Meltdown & Spectre Emerge

Description: This week we observe that the Net Neutrality battle is actually FAR from lost. Computerworld's Woody Leonard enumerates a crazy January of updates. EternalBlue is turning out to be far more "eternal" than we'd wish. Will Flash EVER die? There's a new zero-day Flash exploit in the wild. What happens when you combine Shodan with Metasploit? Firefox 59 takes another privacy-enhancing step forward. We've got a questionable means of sneaking data between systems; another fun SpinRite report from the field; some closing-the-loop feedback from our listeners; and, finally, a look at the early emergence of Meltdown and Spectre exploits appearing in the wild.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-649.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-649-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We're starting to see the first exploits of Spectre and Meltdown. Fortunately, they're not potent yet. But Steve will give us the lowdown on that. New Firefox capabilities and a whole lot more, it's all coming up, as always, next with our Securer in Chief, Steve Gibson. Stay tuned.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 649, recorded Tuesday, February 6th, 2018: Meltdown and Spectre Emerge.

It's time for Security Now!, the show where we cover your security online, your privacy. We cover all the big exploits, all the little bugs, and we do it all with style, grace, and panache thanks to Captain Steven "Tiberius" Gibson of the good ship What the Hell's Going on Today. Hello, Steve.

Steve Gibson: Leo, great to be with you. Boy, next week is going to be Episode 650.

Leo: Yikes.

Steve: Which of course makes this one 649 since we're incrementing by one. And once again we've got those words in the title. This one is "Meltdown and Spectre Emerge."

Leo: This is like the eighth week in a row or something.

Steve: Well, yeah, that's - what would it be? Well, the fifth week because this is how the year began was with this news. And now AV is picking up exploits. So this takes us from, oh, it's only theoretical, nobody needs to really worry about this, blah blah blah, to uh-oh, yeah. Not as bad as it's going to be. That's why it's just emerging. But we'll be talking about that this week.

We're also going to talk about the fact that it's very clear that the Net Neutrality battle is actually far from lost, that is, what Ajit Pai has done probably doesn't matter that much. It would have been nice to leave it the way it was because everything was tied together and unified. But we do have a system of federation. And, oh, it's going to come back to bite the FCC. Also, Computerworld's Woody Leonard enumerated the crazy January we have just suffered through of updates. And I just got a big kick out of the way he put it, so we'll touch on that briefly.

EternalBlue is turning out to be far more eternal than we would wish. We pose the question, will Flash ever die, speaking of Eternal Blue that hasn't, will Flash ever die. There's a new zero-day Flash exploit in the wild. And actually a couple interesting things that our listeners can do. In the newer versions of Office there are some ways to mitigate Flash also that I want to touch on. We'll also pose the question, what happens when you combine Shodan with Metasploit? It's not good. Firefox 59 is taking another privacy-enhancing step forward.

There's also a questionable means of sneaking data between systems. Some security outfit said, oh, we came up with a new way of getting data between two systems. But it's hard to imagine something kind of less practical in actuality. So we want to take a look at that. I have another fun SpinRite report from the field that I read to Lorrie a couple days ago, and she got a kick out of it. I thought, well, okay, I've got to share that with our listeners.

We've got a little bit of closing-the-loop feedback. And then we're going to take a look at what has been happening thus far in January as we come into February with actual evidence of Meltdown - we really do need to combine these somehow.

Leo: Smeltdown.

Steve: Smeltdown or Specdown or Meltre or Meltre or I don't know. Really, it's like annoying. Anyway, they are in the wild. And after our first break we'll take a look at this week's picture, which I've had on the back burner. I thought we were going to, like, not need it because we were going to have closed this chapter. But no. Chapter's still open.

Leo: I got a - I should have tweeted you. I had noticed a Picture of the Week. It's a new - I have to find it. There's a new dating club of people who share the same password. And I thought you would - match with the person who shares your same password. I thought you might enjoy that. I'll see if I can find it.

Steve: Imagine what else you may have in common.

Leo: Just imagine.

Steve: Well, unfortunately, Intel doesn't.

Leo: Uh-oh.

Steve: And, well, I mean...

Leo: We know, doing it right.

Steve: Yes, they're scrambling to deal with, as we know, a set of very, very clever exploits of some fundamental architecture of all modern processors. Last week's Picture of the Week was people have been having fun with the Intel logos. And I saw this one where someone said, yeah, this sort of reminds me of the original Intel logo that they left behind. And anyway, it's sort of fun.

Leo: Oh, this is funny. I get it now. I had to look at it for a bit.

Steve: Yup. It shows the kernel memory leak coming from inside - it's that circle where the ends don't meet, originally, deliberately, for the Intel Inside logo. But now this one has added a little red arrow.

Leo: They have to close that ring, I think.

Steve: That's right. The ring has not been closed, and it's like, whoops, we have little kernel memory leak.

Leo: That's hysterical. I love that.

Steve: And maybe not so little, too. So, yes, we're working through the plethora of imagery that was generated as a consequence of this, which we'll be talking about at the end of the podcast when we talk about what's been going on. I did want to note some follow-up and a little more detail about what I mentioned last week because Wyoming and New York's governors had both signed some executive orders. And just as we were starting the podcast, news came that the California Senate had passed, I think it was 21 to 12, some similar legislation, the idea being that states have a lot of power over what happens within their borders.

And while even though the major ISPs were lobbying hard to prevent states from going their own way relative to Net Neutrality, it turns out there are strict limits over what the federal government can do because, for example, states have control over their purse, and they can decide how they want to spend their money. So what happened was that we had a bill that passed through California's Senate, that is now moving on to the

Assembly. And I just checked my notes, it did pass 21 to 12. However, this looks like it's just the beginning. There are also a group of 22 state attorneys general that have gathered together and filed a suit against the FCC.

It turns out it's not possible, much as the FCC would like to simply change their own rules, there are laws that govern what the FCC is able to do. And it's necessary for the FCC to demonstrate that conditions have changed sufficiently since an original law was put in place for them to use that to justify subsequent change. And that hasn't happened since the Net Neutrality law was brought up. It was called the Open Internet Order in 2015. So it turns out that 22 states have joined together in an action against the FCC to challenge Ajit Pai's much noticed decision just to toss that 2015 decision out.

But more importantly, the EFF, of course, our friends at the Electronic Frontier Foundation, are very much involved in this. They posted sort of a response to California's action, sounding a little annoyed. Apparently they were involved in the lawmaking. And for whatever reason, I don't know anything about what's going on behind the scenes politically, but the legislation which California has passed so far they worry would not stand legal challenge. Ernesto Falcon, writing for the EFF, said that there were much stronger things that California lawmakers could have done, and he enumerated three which are really interesting, so I wanted to share them with our listeners because this gives us a sense for just how much clout states actually do have over what goes on inside their borders.

First, Ernesto writes: "California spends hundreds of millions of dollars on ISPs, including AT&T, as part of its California broadband subsidy program." He says: "The state could require that recipients of that funding provide a free and open Internet, to ensure that taxpayer funds are used to benefit California residents rather than subsidizing a discriminatory network." He writes: "This is one of the strongest means the state has to promote network neutrality," and it's missing from this legislation which just passed which was SB 460.

Secondly, he writes: "California also has oversight and power over more than four million utility poles that ISPs benefit from accessing to deploy their networks." He says: "In fact, California is expressly empowered by federal law to regulate access to the poles, and the state legislature can establish Network Neutrality conditions in exchange for access to the poles." He says: "Again, that is not in the current bill passed by the Senate."

And then, third: "Each city negotiates a franchise with the local cable company, and often the company agrees to a set of conditions in exchange for access to valuable, taxpayer-funded rights of way. California's legislature," he says, "can directly empower local communities to negotiate with ISPs to require Network Neutrality in exchange for the benefit of accessing taxpayer-funded infrastructure." He says: "This is also not included in the California bill."

So I wanted to do some more digging into this because of course this has been a big topic of ours and of huge interest. And the voting public has demonstrated overwhelming support for the concept of Net Neutrality, which for reasons of the way Washington works with power brokers and lobbying and money, isn't what our government has been doing. But it looks like, at the state level, there is very, very strong capability of enforcing this locally. So as I noted at the top, I think we're far from losing this.

Leo: Well, I'm glad you're bullish.

Steve: What do you think?

Leo: We had Denise Howell on The New Screen Savers. And of course the issue always comes down to interstate commerce. States are forbidden from interfering with interstate commerce, and we imagine the Supreme Court, in the long run, will have the deciding vote on all of this. And meanwhile, while this is going to be battled, as it will be for weeks and months and maybe even years, we won't have Net Neutrality because that's the current situation. So it'll be interesting to see what happens. There's all these lawsuits. There's 21 lawsuits from 21 states. And we'll see.

Hey, I just wanted to show you something. There's a live camera, I don't know if you - did you watch the launch, the Tesla, I'm sorry, the SpaceX launch?

Steve: Thankfully, I had your stream on, so I was watching TWiT at that time.

Leo: Huge success.

Steve: Oh, my goodness.

Leo: The Falcon Heavy, largest rocket ever made, launched successfully today. Yeah, we just kind of broke into MacBreak Weekly to show it. And I think we all knew that Elon Musk, who has created SpaceX, had put his original Tesla Roadster, cherry red Tesla Roadster...

Steve: Okay, now, will you explain that to me? Because I haven't been following along.

Leo: Well, it's a test, right, it's a test launch. So they need weight. So they just, instead of putting concrete bricks in there, he said, well, I'm just going to put the car in there. And he put a - he didn't need it any more, I guess. He can get another one, I'm sure. And he put a spaceman in it, in the driver's seat. And it's playing a loop of David Bowie's "Space Oddity"...

STAFF: In a vacuum.

Leo: Over and over again. Of course you can't hear it because it's in a vacuum; right? But, hey, it's not the point. "I'm a star man waiting in the sky," you know. No, it's "Ground control to Major Tom," isn't it. Yeah, yeah, it's "Ground control to...." Anyway, there's also - I didn't realize this, and John just told me. The capsule with the car had fairings when it was launching, of course. Those fall away. And the car is now out in the open as it rotates the Earth. And of course Elon being Elon has put a camera on it. And this is live footage, watch this, from the roadster floating in Earth orbit right now in space.

Steve: Oh, you're kidding me.

Leo: I'm not making this up, but you'd think I was. This is the planet. It's coming in over the driver's right-hand shoulder. It is a stunning view. So I just wanted to point that out. The Starman feed is live on YouTube, and actually I've rewound a little bit to show this shot. But it's YouTube.com, I think /starman. Anyway, search for "starman live" on YouTube, and you can - is that not a shot? That's a live shot from the capsule.

Steve: This is unbelievable.

Leo: Because Elon is a madman in a very good way.

Steve: Oh, lord.

Leo: Isn't that great? Forget sci-fi. Forget "Altered Carbon." This is it. This is real. Okay. Enough said. On with the show. I just had to - I found that...

Steve: We do live in fun times, Leo.

Leo: Oh, it's exciting. It's really exciting, yeah.

Steve: So Woody on Windows, that's the name of the column that Woody Leonard has been writing for quite a while in Computerworld. Woody of course, you and I both know him, he's been around forever.

Leo: Woody Leonard, yeah, great guy.

Steve: He's won a whole bunch of awards over the years.

Leo: Great guy.

Steve: And he's pretty much disgusted with January.

Leo: Good, because it's February now, so that's okay.

Steve: Yes, yes. He called it "The perfect end to a perfect month." And this, of course, this was on Wednesday the 31st. He says: "Yet another Win10 1709 Cumulative Update." This one was KB...

Leo: This is live.

Steve: Why don't we just stop the podcast, Leo.

Leo: Let's just watch this. Watch this.

Steve: Sit here and, oh. So was that the moon that we saw leaving the first time, and now we're...

Leo: No, I think it was the Earth. They're in Earth orbit right now. But he has multiple camera angles, obviously. And so this is another angle of the planet.

Steve: This is ridiculous. I mean, it's ridiculously perfect.

Leo: Gives new meaning to that little blue marble, doesn't it. Wow.

Steve: Wow.

Leo: Sorry, I didn't mean to interrupt you. Woody, what's Woody saying?

Steve: Who? Who? Woody who? Oh, look at that. That's just ridiculous.

Leo: That's a live shot.

Steve: Okay. We're teasing our listeners.

Leo: Well, yeah, you'll be able to go back and see it, if you have the video. Or even if you don't, you can go to YouTube. Wow. Wow, wow, wow.

Steve: Yeah.

Leo: Yeah.

Steve: So in his column - oh, and by the way, I did fire up my Win10 machine that I'm talking to you over right now. And sure enough, there was another Windows 10 update. The good news is I've learned the lesson. I fire it up hours before the podcast so that it has a chance to revamp itself. And sure enough, KB4058258 was waiting for me.

Leo: Now I've got to try, mm-hmm.

Steve: Uh-huh. Another cumulative update. So he enumerates. I won't go through it in

painful detail. Woody does in his column. But in short, Woody notes that we received patches last month on January 3, 4, 8, 9, 11, 12, 17, 19, 22, 23, 24, 26, 29, 30, and 31 - 15 out of the 31 days. And you might say, okay, January 1st doesn't count because that's New Year's. But still, half of the days of the month we received updates and patches.

And where did I have it? Oh, yeah. He says: "Today is the 15th day this month" - he's writing on the 31st. "Today is the 15th day this month that we've seen Windows patches, yanked patches, patches of patches, and re-re-re-patches." Then he says: "Welcome to the third cumulative update for Win10 Fall Creators Update this month." So anyway, I just, yes, we all know this has been going on. We've been covering them. As we've been discussing, in some cases it's up to users to go get them because Microsoft, I don't know, just got tired of...

Leo: And it fails sometimes, right, without warning. Yeah.

Steve: Yes, yes. And some of them killed AMD. Then they had the "Fix the AMDs that we killed before." And then we had the "Oops, we're sorry, the firmware" - okay.

Leo: There's a message from space that says "Don't Panic." Did you see that on the screen?

Steve: And now we have a camera looking over the right shoulder of Starman.

Leo: Yeah. Starman's well covered. There's the - if you just go back, you can see where it is and so forth. But, yeah, the screen on the roadster says "Don't Panic," a tribute, of course, to Douglas Adams and "Hitchhiker's Guide to the Galaxy." I love that.

Steve: Yup, the cover of the book. And for our listeners who didn't see the return to Earth of the two boosters, oh, my god.

Leo: That was like ballet. That was synchronized swimming. I mean, unbelievable.

Steve: Again, just go find - you have to go find the return.

Leo: Well, you won't have to go. It'll be everywhere.

Steve: It's true. It'll be in your face. Yeah, that's true. You're right, it'll be on mainstream news, I guess, this evening.

Leo: Oh, absolutely, yeah. You couldn't do a more photogenic, public-pleasing PR.

Steve: It's ridiculous. It's just...

Leo: Yeah, yeah.

Steve: Unfortunately, speaking of ridiculous, cryptocurrency mining malware has now infected over half a million PCs using...

Leo: This is becoming a plague.

Steve: Yes, it really is. So to remind our listeners, EternalBlue was the name given to the Windows SMBv1 exploit, which was believed to have been developed by the NSA and was leaked by the Shadow Brokers hacking group. And that was earlier last year. It was famously leveraged by the widespread WannaCry ransomware, which we covered at the time, devastating the U.K.'s NHS, Health Network System. I mean, they had to basically take their health network offline because of the ransomware that attacked it. And of course it was also used in that Petya/NotPetya worm.

Microsoft issued a security update to patch the flaw being leveraged by the EternalBlue exploit, which used the first version of server message blocks, that is, SMB, the Microsoft Windows file and printer sharing, basically, and a lot of other stuff got loaded onto that over time. They issued a patch last March 14th, okay, so not quite a year ago, but coming up on a year ago. Middle of next month a year ago. March 14th, 2017. And surprised us by even patching the long unsupported Windows XP for the same flaw, essentially at the time acknowledging the severity of this problem. So at its peak, tens of thousands of machines were infected.

Okay, now, fast-forward to today, as I said, nearly a year later. EternalBlue exploitation, despite having been patched nearly a year ago, remains alive and well, with several cybersecurity firms reporting the discovery of a new cryptocurrency mining virus that's being spread - so I guess that maybe makes it a worm, although it was described as a virus, and I didn't dig in - being spread using EternalBlue.

Researchers from Proofpoint discovered a massive global botnet which they dubbed Smominru, S-M-O-M-I-N-R-U. I guess that's Smominru, that's using EternalBlue to spread and infect Windows computers to mine Monero cryptocurrency. Now, what's interesting is that, since this virus or worm is mining Monero, the money needs to go to the holder of the Monero payment address, which allows its success to be audited.

Okay. So it's been active at least since May of 2017. So two months after Microsoft issued the patch, this thing started going. It's infected over, what, maybe now that makes it nine months, more than 526,000 Windows machines, most of which are believed to be servers still running unpatched versions of Windows, according to these researchers from Proofpoint. Based on the hash power associated with the Monero payment address for this operation, which is auditable, the mining network is - they've verified it's a massive botnet which has mined approximately 8,900 Monero valued at approximately $3.6 million USD.

Leo: Whoa. So this is worth it.

Steve: Yes. Thus the problem we have here. And the mining botnet is currently running at the rate of roughly 24 Monero per day, generating $8,500. So it is producing...

Leo: That's a lot of incentive.

Steve: Exactly. And as we know, security is about incentive. I mean, everything we learn is that security is to varying degrees porous. And if nobody really wants that much to get in, then some adequate security keeps them out. But if somebody is absolutely determined to make something happen, unfortunately, there are always ways to get in. And so here they've made $3.6 million USD at the current valuation of a Monero. And they are earning, if you call it that, $8,500 per day by stealing processing power from the machines that have been infected by the exploit which we believe was discovered and in the toolkit of our U.S. National Security Agency for some time.

So you know, Leo, we've been doing the podcast now, we're in year 12. I wouldn't have believed this story. I mean, it sounds like science fiction. It sounds like, what? And here we are. Yeah, wow.

Leo: Thanks, NSA. Thanks, Microsoft.

Steve: Uh-huh.

Leo: Thanks, Intel. Thanks to all of you.

Steve: Well, and this also, I mean, further demonstration, we're about to talk about this Adobe Flash zero-day. But this shows that, while patching is necessary and important, obviously, there is just this bell curve, you know, there's this huge distribution of systems. Our listeners are following along, trying to get their firmware updated - well, or maybe not now, now waiting until Intel fixes the Spectre firmware. Then, I mean, people in our orbit here are on the leading edge of being secure. They're taking action. They're segregating their networks. They're using firewalls. They're using secure browsers. They're running uBlock Origin and understanding about advertising ware and so forth.

So we're at one end. On the other end are servers that still have Code Red and Nimda worms scanning the Internet, looking for exploits. So, and then we have everything in between. But here's something that's been patched for which there are servers publicly exposed on the Internet that, against all reason, have file and printer sharing ports exposed that allowed them to get taken over by the use of the Eternal Blue exploit and run this Monero miner on their server hardware. And so it's like, well, it's a crazy world, but it's the one we're in.

Leo: That explains something because I had a woman call the radio show on Sunday, maybe you heard it. She had ESET Security Suite. And she said, "Somebody keeps trying to get into my printer."

Steve: Oh.

Leo: And I said, well, that's Internet Background Radiation, a term you coined, which is there's always script kiddies and other scanning IP addresses looking for vulnerabilities. I said, merely telling you somebody's trying to get in, that's fine. That's normal. If they get in, then you'd want to know about it. What I didn't realize is they're not trying to get in to use that old exploit, to just print something and go ha-ha. They're trying to run a bitcoin or a Monero miner on your printer.

Steve: Very likely. I mean, yes.

Leo: Wow.

Steve: Commandeering processing power. And notice that one of the mitigating effects of cryptocurrency mining now is, as we know, the amount of power consumed. At least in the case of the bitcoin blockchain, it takes more electricity, especially now that the bitcoin value has dropped so much from its brief high a couple weeks ago, it no longer is profitable in many states, like in California, because our electricity costs more than it does to generate a coin. There are states where it still makes sense. But the key here, and the reason essentially this $3.6 million had zero cost, and the 30 whatever it was I said, 3,600, or no, $8,500 per day these guys are making, multiply that by 30 for a month. I mean, they're still making serious coin. But it's zero cost because other people are paying for the power. Wow.

Leo: Wow.

Steve: But so I guess the point I want to make is that, while patching after the fact is important, it is also very clear that proactively pushing updates, I mean, which despite the controversy of Windows 10 and how heavy-handed Microsoft has been, and how upset people have been to have their machine reboot in the middle of the day or when they're trying to use it and so forth - and to give them credit, Microsoft has backed off and now is much more negotiable along those lines.

But the point is, you're going to get patched. Sooner or later, if your machine is on the network, it's going to find out there's some updates, it's going to get them, and it's going to, sooner or later, it's going to patch itself. And step by step we've been marching sort of backwards against this. I remember when updates sort of became - our machines were going to go get them. A lot of the old timers among us said, what? I don't want my machine getting updates. I want to go get them myself. And so you remember those days, Leo.

Leo: Oh, yeah.

Steve: And so we've sort of been successively moving away, and it's hard to argue when we see all of these servers that are sitting there exposed, it's arguably being abused. And, I mean, the good news is from their standpoint all that's being done is that their processors are running much hotter that they otherwise would and consuming more power than they otherwise would because people are using them to mine cryptocurrency rather than crawling inside and seeing what's going on in their network and abusing them

more deeply. No, it's just about money. Which in their case is probably a good thing. But it's hard then to argue against any company being as deliberate as Microsoft has been about our ecosystem is going to be patched moving forward, period. It is going to happen.

Leo: Well, you can see why. You can see why. LawnDog wonders in the chatroom if there's - you could monitor for outbound traffic. Would there be a signature that would let you know that you've got a Monero miner on one of your systems?

Steve: I would say that the easiest indication is look at the processor. It's going to be pinned. It's going to be at 100% or 95%. And hopefully that's not normal. You know, GRC's runs around half a percent, if that. I mean, it just sort of ticks along doing nothing most of the time. But the real key is that these miners are going to be saturating the processor. So that's the thing to look for. And certainly there is some communication back to the command-and-control server in order to, you know, that this miner is using. But anyone running a server should have some sense for how much their processor is in use. And that's, I mean, it's like a bright light in the dark if your processor is pegged at 100%. That's like, okay, wow.

So one last comment, and then we'll take a break. Oldies but goodies. Adobe Flash is another thing that - as I was putting this story together and noting that Adobe has just today pushed out an update for this newly discovered zero-day, and the fact that we now use the Adobe Flash Player. Remember it didn't used to be Adobe's Flash Player. They bought that, right, from Macromedia? And I just wonder if they are happy with the association of their name, Adobe, with Flash Player.

Leo: And whose idea was it to buy this?

Steve: Exactly. Here we are, how many years downstream, and they were probably thinking, oh, yeah, we're going to expand our repertoire, and we're going to buy this Flash Player from Macromedia, and that's going to be a really great thing. And now Adobe Flash has become, I mean, like a horrible thing. Adobe Flash Player, that's not a good thing. And so I just wonder, on balance, if they were to do it again, is this an acquisition that they would think made sense over the long term? Because how did they ever really make any money on it? It's not clear to me.

Leo: That's a good point, too. Well, no, I think they sold tools for people writing Flash plugins and stuff like that. So I think they probably did make money on it.

Steve: And then I think those tools, though, I think that the actual language went mainstream so that there were third parties making compilers and construction kits and so forth.

Leo: That's an interesting question. I'd love to - nobody at Adobe will ever admit. But I'd be very curious to see if Flash, on balance was a good idea.

Steve: Yes, because Adobe Flash Player, it's like, ooh, you know, it's got your name right

off the bat there. So what happened was that South Koreans spotted a new Flash Player zero-day being used by North Korean hackers against them when they were researching North Korea. So far it's been used in targeted attacks. It looked like it was email or spreadsheets or something. Somehow the Flash Player was embedded. And so the idea being I guess probably watering hole attacks. The Flash Players were stuck in things that South Koreans were obtaining from North Korea, and the North Koreans laced them with an exploit that had never previously been spotted in the wild, thus zero-day. Adobe issued - Adobe, the owner of Flash Player for better or for worse...

Leo: Oh, those guys.

Steve: Yeah, issued another critical vulnerability security advisory saying that it exists in Adobe Flash Player 28.0.0.137 and earlier, saying that successful exploitation could potentially allow an attacker to take control of the affected system, saying Adobe is aware of a report, blah blah blah. Anyway, there's a CVE number. We are now at 28.0.0.161. You probably know, if you have Flash Player, if you need it for some reason, if there are things that you use. I do have it in my system, but it is locked down. I have to, like, explicitly permit three different dialogs because Firefox has it set on probation, uBlock Origin says, uh, what? And there's like something else, I have a plugin that's also doing something.

I mean, for me, you have to really, really want to run something in Flash. And I know that because I went to get.adobe.com/flashplayer/about in order to check which version my system had. And sure enough, I had an older one. But in order to do that, that page tries to run the Flash Player in your system, if any, in order to have it report its own version. So I had to, like, successively click on, yes, allow this time. Yes, only now. Yes, I'm sure. Yes, I'm not currently intoxicated.

Leo: You don't want to run Flash. What, are you crazy?

Steve: Exactly. So anyway, at this point, as we know, HTML5 has duplicated the functionality that Flash used to provide. There are still a few crazy websites that are trying to exist, like Flash-based, where the entire website is Flash. And I just say to them, well, good luck.

Leo: Good luck, yeah.

Steve: Good luck. Not much traffic coming their way lately. But the things that, I mean, once upon a time you had to have Flash in order to do some fancy stuff. This is now all moved into mainstream web standards. So it's just - it's over. Nothing but inertia, like the websites that are Flash-based. And there are some games, I guess, that are Flash-hosted games. But, boy, it's just not worth the security risk.

Leo: No, no, no.

Steve: So there is an update available now. Get it if you know you need it. But in any event, you definitely want to be running Flash Blocker stuff on your web browser. IE,

Microsoft is now taking responsibility for keeping its Flash updated, and so it'll keep you current.

I did want to also note one other thing because the way these things were coming in was through Office attachments. In the newer versions of Microsoft Office there's something called Protected View. Under File > Options > Trust Center there's Trust Center Settings. And under that is something known as Protected View, where you've got three checkboxes: Enable Protected View for files originating from the Internet, Enable Protected View for files located in potentially unsafe locations, and Enable Protected View for Outlook attachments. And you want those things on because that's just general, I mean, we've been complaining about Microsoft running scripts in email as long as this podcast has been going. For 12 years it's been why does Outlook run scripts?

Leo: No, please, no.

Steve: When did email ever have a valid use for a script? It just never made any sense. So if you are a Microsoft Office user, File Options > Trust Center > Trust Center Settings, and enable Protected View. You're able to then selectively say, oh, yes, I want to run Flash that I've received from North Korea, please.

Leo: Please let me.

Steve: Please let me. Otherwise, no. And so that's certainly the way you want your default to be.

So speaking of, okay, now, I don't know if this is a smart guy. This is borderline irresponsible. But I guess if you can, then it'll happen. What do you get when you combine the Shodan global search engine...

Leo: Yes?

Steve: ...with the Metasploit exploitation toolkit?

Leo: Oh, no.

Steve: Yes. You get...

Leo: Rosemarie's baby.

Steve: ...Autosploit.

Leo: Autosploit. Oh.

Steve: Autosploit, now available on GitHub. As its pseudonymous author Vector states on GitHub: "As the name might suggest, Autosploit attempts to automate the exploitation of remote hosts."

Leo: Because why work hard if you're going to exploit.

Steve: Because, yeah, because script kiddies need something more to do. "Targets are collected automatically by employing the Shodan.io API. The program allows the user to enter their platform-specific search query, such as Apache, IIS" - webcam, DVR, whatever - "upon which a list of candidates will be retrieved."

Leo: Oh, man. Fully automated.

Steve: What do you feel like attacking today? You want to go after toasters or microwaves?

Leo: Actually, this is - I have grudging respect, I mean, this is kind of brilliant.

Steve: "After this operation has been completed, the exploit component of the program will go about the business of attempting to exploit these collective targets by running a series of Metasploit modules against them. Which Metasploit modules will be employed in this manner is determined by programmatically comparing the name of the module to the search query. However," he says, "I have added functionality to run all available modules" - because why not.

Leo: Why not? Just check them all.

Steve: You just might get lucky. You know, the microwave exploit might work on the toaster - "...against the targets," he says, "in a Hail Mary type of attack, as well. The available Metasploit modules have been selected to facilitate remote code execution" - because why not - "and to attempt to gain reverse TCP shells and/or Meterpreter sessions." We'll discuss that.

Leo: Meterpreter.

Steve: We'll discuss Meterpreter sessions. He says: "Workspace, local host and local port for MSF facilitated back connections are configured through the dialog that comes up before the exploit component is started.

"What is Meterpreter, you ask? Ah. As Offensive Security" - that's the name of their site and company, Offensive Security - "describes it, Meterpreter is an advanced, dynamically extensible payload that uses in-memory DLL injection stagers and is extended over the network at runtime. It communicates over the stager socket and provides a comprehensive client-side Ruby API." Because we could. "It features command history, tab completion, channels, and more." And I got a kick out of this because finally...

Leo: Nice UI job, by the way. Very nice.

Steve: Very, yes. The GitHub presentation notes under the topic of Operational Security Consideration: "Receiving back connections to your local machine might not be the best idea from an OPSEC standpoint." In other words, having the infected targets calling you. "Instead, consider running this tool from a VPS [Virtual Private Server] that has all the required dependencies available."

So anyway, essentially this takes the notion of finding and exploiting publicly exposed machines to a new level. And, predictably, the wider security community is unhappy. A guy named Richard Bejtlich with TaoSecurity replied on Twitter in response to Vector's announcement there: "There is no need to release this. The tie to Shodan puts it over the edge. There is no legitimate reason to put mass exploitation of public systems within the reach of script kiddies. Just because you can do something doesn't make it wise to do so. This will end in tears."

Leo: Oh, what a fuddy-duddy. Come on.

Steve: I don't know. He's a spoilsport, Leo.

Leo: Spoilsport. A metasport.

Steve: A metasploitsport, yes. If you put your toaster on the 'Net, it's because you want somebody else to [crosstalk] bread.

Leo: Yeah, 'sploit me. Come on, 'sploit me.

Steve: That's right. And now it's just a pushbutton away.

Leo: Wow. "It'll end in tears," I love that.

Steve: It'll end in tears. No good can come of this.

Leo: Well, that's obvious.

Steve: Yes, get Autosploit on GitHub and have fun. But again, you do want to be careful because you'll have people knocking at your door. You don't want...

Leo: It's a federal offense. Let's not...

Steve: Yes, there is that, too. Just because you can doesn't mean it's legal. Firefox is

continuing to advance its users' privacy, starting with Firefox 59. Leo, do you remember those quaint old days when we were on Firefox 3?

Leo: Three, four, yeah.

Steve: We had 3, had 3.2, 4.

Leo: Those were the days; huh?

Steve: Fifty-nine and counting. So starting with 59, private browsing mode will strip the path information after the domain from referer headers. So a little bit of review. Remember that the referer header, which is famously misspelled. I just love that.

Leo: Yeah.

Steve: It's funny, too, because I'm sure that some copy editors somewhere are correcting them because in various stories that I saw discussing this, they fixed the spelling incorrectly.

Leo: Yeah, no, it's one "r." It's one "r."

Steve: It's one "r," yeah. It's R-E-F-E-R-E-R is the actual usage in the HTTP specification, which of course is misspelled. It should be R-E-F-E-R-R-E-R. Anyway, this header, when a web page reaches out for any reason, it's going to get other assets like images or, controversially, ads. Or if you click a link to go from that page to another page, part of the request for that asset or the next URL for the next page, it includes the referer, which as its name suggests, provides the URL of the page making the request. So that's very handy for servers to know who's pulling images.

Back once upon a time when bandwidth was expensive, people objected to their images being stolen by other sites. So, for example, people would put up a web page and find some cool image somewhere else. Rather than lifting the image itself and hosting it on their own server, they would just grab the URL so that someone looking at their page would have their browser go get the image from its original source. So on one hand you might say, well, that's nice, they didn't steal the actual image file. On the other hand, they were causing the bandwidth to be stolen, you might argue, because the sourcing server kept having to give the image to every other browser on the planet that asked for it.

So in fact at one point that practice was blocked because servers would start looking to see whether the referer was the site's own page, which was what was intended, or some random page somewhere else. Now, you could also use it to collect some interesting forensics because you could say, oh, look at all the different places that are pulling images from us. You might argue that they were bandwidth pirates, but still.

Then of course the other true benefit was that you were able to see what sites had links to your site, that is, if you click on a link on a page, like to jump from one domain to

another, it was useful to get the referer data. You could see where inbound visitors were coming from. The problem, however, is that you must be careful to use this properly. There have been problems found. For example, since the entire URL is normally in the referer header, if that contains sensitive information, that sensitive information can leak.

In one instance, for example, researchers at the EFF, our Electronic Frontier Foundation, found that the referer headers that were being sent from HealthCare.gov were passing on data about the age and zip code of the visitor to HealthCare.gov, along with whether or not they were a smoker, and their income. So, whoops, a bit of a privacy problem there. You've got to wonder who designed that site, but that's beside the point. The problem is it can be a privacy problem. So what Firefox is doing starting with the next version, is they will be stripping the path information from referer headers when you are in private browsing mode.

So normally the referer contains the entire URL. But you can think of the URL as being broken into two pieces: the domain, you know, www.google.com/, and then all of that other data, the page and also the query information. So what Firefox will be doing is it will be retaining the domain information in case that's useful to know where someone came from, or if advertisers want to use that to glue credit for the ad being pulled by a given site, then that will be possible. I got distracted by the video that you guys are showing. You guys are having fun over there.

Anyway, so to prevent this type of data leakage, that will be the default for private browsing mode in Firefox. However, in the coverage for this, I noted that Firefox does allow users a great deal of flexibility with their referers. So I wanted to aim our listeners at a wiki page on Mozilla.org. It's wiki.mozilla.org/security/referrer, in this case spelled correctly, so R-E-F-E-R-R-E-R. There are some very powerful settings available in about:config. So if you're a Firefox user, put about:config in your URL. And as we know, you get a bazillion, that's the actual count, various entries.

Then you need to search for, spelled incorrectly, R-E-F-E-R-E-R. And that'll bring up a bunch of very interesting settings which have normal defaults, but it's worth considering if you want to change them. For example, there's network.http.referer.trimmingPolicy, where you're able to control how much referer to send, regardless of the origin. That is, whether it's the same origin - or that is, how much referer. The default is send the full URL. You could set it to one to send the URL without its query string, or two to send only the origin, that is, only the www.domain.com whatever it is.

You also have a setting for, similarly prefixed, .XOrigin. It's .XOriginTrimmingPolicy. Where, depending upon whether you're in the same domain or not, the same origin or not, you can have Firefox strip out the query string or send only the origin. And then there's also a XOriginPolicy where, again, you're able to control how much referer to send. So anyway, there's a lot of cool stuff buried in Firefox, if you put about:config and then search for "referer," R-E-F-E-R-E-R. Without doing any of that, the next version of Firefox will start stripping the path and query tail from all referer URLs. So again, continuing to look after our privacy.

And I did say I wanted to talk about something that I think is really sort of specious, but it's just sort of interesting. It got coverage in the press, which sort of surprised me. Researchers at Fidelis Cybersecurity say that they have identified a new technique that attackers can potentially employ for covertly exchanging data using standard security digital certificates. X.509 is the format of these digital certs. And they say that the method builds on previous research involving the abuse of text fields in digital certificates to move data across a network. It takes advantage of the way digital certificates are exchanged during the initial TLS handshake. Or the mutual authentication process that

happens when two systems attempt to establish and resume a secure session with each other.

Jason Reaves is the principal threat research engineer at Fidelis, and he said: "Most other research involving using X.509 certificates for data transfer" - now, remember, certificates are normally used only to authenticate the identity of the endpoints. So the idea of using them for data transfer is, first of all, bizarre. But, he says: "Most other research involving using X.509 certs for data transfer involves the use of text fields in the certificate such as 'Subject' or some of the other common fields such as 'notbefore' and 'notafter'." Okay, right, those are the start and end dates of the certificate.

He says: "Researchers have previously shown how attackers might use these text fields to covertly send and receive data between systems. Our method," he says, "is embedding data inside of a certificate extension. This means you can send data between two systems purely from the TLS negotiation."

So I'm thinking about that, and it's like, wait a minute. All of the certificate body, all of the contents is subject to the signature of the signer. So the only way that you can be offering valid certificates is if all of these wacky data-carrying certs are signed by a third party that the recipient trusts. Now, maybe they're suggesting, and this was not clear, they're suggesting that you just manufacture bogus certificates, which are invalid, that don't carry a valid signature, and thrust them upon the other endpoint. And so I guess if something was watching the other end, it could be extracting this data from invalid certs which are going to be rejected anyway.

I mean, the whole thing just, like, okay. It just was bizarre. I was surprised that it got picked up in the security press as, like, something useful in some fashion because I just can't see. Maybe, as I was like brainstorming, thinking how could you make this work, well, if Let's Encrypt was involved, that is, if you were using their dynamic API in order to make certificates for you on the fly, which can be done, then you could be minting valid certificates and be sticking extension fields into them in order to send data to the other end. But, okay, why not just let the TLS connection come up, and then you've got encryption. And if you want to send data. I don't know. The whole thing seemed kind of wacky. But who knows. Maybe it makes sense.

I did have yet another interesting application of SpinRite, not quite as strange as last week's. But I got email from Matt Bokan with the subject "DJ Steve - SpinRite Gibson saves the concert." And he wrote: "Hi, Steve. I'm not a music producer, but I've got a friend who's making music on a Korg synthesizer which cost many, many kilo dollars. These beasts," he writes, "have had SSDs in them for a long time now." And of course our listeners know where this is headed.

"On the day of the concert there was a sound check, and this synth has all the rhythms, beats, and whatnot of the band stored inside. But on that day it wasn't working properly. Everything was having such a lag, or wouldn't load properly, that the concert was in danger and in the process of being canceled. I had helped this man set up and connect his living room with projector, surround, and PC in my youth. So I got a call to ask if I could somehow transfer the data to another identical synth if they could find one on short notice."

He says: "I didn't even know about Korg synths having SSDs. But when I found out, I ran SpinRite, and half an hour later the synth was singing like a rock star. The concert was not canceled at the last second, and thousands of concertgoers sadly don't even know that Steve Gibson was the true rock star that day." Okay, well...

Leo: Aw. That's sweet.

Steve: Matt, thank you very much for sharing your story. It's certainly appreciated.

Leo: Yeah, it's neat.

Steve: And yet another application of SpinRite. Two interesting bits of closing the loop. Chris Duncan, who tweeted from @cyberdunks, said: "@SGgrc Is it possible for @intel to encrypt and decrypt the on-chip cache on the fly using its own keys as a mitigation for Spectre? That way the chip knows what's in the cache for speculative execution and not the OS." And I thought that was interesting. And it sort of gives me an opportunity to explain a little bit more why that won't work. And that is that it's not the access to the data by the other application in the cache which is the problem. That is, the caching is transparent. And the system handles loading the cache and invalidating the cache and all of that on its own.

The trick is the timing. That's why we keep talking about access and the need for high-resolution time. The idea is that the contents of the cache is inferred by a malicious bit of code doing something and carefully looking at how long it took to do that something, whatever it was. If it took no time at all, then the code knows that the system had what was necessary to do that little bit of work, that fetch or that store or whatever, in the cache already. If it does something, and it takes longer, that is, literally the instruction, that one instruction takes longer to execute, then the malicious program knows, it's able to infer that what it asked the system to do could not be fulfilled from the cache's contents.

And it's because the cache is globally shared among all the processes on the system, because of that global sharing, it's able to infer what other processes had caused to be loaded in the cache from its own timing. And this is why this is not a bug in the Intel architecture. This is just - it's hard to call it a feature. I mean, it is a feature, but it turns out it's dangerous. It's very clever on the part of researchers to have figured out that this cache, the delay of a single instruction being executed could be leveraged into inferring the contents of the system's cache. And it's because all the other processes are sharing that cache, they're leaving evidence of what they've done behind, based on just what's still in the cache or not.

So as you can see, whether the system, no process is like seeing encrypted or decrypted contents of the cache, so encrypting it or decrypting it wouldn't change this. It's whether it's there or not, that go/no go. Was there a delay while the processor went out to fetch what was requested, or was it still available in the cache? So by that very delicate timing difference it turns out malware is able to leverage that to create an information disclosure problem. And that's where we lost January to was all of that. And as we'll be covering in one second here, that's not the end of it.

I did want to mention, this came back onto my radar from someone tweeting. Rob Fairhead tweeted: "Upgraded my network following the excellent Ubiquiti Home Network project at" - and then he gives a GitHub address. It's github.com/mjp66/Ubiquiti. Which he said: "...first seen on @SGgrc #SecurityNow." He says: "Now have segregation for IoT devices and guest network. Longtime itch scratched." And I was reminded of that. So I went to take a look at what Mike Potts had done. He now has, off of that page, a 105-page beautifully detailed, really it's a work of PDF art. And I noted that he had recently

updated this to add support for the Quad 9 DNS provider.

So Mike is clearly, I mean, he is a Security Now! listener. He's a Ubiquiti router user. And he put together this beautiful how-to on using the amazing power in that $49 Ubiquiti EdgeRouter X in order to create a highly segmented network where all of your dangerous toasters and microwaves and IoT devices can be on your network with access to the Internet. But anything nasty that might climb into them cannot get over into your mainstream PCs, where presumably you want to keep anything bad out. Anyway, I just wanted to point our listeners again to Mike Potts' beautiful work on this now 105-page PDF, which now he recently edited to add support for using the Quad 9 DNS. Nice work, Mike.

Okay. What's happening with Meltdown and Spectre? The good news is there is good news.

Leo: That's good news.

Steve: That's good news, yes. Because last week we were saying, oh, don't worry about this. It hasn't appeared in the wild yet. It's all theoretical. Yes, it's a problem. But yes, Intel hasn't yet issued firmware. And I ought to note that apparently they just did. I found out about this last night while I was pulling things together, and I was doing that away from my normal office environment. I have a NUC that Intel - what was called Next Unit of Computing, NUC. It's a kind of a - it's mini form factor PC. They have issued Spectre mitigation firmware for their own family of NUCs. I don't know about anything else yet, but I will certainly keep an eye on that. So it looks like they are beginning to come out, to come back now with firmware which they feel confident for Haswell and later processor architectures will work without the, quote, "higher incidence of rebooting" unquote problem. So that's good.

However, an organization known as AV-TEST, which is an independent organization which evaluates and rates antivirus and security suite software for Microsoft Windows and Android platforms, has been monitoring an increase in hits by AV software on the known patterns used by Meltdown and Spectre. By January 17th of last month, so middle of the month, AV-TEST reported it had seen 77 separate malware samples related to the Meltdown and Spectre CPU vulnerabilities. By the 23rd, that number, 77, had grown to 119. And by last Wednesday, which was January 31st, last day of the month of January, they had collected a total of 139 samples from various sources, researchers, testers, and AV companies. These do not appear to be weaponized, active, like successful data exfiltration, but this is the beginning.

So what we're seeing, what this looks like is this is clearly on the radar of every attacker, every malware author worth their salt. We were just talking at the beginning of the podcast about the fact that EternalBlue, a now long since nine months ago patched Windows vulnerability, is in active use today. It seems obvious that these sorts of vulnerabilities, the Meltdown and Spectre vulnerabilities, which have not yet been weaponized, are going to be. And what's necessary is to somehow arrange to get code to run on a platform. As we've always said, shared hosting is the biggest problem; but there's one place where code can run easily from a remote source, and that's in our browsers. Both JavaScript and WebAssembly are two ways that your computer is running code from someone else.

Now, obviously Google is on top of this. They've got their Retpoline solution. Remember, that's the Return Trampoline mitigation that we talked about, which is a mitigation

against Spectre, which does not require firmware updates. People are going to feel more comfortable with their firmware updated, especially hosted serving environments. That's really - the hosted server environment is the biggest target, where you would be running some code from one customer, and you'd want isolation between it and code running on shared platforms from other customers.

However, end users are - there's some danger. But, boy, if you've got something malicious running in your own local system, our own OSes are so insecure that they don't have to bother with Retpoline or, I mean, they don't have to bother with Meltdown and Spectre in order to get up to mischief in your own system. The place where we as end users need to be on the lookout is our browsers, which are constantly, as we know, now running code from other sources.

The good news is Google is on this. I'm sure that Mozilla is on it, and Microsoft is on it. Hardening our browsers is really what has to happen. And that's a relatively contained code base. So basically it means using Google's very clever and publicly released Retpoline technology on the code of the browsers so that no JavaScript or WebAssembly code obtained from a third-party source can be successful. But the reason this is still of interest to attackers is, as we've said, just as it's been nine months since Microsoft patched for the EternalBlue exploit, and it's now being widely used, there are going to be a huge, huge number of browsers that are not being updated.

So it still makes sense for bad guys to see if there is a way to take these Meltdown and Spectre vulnerabilities and use them to exfiltrate information from users' browsers, like the username and password database that so many browsers now contain in order to fill in forms for users. And in fact proof-of-concept browser exploitation code has been found in the wild for IE, Chrome, and Firefox. It now exists. It has not been proven to be effective, but the bad guys are poking at it and looking at it and developing proof of concepts and trying to use it.

So I would say one thing we want to do, if your firmware is updated, then you have no vulnerabilities. That is, you're running Windows. GRC's InSpectre app shows that you're secure for both exploits, Meltdown and Spectre. You get yes, you're safe, yes, you're safe in both cases. Then you're okay. The firmware, as I mentioned, as we know, got released and then retracted, and Intel is apparently now working on beginning to get it out again. So if you can get your firmware updated, again, you're safe.

If you can't, then as long as your browser is updated for Spectre mitigation - because it's the Spectre, of the two, Windows and presumably any of the other OSes, those immediately mitigated against the Meltdown problem. As long as your browser is Spectre-proof, then you're okay. And I'm sure that here before long we will see some browser actual test site stuff to verify whether browsers are safe or not from Spectre. But so that's where we stand. We are now seeing emerging, as would have been predicted, people beginning to play around with these things.

The good news is there's enough of a window, the industry responded immediately, we got immediate Meltdown protection, and that was the easiest to exploit and the easiest to fix. The browsers can fix Spectre. And I would argue that that's the big source of vulnerability is letting either JavaScript or WebAssembly run in our browser, in a browser that is not - where both we do not have updated firmware and the browser has not been updated to employ its own mitigations. If either of those is true, if the browser is updated or you've got updated firmware, then you're okay. So we will of course be keeping an eye on this moving forward and also keep track of where Intel is with firmware updates because that's really what we want.

Leo: Well, I'm just glad that it's not an effective exploit. But you're right, that's the first step, isn't it. Just get something, yeah.

Steve: Yup.

Leo: Well, thank you for keeping us apprised of the ongoing situation in the Spectre/Meltdown, or Speltdown. Or Mectre. We haven't yet really - Melctre. Steve Gibson does this show every Tuesday at 1:30 Pacific, 4:30 Eastern, 21:30 UTC. If you want to come by and say hi, you can watch at TWiT.tv/live. Please join us in the chatroom. A great bunch of people in there at irc.twit.tv. Kind of a support group for those of us shivering in our boots.

We are now - we have a Flash Briefing, you know, we do of this show. Sometimes we do an on-air Amazon Echo. And it is now available, not only in the U.S., but Canada, Australia, the United Kingdom, and India. So if you live in one of those countries you can get a little bit of TWiT in your Flash Briefings, some technology news from this show and many of our other shows, just by going to your Echo app and searching for TWiT and adding that to the Flash Briefing. And you know, I think it's like 50,000 people do it now. The number's going up fast. So thank you for doing that.

It's also time for our annual survey, another thanks to you for doing that. We don't really want to pry into your personal affairs. We're not trying to get your information. But on the other hand we kind of need to know a little bit about you to help us talk to advertisers, know which advertisers you're interested in and what programming you're interested in.

So once a year, just once a year we do a survey, completely voluntary, you don't have to give us any information, you can check the boxes you care about and not the ones you don't, or all of them, if you want. It should just take a few minutes. It's at TWiT.tv/survey. It is very helpful to us. So if you get a chance, and you're in front of a browser, TWiT.tv/survey. I promise we are not going to use any of this information personally about you. It's all in aggregate, as I explain on the website. You can see my attempt at explaining what's going on at TWiT.tv/survey.

If you like this show, you can get a copy of it from Steve at his site, GRC.com, Gibson Research Corporation. In fact, while you're there get SpinRite, the world's best hard drive and macovery rutility, recovery mutility, you know, the best program for recovering your hard drives. SpinRite is his bread and butter, so you support Steve when you buy a copy of SpinRite. And of course you keep your drives happy. That's GRC.com. While you're there, there's lots of free stuff, too, not only this show but free programs, lots of information. It's fun to browse around. If you do want the program, he has not only MP3 audio of it, but he has the transcripts. Elaine Farris, she's back now?

Steve: Yup, yup. Well, actually this week also we will be a little bit late.

Leo: A little late, okay. But you'll get it there. And what's nice is you can search those, which is a great way to find something in any of the 649 shows. But you can

also - some people like to read. Some people are more visual, they like to read while they listen, whatever. It's there for you. We have audio and video of this show at our website, TWiT.tv/sn. We also put it on a YouTube channel.

And more importantly, go to your podcast app on your phone, on your tablet, on your computer, and subscribe, whether it's iTunes or Pocket Cast or Overcast. And that way you'll get an episode each and every week, the minute it's available. You won't have to wait. GRC.com is Steve's site. You can tweet him at @SGgrc. That's where he gets his questions, his comments, his ideas. And you will be, I presume, back next week?

Steve: Yes, for Episode 650. Woohoo!

Leo: Yay. What's that in hex? No, pretend I didn't ask.

Steve: No.

Leo: Thank you, Steve. We'll see you next time on Security Now!.

Steve: Thanks, buddy.