.

Leo: Have we seen any exploits in the wild?

Steve: No, no.

Leo: One thing I worry a little bit about is that we're starting, I mean, we talk about all exploits as if they're the same, but really they're not.

Steve: Right, right.

Leo: And I think that exploit hunters are incentivized for a variety of reasons to publish these. But they may be very, very hard to implement.

Steve: Yes. And in fact it's easy to make the point that, when people updated their systems with that first round of Intel patches, and they started crashing and bluescreening and having all kinds of problems, the fix was much more of an actual problem...

Leo: Much worse, yeah.

Steve: ...than what it was trying to cure.

Leo: Yeah. We need to start, I mean, they've always had severity ratings. But everybody says everything's severe at this point. So exploitability ratings, or real threat ratings. Because "severity" really doesn't reflect the real threat. It reflects merely how far in somebody who uses the exploit could get, how bad it would be if somebody used the exploit. What it doesn't address is how...

Steve: We need the GAS rating, Leo.

Leo: What's that?

Steve: G-A-S, give a s**t.

Leo: That's exactly what we need. That really is. And I don't think we have good information about whether this is one to freak out about, or that's one to freak out about. I mean, we know ransomware. We know Petya was something to freak out about and still is, legitimately so.

Steve: Yup, yup.

Leo: But we've not seen any of these exploited in the wild, despite the fact that most computers are still not patched and may never be patched.

Steve: So it's "Should I GAS or not?"

Leo: Should I GAS?

Steve: Yes.

Leo: Yeah. Well, the best way to keep track of all this, of course, is keep listening to Security Now!. We do it every - normally Wednesday. We're on a - no, no, we're on our regular time. What am I saying? Everything else got moved. You were the floating island. Everything else moved around.

Steve: You're a little jetlagged, my friend, but other than that we're right on time.

Leo: Well, no, you know, I've thought you were on Wednesday ever since you were on Wednesday.

Steve: That's right.

Leo: But it is Tuesday, and has been for years, 1:30 Pacific, 4:30 Eastern, 20:30 UTC. If you want to watch live, you can go to TWiT.tv/live. We are a little later than usual, that's for sure. You can also get copies of the show from Steve's site, GRC.com. He has the 64Kb audio and transcriptions. And I know that's hugely valuable to a lot of you, not the least because you can search for information from the shows because the transcripts are searchable. Isn't that nice?

We have audio and video at our website, TWiT.tv/sn, and you can always subscribe, whatever podcast app you use. I promise you we don't collect information about you. We do collect your IP address. Somebody does. Maybe we don't do that. But I think we do. Cachefly does, the company that does the downloads. And I just don't know what to do about that.

Steve: Well, and Leo, let's remember, anything your browser touches is an IP address.

Leo: It's saying hello, yeah.

Steve: Your IP is just being sprayed all over the globe.

Leo: Well, and I looked, and there is some debate about whether the EU will consider IP addresses personal information. And if you were to err on the side of caution, you would say that it is. In which case I don't know what we're going to do. We may have to block this show to all people who live in the EU. I think I found a JavaScript snippet I can use. No, actually that site's down. I think that maybe they got forced off the 'Net, yeah.

Steve: Wow, interesting.

Leo: Steve Gibson, have a great week, and we will see you next time.

Steve: Okay, my friend. Right-o. Bye.