News & Memory

Description: This week, Leo and I catch up with the past week's news. Did Microsoft lose control of its secure boot Golden Key? We discuss AdBlock, unblock, counter-unblock, and that counter-counter-unblock is well underway. Leo tells a story from the field about Avast A/V. A "security is hard to do" mistake is found in an update to the Internet's TCP protocol. We talk about Microsoft's evolving Windows Update policies, an ber-cool way for developers to decrypt and inspect their Firefox and Chrome local TLS traffic, a nice write-up of our "three dumb routers" solution, trouble with Windows Identity leak mitigation, yet another way of exfiltrating data from an air-gapped PC, and some fun miscellany. We wrap up with a discussion of Intel's forthcoming memory breakthrough.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-573.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-573-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. A big clarification: We've all been reporting the story about Microsoft and the magic golden key being released. It turns out it ain't that way at all. We'll have a clarification from the one guy who understands what's actually going on. He'll also take a look at a very exciting new kind of memory. Intel has just announced it. It's going to change everything. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 573, recorded Tuesday, August 16th, 2016: Memory and micro kernels.

It's time for Security Now!, the show where we cover your security and privacy online with this guy right here. This here is Steve Gibson of the GRC, Gibson Research Corporation, at GRC.com. He joins us every week after extensive research on the scene, and he lets us know what's going on.

Steve Gibson: Frantic, panting, pull everything together research.

Leo: You're not a last-minute guy, though. I don't think so. You're very thorough.

Steve: Well, I start worrying about it, I mean, I'm collecting all week, and then what often happens is I'll make a final posting to the SQRL newsgroup in the, like, middle of

the afternoon on Monday, saying, okay, here's an update to the client. It does, you know, these things are added. Now I've got to switch my attention over to the podcast. So it's a full, focused 24 hours that I put in for the podcast, with then sort of a background collection of things that I see.

Leo: Wow.

Steve: No, well, I mean, it shows, and I'm glad.

Leo: It does show. It does show. We're very grateful for it. You work hard.

Steve: And, boy, this is an example. A really interesting question we answer about whether Microsoft lost control of their so-called "secure boot golden key." We have the Adblock unblock, counter-unblock, and counter-counter-unblock being well underway. I want you to share, Leo, your story from The Tech Guy this weekend.

Leo: Oh, did you hear that?

Steve: About the woman who had the Avast AV problem.

Leo: Yeah.

Steve: We've got really a classic "security is hard to do" mistake in a core update to the Internet's TCP protocol, which ended up inadvertently creating an exploit, which again, it's like, security is hard, as we often say. We've got - I want to cover Microsoft's evolving Windows Update policies, which is all good news for everybody.

I stumbled upon, actually, thanks to a Twitter follower who sent me a link, an ber-cool way for developers to decrypt and inspect their Firefox and Chrome local TLS traffic, which, like, is one of the reasons that HTTPS is sort of a pain for developers is your browser encrypts things. And so you can't see it if you're just doing a packet capture. It turns out there's a way, without needing to set up, like, a Fiddler/man-in-the-middle sort of deal of any kind.

Also I found a really nice write-up of the "three dumb routers" solution that was done. We've got some trouble with, as I mentioned last week, the Windows identity leak mitigation causing all kinds of problems. And so that's not the way to do it. I'll talk about the way to do it. Yet another way in the never-ending stream of ways of exfiltrating data from an air-gapped PC. We've got a bunch of fun miscellany. And then, actually coming from two topics you covered on TWiT, one about micro kernels and the other about Intel's forthcoming memory breakthrough. I was listening to TWiT, and I had some things, sort of a little more deep dive into those two topics that I thought all of our listeners would find interesting.

So a catch-up on the news and some fun miscellany. I figured out, by the way, what was CUJO'ing my system at Level 3, so I'll talk about that and a whole bunch more.

Leo: Wow. Jam-packed. And I can't wait. I'm actually really intrigued with - we had a fascinating TWiT on Sunday. Greg Ferro from the Packet Pushers Network, obviously a geek of high water, brought up some very interesting things. So we'll talk about that in just a second.

Steve: And you had Allyn on the show also.

Leo: Had Allyn Malventano. It was a great show. It was fun. Devindra Hardawar. It was just jam-packed. Alex Wilhelm. I hear a magic sound.

Steve: Well, as I promised, I did not mute all of my devices.

Leo: But that was not a "yabba dabba doo." What was that?

Steve: No, that is exactly 2:00 o'clock. We have an hourly chime.

Leo: Oh, okay.

Steve: You know, there are many things that Mark Thompson and I are aligned on. When we first met each other actually is when you were present at the first Gnomedex.

Leo: I never forget it. So much fun.

Steve: Where I was the keynote speaker for the first Gnomedex. He and I met for the first time. We had both chosen the same speakers for our stereo systems, both the same make and model of projector for our projection TV. And as we went through, we had independently chosen, made all the same decisions. One way that we're very different, though, he's also a big anime person. And just no, thank you. I just never got into anime. He can't stand anything making a noise. And I use sound as another input channel.

And so when Amazon is, you know, I get a notice for delivery, I get some monkeys that are jumping around. I have an hourly chime. Of course, famously, yabba dabba doo when a copy of SpinRite sells. And so I just, you know, I have a lot of sounds. And my PC also, that's the other thing, things like, for example, you can associate a process starting and a process stopping with sounds. And so, like, things are going click click click click and bing bing bing boink. But it's useful for me to know when something is going on because it's like, okay, wait a minute. What is that? I didn't do anything. Whereas Mark has no sound. And actually...

Leo: Yeah, I'm kind of with him. That would drive me crazy.

Steve: ...a tremendous investment in a big PC enclosure that manages to cool some

serious heat-generating machines absolutely silently. So that when he's working, you know, he's surrounded by screens the way I am. But there's just no sound. Whereas it's a little bit of a carnival over here. So, you know, I always turn all of that off so as not to have the podcast disrupted. But many people have said, "I would really just love to hear my own purchase of SpinRite." So as I mentioned last week, okay. We'll see how that goes. So for what it's worth, every hour we'll get a chime, and maybe - I already heard the monkeys before this, so Amazon's done its delivery. So I don't think we'll hear them again.

Okay. So the big news is the massive misreporting across the entire industry of this Microsoft secure boot golden key. That did not happen, not in any way, shape, or form.

Leo: Oh. What?

Steve: It was complete, completely wrongly reported. Threatpost said, "Microsoft mistakenly leaks secure boot key." 9to5Mac says, "Proof Apple was right to fight the FBI." Ars Technica, "Microsoft secure boot firmware snafu leaks golden key." None of that is true. Complete misreporting. In my notes I said...

Leo: What?

Steve: The report on this has been 1,000%, meaning very, very inflammatory and incorrect.

Leo: Okay. So just to - because I've been reporting it that way. We don't have the sophistication to know this is wrong. My understanding was there is secure boot. In most cases, by the way, you could turn it off. Otherwise I wouldn't be able to put Linux on all these machines that I put Linux on. You just disable it in the startup BIOS. But on some Microsoft machines, like the Windows RT tablet, you can't disable it. And apparently there's...

Steve: And phones.

Leo: And phones, that's right. But so as is often the case, like with the CSS, the DVD encryption and the Blu-ray encryption, there's a key, a master key; right? And Microsoft, it's said, please correct us, published this master key inadvertently in a source code posting on GitHub. All of that's not true?

Steve: No.

Leo: Oh, my god.

Steve: So, okay. So the hackers who figured this out are absolutely talented, and they did a terrific job of cleverly uncovering an exploit.

Leo: Not to mention publishing that exploit with fine synthesizer MIDI music and an animated golden key.

Steve: And, now, see, this is an example of what does drive me crazy, is like...

Leo: It's hard to read this. You have to cut and paste it, frankly, is the only way to read it.

Steve: The good news is you can do a Ctrl-A, and it will mark the entire thing, copy, and then drop it in Notepad. Immediately then close that page. Lord knows what it's doing to your processor. And then at length - so those are actually hashes. That's not the key. Those are hashes...

Leo: Ah, okay.

Steve: ...of individual bits of secure boot.

Leo: SHA 256-bit hash.

Steve: So here's what happened. What they did was uncover a mistake. And again, I want to make it clear. These guys are good. They did a great job. Unfortunately, they mixed a little of their own personal agenda in with reality, trying to draw a conclusion that was unwarranted. So, but again, great reverse-engineering and cleverness. Just not what - essentially, they used the term "golden key." And because their write-up was itself very confusing and dense, I had to just, like, go back and forth and make notes and leave a trail of crumbs and, like, okay, wait a minute now, and, like, basically decode what they published because it was really, I mean, these guys are not technical authors. They're hackers.

So what this actually was, was an implementation design error in the handling of boot permission policies which can be used to trick older version of the UEFI secure boot manager using some components of an update. So the so-called Redstone version of Windows 10, which is 1607, version 1607...

Leo: The anniversary update, the current, yeah.

Steve: We know it as the "anniversary update." It added some new technology, the concept of supplemental secure boot policies, which can, for example, be used for test signing development code. And of course that could also be malicious rootkits and so forth. So there is, there's a fundamental problem with secure boot. You just referred to it earlier, and that is, what if I want to run Linux?

And we did a whole podcast on the UEFI secure boot technology and how it starts from a known unmodifiable piece of integrity on the motherboard which cannot be changed, and carefully brings the system up by verifying the signature of everything that it loads

before turning control over to it. And if you do that perfectly, what you end up with is a system in a known state. What you also end up with is a system that you can't change. So there's a tension in this approach with people who, for whatever reason, want to change; and for non-malicious purposes, like developers needing to develop kernel drivers, boot-time drivers, and developers.

So with this anniversary update, Microsoft created an extension to the boot policies known as supplemental policies, which have weakened verification, but which that same new boot manager understands. So there was no problem with that. These guys realized that what that did, though, was create a vulnerability because these new supplemental policies with a lower degree of verification could be used to fool older boot managers which didn't know to specifically check for them. So they looked like regular policies. The updated boot manager is aware of them.

And so this is a little bit sort of like that weird problem we have, like with people doing fresh installs of Windows 7, where Windows Update has changed so that you can't use Windows Update to add the updates to Windows 7 until you manually update Windows Update so that it then knows how to read the data from Microsoft's changed Windows Update servers. Similarly, Microsoft enhanced the flexibility of secure boot in a way that, if you're using sort of synchronized pieces, everything's fine. But they just missed the fact, and these guys caught it, that you could take the new policy parts from the update and use it to fool the pre-update boot manager. That's all that is. There's no key involved.

So what they wrote, you can see where the press got this, they wrote: "You can see the irony. Also the irony in that MS themselves provided us several nice 'golden keys,' as the FBI would say, for us to use for that purpose."

Leo: Ah.

Steve: So they were using the term referring to the FBI's use or request for some way of unlocking the system. But the press picked it up incorrectly.

Leo: Like there was a key, right.

Steve: Thinking that it was actually a key.

Leo: Of course that rotating floating golden key in the demo scene might have had something to do with it.

Steve: Oh, I mean, yes. And so, well, and here's a little bit of their own agenda. So they said: "About the FBI: Are you reading this? If you are, then this is a perfect real-world example about why your idea of backdooring cryptosystems with a 'secure golden key' is very bad." Well, no, that's not what this is at all. But then they continue:

"Smarter people than me have been telling this to you for so long, it seems you have your fingers in your ears. You seriously don't understand still? Microsoft implemented a 'secure golden key' system." No, they didn't. "And the golden keys got released from MS own stupidity." Well, no, they didn't. "Now, what happens if you tell everyone to make a

'secure golden key' system? Hopefully you can add two plus two. Anyway, enough about that little rant, wanted to add that to a write-up ever since this stuff was found."

So, you know, these guys have an agenda, and this was a platform for allowing them to express themselves. Unfortunately...

Leo: Yeah, I mean, we share their agenda, but it doesn't have anything to do with this.

Steve: No. Not at all.

Leo: Now everybody, including me, including everybody I ever talked to, misunderstood this. Thank you for setting the record straight. I wish you would call everybody and tell them. Because I haven't seen one article saying, "No, no, that's not it." I guess it is kind of hard to understand, frankly.

Steve: Well, think of how fabulous this would be if it were true.

Leo: I know. I know.

Steve: And, I mean, it's not true. So, and now the problem is I don't know how Microsoft mitigates this. I mean, this is - I don't mean to downplay this. This is a big mistake because this does allow non-updated boot managers, pre-anniversary update, to be fooled with some pieces that are from the anniversary update. So this was, I mean, this is absolutely a mistake. But what it wasn't was the disclosure of a golden key. So anyway, and believe me, I don't blame the press for not digging in because it was hard to determine that that's what this was. And then when I thought I understood it, then I read it again several times to make sure I was right. It's like, okay, yeah, that's what these guys did. Which is nice work. But the whole golden key is an absolute red herring, referring to the notion of backdoor systems. But this isn't that. This was a mistake.

Leo: Right. Microsoft has put out some fixes, the ROS and Slipstream. The hackers who discovered it said those don't do anything. They say in their write-up - now, I'd like to get some clarification on this, too - that Microsoft really can't fix it because it would break these older systems. Is that the case?

Steve: My point. That's my point. It's like, I mean, we have an enduring problem.

Leo: Right.

Steve: What this would - the only thing you could do, or Microsoft could do, would be to release, securely release an update to the boot manager. That they could do. If they brought, that is, for people who didn't, for whatever reason, didn't want to update to the anniversary edition of Windows 10, Microsoft should at least, and I imagine they will, I think we can foresee this, there will be an update to all Windows systems that support

secure boot. They could be updated for awareness of this new supplemental policy system, and that would then shut everything down. But that of course requires action from the entire industry of users. So it's a big problem, but it can be fixed. But it does require that the existing boot managers be taught about these changes. And it just slipped by, it just slipped underneath Microsoft's radar.

Leo: And it doesn't affect, like, main line of business stuff for Microsoft, either. So there's not a huge incentive to fix it. You point out, correctly, it's not about installing Linux on it, it's that a rootkit or something else could use it to modify the operating system.

Steve: Right. Well, and in - oh.

Leo: What? A yabba dabba do?

[Yabba dabba doo]

Leo: Wait a minute. Did you have a precognition that you were going to get a yabba dabba doo?

Steve: No. What happens is...

Leo: Does the hard drive make a unique eh-eh right before it plays yabba dabba doo?

Steve: I have a real-time monitor to GRC's servers. So that gave me the first indication.

Leo: Oh, my.

Steve: Then GRC's servers sends a text message to my phone, which is tied with iMessage to all the other devices.

Leo: That's so funny. That's so funny.

Steve: So I did hear it first. And actually that was a corporate purchase.

Leo: Nice.

Steve: That was four yabba dabba doos. So whoever that was, thank you very much. If you're listening, I appreciate it.

Leo: Probably somebody listening to this, saying Steve Gibson's the only guy that got this whole story right in the whole world. And thank you, Steve. Here, I'm going to buy a corporate license. That is awesome. So I just want to clarify, because I'm going to, I mean, I now have the obligation to, in every place where we said this, including Windows Weekly tomorrow and on TWiT on Sunday, to say we got it wrong. This doesn't do that. This isn't a golden key. This just involves using bits and pieces of the Redstone update to modify the secure boot in older versions of Windows, like Windows RT and Windows Phone.

Steve: Correct. So, yes, the supplemental policy pieces that were added, that technology added to Redstone is a means of permitting developers to install test code.

Leo: Ah, right, right.

Steve: Essentially to allow secure boot to believe that their test code is valid.

Leo: For development purposes, yeah, yeah, okay.

Steve: But there are things missing, some device ID pieces and other pieces, that just aren't necessary there. But the updated secure boot knows about that, which is what makes that safe. Yet what these guys discovered is you could take those supplemental policies, and they would work on older versions of secure boot across the board, allowing anybody to use that to install their code, basically to completely subvert any pre-anniversary update secure boot technology.

Leo: Got it. Okay. Well, I will fix that. And I did get - it's not that I didn't try to read the original post. And I did. But I don't have the skills you do to parse it, and I didn't understand a word. So, not that it was very well written.

Steve: And it even jumps around, you know. They had a lot of fun doing it. But, like, the text jumps around. So it's like, okay, wait a minute, wait a minute. What? What?

Leo: Yeah.

Steve: And so I copied it to a text file.

Leo: Well, lesson learned. When it comes to highly technical subjects, even the tech press often gets it wrong.

Steve: Again, you couldn't have dangled a bigger golden carrot in front of the press.

Leo: Right. We all wanted to believe it. Right.

Steve: Oh, yes. I mean, it's a fantastic story. It's just not true, unfortunately.

Leo: Yeah, yeah.

Steve: So, anyway, that's why we have the podcast.

Leo: Thank you.

Steve: I got a kick out of you talking about this on TWiT and needed to talk about the foreseeable ad blocking-unblocking wars which are now underway. And I appreciated you noting that it was easy to block third-party ads. We should back up a little bit. This is Adblock Plus versus Facebook. And as we know, Facebook's revenue model is advertising. And so Facebook is not happy at the idea that users would be empowered to block ads on their devices. And the response on Sunday's TWiT show was universal about ads on phones. I mean, it's just such a problem there on mobile devices that the ad blocking has been a major win.

So Facebook has worked around Adblock Plus's blocking, and the news is that Adblock Plus has now worked around Facebook's unblocking of their blocking, and back and forth. And what you properly noted was that blocking third-party sources is trivial because most ads have traditionally been pulled from a domain other than the page's domain, the site from which the page was pulled. But it is the case that one of the ways - and we always knew this was possible is that, if advertising came from the same domain, that would make it more difficult. But of course developers are going to just naturally have different URLs. Maybe there'll be a subdomain, you know, ads.facebook.com, for example. So it's from the same primary domain, but a different subdomain.

Well, clearly, by default, something just blocking third-party ads would permit those. And so ads appear. Then the adblocker says, oh, well, we're going to block ads.star stuff. And in fact there are some rules of various sorts that do that. And so then Facebook says, oh, okay, fine. So then they stop using domains, and they put something down in the URL path, but probably still looking different than non-ad content. It'll have a different URL pattern. And so, again, I mean, the problem is Facebook is fundamentally trying to do something probably impossible. And that is, they are trying to force what the browser client shows its user. And that really isn't a level of control that this system was designed to offer.

The only way I could imagine it could be done, and it would kill so many other things, and that would be just to turn a page delivery into a scrollable PNG image. So basically you go to a page, and what you get is one pre-rendered PNG image that you then scroll down. And then there's no way to block it without taking it all apart and masking it and things, which would be another level. But fundamentally, as we know, the HTML goes to the browser. It parses it and finds lots of references to images and things and then requests those. But that's at its choice.

And so what the ad blocking does is put a filter in that action, that secondary fetching, to say, eh, let's not get things that look like this. And so this back-and-forth continues to

proceed. And, I mean, you could even imagine it going as far as, you know, if ads had fixed sizes, then an adblocker could say, oh, we're going to block images of those sizes. And so then the ads start having fudgy sizes, and back and forth. Problem is I just don't think that Facebook is going to be able to win this one. It'll be interesting to see how this goes. But this was entirely foreseeable, and it's happening just like one would imagine.

Leo: Yeah, and I don't know how you end this; right? It's just round after round of escalation.

Steve: Yup. This is one where...

Leo: I like the idea of a solid PNG. I hope they don't do that. But that would work; right? That would - there's nothing you can do about that.

Steve: Yeah, although you'd lose all of the dynamic functionality.

Leo: You have no interactivity, yeah.

Steve: Yeah. And so unless you did some sort of an interactive overlay on top of that - and actually, I mean, the document object model, the DOM in today's browsers is powerful enough that you could do that. You could pre-render the page with the ads in place and then overlay hotspots and control some things on top of it.

Leo: You wouldn't have AJAX-y stuff, but...

Steve: Yeah. Maybe it'll happen. So tell us about, Leo, the woman that called with her Avast problem.

Leo: Yeah. I wanted to run this by you. And I've talked to some people since who confirmed my theory, although it's never...

Steve: I listened to you, and I thought you were dead-on.

Leo: Yeah. Rose was a community manager for a local synagogue and used Facebook to post information about events and so forth. And at some point her Facebook stream was hijacked, and Turkish, spammy links in Turkish to porn sites started to fill her feed. Of course immediately most of her friends unfollowed her. They didn't want to see that. And the synagogue probably did the same. And she went - she did everything I thought she should do. We went - it took me half an hour with her. I did 15 minutes on the air and then I stuck around during the news break to talk some more about it. Took me a while to figure out what was going on, or at least a theory about what was going on.

And she did all the right things. She went through her Facebook connected apps and disconnected everything. She changed her Facebook login. She did everything she could - she brought it to a, she said, Israeli security guy, and he cleaned it, ran Malwarebytes and stuff. And I was very puzzled because the thing that really triggered a warning bell for me is she said every time I go to Facebook, it says you have to change your password. And she would not - I said, well, did it then email you or call or send you a link? She said, no, it was right there, it just says "change your password" on the Facebook page. And I thought, that is not normal behavior. That's not what I would expect from Facebook.

So finally, after quite some time, I should have done this right away, I said, well, you're on - look at the browser bar. You're on https://facebook.com. Yes. Is it green? Yes. Click the padlock. Okay. Does it say you're on a secure site? Yes. Who's the certificate by?

Steve: Nice work, Leo.

Leo: Avast.

Steve: Yup.

Leo: Avast, which is of course an antivirus company. And I dimly remembered us talking about a flaw.

Steve: Yes.

Leo: First of all, Avast, and this is not unusual, I think other antivirus companies do this, and other security companies, put a man-in-the-middle certificate in so they can scan all the traffic, SSL traffic.

Steve: They have to now.

Leo: Yeah.

Steve: If they want to look into your secure connections, that's the way to do it.

Leo: Right. Not the way you want them to do it. And that's what Superfish from Lenovo did, and others. So that was a man in the middle. But we had talked about the fact that that particular - I believe the Avast root certificates had leaked.

Steve: They lost their private key, yes.

Leo: That's what I thought.

Steve: They published it by mistake.

Leo: So what I assumed - and I never got confirmation from Rose. But what I told Rose is, well, first of all, get Avast off of there. And you don't want to go to any site that doesn't say the certificate comes from that site. But what I in the back of my head was assuming is that some Turkish actor had figured this out and was doing a man in the middle to her using the Avast certificate and hijacking her Facebook traffic.

Steve: And the moral of the story for our audience is this is the problem with a third-party interception. I would argue that, in a corporate setting, the border proxy which is intercepting secure connections is on balance a good thing because the corporation has a need to protect their networks and their users from all incoming traffic. And the only way to do that is to look inside those connections. And in a corporate setting you're using corporate facilities, corporate bandwidth, and so forth. So employees need to understand they have no expectation of privacy when they're using the corporate equipment. And IT has an obligation, I think, ethically, to make it clear, you know, to print that in a little notice that's stuck on the top of the monitor to remind people that communications with this computer are being screened for security purposes.

In that setting, though, we've got hopefully professional-grade IT running real well-designed, hopefully well-designed hardware screening this. The problem with changing that model to everybody's PC is pretty obvious. And that is, suddenly that same technology which is very powerful is operating in your computer. And we've talked about how the unfortunate side effect of this next generation of very intrusive antivirus is it increases the attack surface. So it could very well be, you know, who knows. Maybe every email she sends out has a little tagline added, "Scanned by Avast AV," and somebody, I mean, what you're basically doing is you're broadcasting the fact that there's this, like, what your technology is...

Leo: I'm using Avast, yeah, yeah.

Steve: ...in your machine. And if any flaw is then found, somebody sends you back a piece of email, knowing that Avast is going to scan it. And if there's any problem with what they've got running in the kernel, that's all it takes. And we've seen this. This is not theoretical. This happens. You receive email, and it takes over your machine.

Leo: What is the, for a less sophisticated user like Rose, what's the remediation that you would recommend? I mean, she could remove the Avast certificate from her authorities, but that's maybe an advanced thing. Uninstall Avast? Would that be sufficient?

Steve: I would uninstall Avast. Although the problem is, if that cert is there...

Leo: That cert's now in the accepted cert authorities.

Steve: In the root. It's in her root list of trusted certs. So you absolutely, I mean, and you're right, Leo, this is a problem. It's because, as you said, she's doing what she can, but...

Leo: She got a so-called "security expert" to look at it. Why he didn't notice this is another matter.

Steve: Yeah.

Leo: So, yeah, this is a tough one to fix because she has to know enough to go into the root certificates and purge Avast.

Steve: And that's my point is that this is high-power technology...

Leo: That shouldn't be used.

Steve: ...that you could argue should not be occurring on systems. The AV people have been forced to do this by us all switching to encrypted connections. I just think staying with Microsoft's solution is probably the right move.

Leo: Yeah. Wow. Anyway, yeah. She was using Chrome. She turned off all browser extensions. She did, I mean, she did everything I would have said. But, I mean, my god, that's such a deep hook into her system.

Steve: Yeah, and Chrome does use Windows - Chrome doesn't have its own security suite. It leverages the security platform of the machine it's on. And in Windows it is using Windows CA root trust store. So that all tracks.

Leo: Well, I'm glad you were listening, and I didn't say anything really stupid.

Steve: No, you did a great job.

Leo: Whoo.

Steve: And it's nice, too, that she's able to click the padlock, dig in, figure out who the cert came from. And you immediately recognized, whoops, that HTTPS cert should not be signed by Avast. They should be signed by wherever you're visiting.

Leo: And as you know, I then took the next segment as a chance to kind of explain the certs and how to check the certs. And I wish everybody knew that. If you're a geek listening to this, and you haven't shown your parents and your friends how to check to make sure the cert is what it says it is, you know, it's more than just looking at the address bar. It's going one step farther, yeah.

Steve: Yeah. So there was an interesting, another news item that I wanted to clarify. Again, it made headlines because it seems to affect so many devices. But what's actually behind it is a much more interesting story. And so, for example, ZDNet had coverage saying "Linux traffic hijack flaw affects most Android phones and tablets." And then they explain a difficult-to-exploit flaw affects all Android phones and tablets that are running Android 4.4 KitKat and later, which comes with the affected Linux kernel 3.6 or newer. That's 1.4 billion devices, including the developer previews that are out now of Nougat.

So what's behind this is really interesting. That Linux kernel 3.6 was released in September of 2012, so just about four years ago, coming up on four years ago. And the developers deliberately added support for a very recent, relatively, update to the core TCP protocol, in order to tighten it up against some known attacks. Unfortunately, in doing so, these researchers who published a paper at last week's 25th USENIX security conference demonstrated a way for what's known as a blinded attacker, that is, an attacker that isn't a man in the middle, has no visibility, but does know the IP addresses of two communicating endpoints.

It turns out that, as a side effect of a change in TCP, and it's like RFC 5912 or something, I mean, it's a high-numbered, late-model, request-for-comment document that has part of the TCP spec. As a side effect, there's a way of an attacker to shut down communications, but without being in the middle, just knowing that two endpoints are communicating, and their IP addresses. And if it's non-secured, that is, if it's, for example, an email exchange, then it's even possible for the attacker to inject their own data into the flow.

And so what was interesting is that, well, first of all, it ends up being - these guys did a beautiful job and leveraged their insight into something quite powerful. In the abstract for the paper they said: "In this paper we report a subtle, yet serious, side-channel vulnerability, introduced in a recent TCP specification. The specification is faithfully implemented in Linux kernel version 3.6 and beyond, and affects a wide range of devices and hosts.

In a nutshell, the vulnerability allows a blind off-path" - that's the term I was looking for - "off-path attacker to infer if any two arbitrary hosts on the Internet are communicating using a TCP connection. Further, if the connection is present, such an off-path attacker can also infer the TCP sequence numbers in use, from both sides of the connection. This in turn allows the attacker to cause connection termination and perform data injection attacks. We illustrate how the attack can be leveraged to disrupt and degrade the privacy guarantees of an anonymity network such as Tor and perform web connection hijacking. Through extensive experiments, we show that the attack is fast and reliable. On average, it takes about 40 to 60 seconds to finish, and the success rate is 88% to 97%." Which is, as our listeners of the podcast for a long time know, that's way up there in terms of attack reliability.

"Finally," they say, "we propose changes to both the TCP specification and implementation to eliminate the root cause of the problem." So essentially this is another "security is hard." The brightest minds in the industry, you know, we don't have college

interns changing the TCP specification. The guys that are changing TCP are, I mean, really know their stuff. Yet, even so, mistakes get made.

And the real problem is - we've talked extensively about TCP in the past, I mean, because it's just a lovely protocol. But the problem is we come back to the original concept of the Internet, where security was an afterthought, or actually it wasn't a thought at all. It was, oh, my god, this works? Just the idea that it worked was a miracle back then, that you could use these autonomously routed packets, where it's just a packet of information that has a destination IP address, and then a grid of loosely connected and not even reliably connected routers are able to forward packets toward their destination. And if you do that enough times, you get there. And the packets contain the IP of where they came from, which allows the recipient to respond. And that's the Internet.

Yet it's like, unfortunately, it also permits all kinds of mischief. And notice that, as I mentioned, you could do data injection if it wasn't encrypted. But even encrypted payloads are still carried in an IP and TCP wrapper, which fundamentally have this problem. And so, for example, I'm not going to go back in and do a tutorial on TCP. We've done that already. And if anyone's interested, it's there in our archives. But the data being sent in both directions, the bytes are numbered in a monotonically increasing sequence, which wraps around, it's a 32-bit count, so it wraps around after 4.3 billion. But it doesn't start at zero. It starts at a hopefully random place.

So once upon a time the starting points weren't random enough. And so knowing where the sequence number was, was enough to allow somebody to spoof. Essentially, that's what this is. It's a way of spoofing. And when you think about it, you've got a worldwide network of loosely connected nodes. The only thing that identifies a packet is, like, while it's being routed, is the IP that sent it and the IP that it's going towards, its destination. But anybody else can drop a packet on the Internet that spoofs its source IP. Which means that the recipient has absolutely no verifiable means to know where it came from. That's the crux of the problem, when you've got this loose confederation of routers that are simply - they don't care what's in the packet. They just send it on its way.

What that means is that, if somebody else can - well, essentially it means that anybody in the world can drop a packet onto the Internet with made-up information, and it will arrive at the destination, and there is no way, none, for the recipient to know that it didn't come from the source. Now, one of the ways that spoofing has been mitigated is with the so-called "sequence numbers," that is, it's known as a TCP window, which identifies the allowable range for packets coming in. And packets that fall outside that narrow range relative to 4.3 billion, that whole 32-bit sequence number space, they're just discarded.

And so what these guys did was they figured out a way to, blindly and off the path, to be able to send, leveraging some features that were meant to increase the security of TCP, they could leverage those against security in order to obtain information from the endpoints, enough to allow them to get port numbers and sequence numbers. And if you know the IP address, the port number, and the sequence number, that's the only disambiguating information that the packet contains, you can then fool TCP. So here we are, what, decades from the time this was created, and the smartest people we have are saying, ooh, let's update TCP in order to fix some problems. And they ended up doing just the reverse by mistake.

Leo: Oopsies.

Steve: So, yes. So they conclude in their paper, they say: "The contributions of the paper are the following: We discover and report a serious vulnerability unintentionally introduced in the latest TCP specification, which is subsequently implemented in the latest Linux kernel. We design and implement a powerful attack exploiting the vulnerability to infer, first, whether two hosts are communicating using a TCP connection? and, second, the TCP sequence number currently associated with both sides of the connection. We provide a thorough analysis and evaluation of the proposed attack. We present case studies to illustrate the attack impact. We identify the root cause of the subtle vulnerability and discuss how it can be prevented in the future. We propose changes to the kernel implementation to eliminate or mitigate the side channel."

Oh, and patches do exist. They were released on the 11th of July, so more than a month ago, but they won't be pushed out into Android, it's expected that they'll be part of the September updates for at least the Nexus 7 phones, along with the other batch of problems that are going to be fixed next month. So, really interesting.

Leo: They said in the article, one of the articles I read, that Google was not treating it as a major security flaw. It's hard to do. It's a flaw, but it's not - it's a nontrivial thing to do; right?

Steve: Correct. And virtually all of our communications is over TLS now. So you can't...

Leo: Right, anyway, right. This is why you want - you've now convinced me that everyone should HTTPS everywhere; right? That's another reason why.

Steve: Yeah. We have the processing power to handle the encryption. We've got certificates thanks to Let's Encrypt. There's just no reason not to do it. And so it's not even just for privacy. It's like for this kind of little edge case issues that you would never expect. Something I don't have in the show notes but was sort of on my mind last week, there was a report in the U.K. about surveillance vans that were going to be driving around and were able to reportedly determine if somebody was pirating some subscription content from a service. It's not in the notes. And there was a lot of speculation about how that could possibly be done.

So I just wanted to mention that - oh, and apparently they would be using WiFi, that is, like sniffing on the WiFi. That caused a ruckus because the presumption was, very much like Google Maps famously did by mistake where they were capturing packets, and they said no, no, no, we're not doing any data interception. And so there was a lot of conversation about, well, how is this possible? How could that be done? Well, and so I just wanted to comment that, if this surveillance van is from the company that is offering the service, then they're watching the recipient and wanting verification that they're receiving subscription content.

And this is very much like the Tor deanonymizing. Remember that, as we've talked about with Tor, while it's difficult to obtain an identification of endpoints outside the network which are communicating, it is trivial to confirm it. So if you have a suspicion, then it's trivial. All you have to do is, for example, block the traffic for a while and see if your suspicion is borne out because the other end suddenly dries up, and then you let the traffic go. And, oh, the traffic comes back again. So you can't absolutely know in that instance. But you can have a pretty good idea.

So, for example, if some - if this van wanted to catch people who were pirating this content, they would have the ability to know the IP address of where the van is parked in front of, passively receiving encrypted WiFi. They could do something, for example, like for that IP at the sending end, change the packet size. Or change, you know, make every 10th packet half size. And that pattern would emerge at the receiving end and confirm that the data stream was coming from that source. So again, in trying to have a level of privacy that the Internet absolutely was never designed to provide, you can't have it. You can want it, but you can't get it because the technology will work against you. It just - it wasn't built to offer that.

Leo: All right. Back to filling our brains with Steve's knowledge.

Steve: Some nice news for people using, staying, who have chosen to stay with Windows 7 and 8.1. Microsoft last week, I think it was, announced that they were going to basically learn from the success of the Windows 10 patching and updating model and begin adopting that starting with the October updates of, in this year, 2016, for Windows 7 and 8.1. So what that means is instead of what has traditionally been, what we've been talking about since the beginning of the podcast, is every second Tuesday of the month from the point they began, I guess it used to just be at random times, then they went to the second Tuesday because IT departments were going crazy with unplanned, unscheduled patch releases. They, as we know, it would be many individual patches. You would run Windows Update. You'd see a list of 13 things. If you were curious you might poke around and look at them.

That's going away. So instead of essentially what are a handful of individual incremental updates, all updates will be merged into a single monthly, one monthly update blob. And from starting in October, each successive month will add to that existing new patch base, offering one single update. Meaning that, if you missed a couple months, all you ever need is the latest one because it will automatically back-incorporate, starting in October, back to October, anything that has happened since.

And then, over time, Microsoft will also move this blob's updates backward in time. They're probably just doing it carefully to make sure that they don't break anything. But the point is that that will incorporate earlier and earlier patches until eventually, and we don't know when, but eventually the monthly update will be able to take an original last edition of Windows 7 or 8.1 - in the case of Window 7 it would be Service Pack 1, the last official image, and bring it current.

Now, of course, as we know, Windows Update itself has changed in such a fashion that, as I mentioned before, if you install a new version, a new build, or the last build of Windows 7 SP1, it can't find any updates because the update server has changed. So you still need to do the Windows Update update, to update Update. And then it will be able to grab one blob and bring your system current. Now, that's all good, except that - and it sort of gives us a rolling mega update rollup. But we should note that it is removing some user control. I know that a lot of people, like the type of people who listen to the podcast, like to see - oh, 3:00 o'clock. Like to see - now you know why I normally mute these things.

Leo: That's actually kind of pretty. I wouldn't mind that. It's not, you know, it's just like a...

Steve: Yeah, I like it very much. So we like to see what's going on, and in some case choose not to do something. And of course the famous instance is the 3035583, which is the infamous Get Windows 10 update. Although I never endorsed the idea of avoiding it, because avoiding Windows Updates is just difficult to do, I know that many people kept, you know, every single month Microsoft would offer 3035583 for their machine, and they would turn that off, saying, no, I don't want that. And so that was, while burdensome, we did have control. So this will - we will lose that. It'll end up being monolithic. You either stay current, or you choose not to.

On balance, I think it's probably a good thing. And just from a pure technology standpoint, when I've dug into individual updates' contents, and I look at all the various DLLs and .sys files that they make tiny tweaks to, and then you imagine all of that overlapping, and all of that overlapping all the way back through time to the beginning, I have no idea how they ever made any of this work. It's like, I mean, how can you say I don't want the 3035583 update, yet other things I do want later? Yet the functionality that that update brought somehow isn't put in by anything that happens afterwards. I have no idea how that happens. It just - it's amazing to me.

So this probably represents a huge improvement. I think it probably improves the stability of these systems going forward, again, at the expense of some control from control freaks who like to be able to say, I don't think I want that one. So, still, I think that's progress. And the other nice piece of news is that - we all know that I frantically built my - I can't think of the name, the Intel chipset with an H, Haswell. I built my Haswell-based Windows 7 machine immediately upon learning that Microsoft was not going to be supporting Windows 7 and 8.1 on the Skylake chipsets. And so it's like, oh, shoot, I need to protect myself from that. Well, then of course we've covered the news that they changed their mind.

Well, they have just changed their mind again, relative to security patches. It was going to be, they originally said in January of this year, that the Skylake-based PCs would not be getting security patches for non-Windows 10 platforms after the summer of 2017, so a year from now. Then they got some pushback, putting it mildly. So they changed it to 2018, the summer of 2018. Still pushback on that. So they finally said, okay, fine, we will support Skylake through the natural end of support for the platforms. So that's, of course, January 2020 for Windows 7 and 2023 for Windows 8.1. So anyway, anybody who is installing Windows 7 and 8.1 on newer hardware will have it supported, thanks to this change in policy all the way through the end of life of updates. So that's good news.

Okay. This is just the coolest hack. As a developer myself who is often needing to look at packet traffic, I have to infer the contents of TLS connections which Wireshark captures for me and displays. I can see IPs and ports. I can see where things are going. I can see the initial identified handshaking packets. But I can't see any contents. And contents is often valuable. In the old days, I would simply, for example, if I was working with GRC's servers, I would use HTTP so that, even if we would normally use HTTPS, I'd just use HTTP for testing because I could see into it.

That's all gone because it's no longer safe, as we were just saying, to use HTTP. And all of the browsers now know that GRC is always secure because we have the HSTS headers, the HTTP Strict Transport Security, which informs browsers to autonomously change any non-secure to a secure connection. So HTTP is silently promoted to HTTPS. The server sends that header with every reply to train every browser that touches GRC, and of course many other sites, too, that are now supporting HSTS, that secure is the only way we want to connect. That means, though, that I'm no longer able to see, for diagnostic and development purposes, what's going on.

Turns out there is a way, and I didn't write down the person who tweeted the link to me. But I have in the show notes the link to the write-up, a detailed write-up of the way to decrypt TLS browser traffic with Wireshark. It turns out that developers can place an environment variable in their system, SSLKEYLOGFILE. So you set the SSL key log file to a specific filename. And Firefox and Chrome, only those two, but who needs anything else, will honor that environment variable. And here's the key. Every TLS connection the browser establishes, it logs the cryptographic keys for the connection.

So again, we've covered this extensively in the past, where the way TLS happens and the way the negotiation occurs with the server providing some information, the client providing its random information, and a pre-master secret and then the master secret. If you had that information, you could then take an encrypted stream and decrypt it. Basically, that's the information the endpoints each have that allows them to encrypt and decrypt the communications for their own purposes. And Wireshark knows how to read that file.

So what this allows you to do is capture traffic with Wireshark with this environment variable in place, and then use the keys that were captured, and Wireshark is now smart enough to decrypt it and allow you to see the plaintext inside. There was some - this was sort of fascinating, some back-and-forth with this most recent release of Firefox. We're at Firefox 48. They almost took it out because there was concern that there was some - that logging the cryptographic keys to a file was a security vulnerability.

The argument, which I concurred with, was wait a minute. Yes, you're exporting it from the browser. But if there's somebody - if your system is already compromised to the level required to leverage that, then all bets are off anyway because there are, you know, you can just simply drop a filter into the TCP stack. There's, at least in Windows, I'm very familiar with the layers of that. And there are all kinds of - remember LSPs, Layered Service Providers, Leo? We used to have a problem with those.

Leo: Don't remember that.

Steve: That kind of went away. But it still exists. So it's easy to stick shims in and intercept traffic. It's just sort of a mess, but you could do it. And so that argument won out. And Firefox, they did not move it to a compile time definition, a def that you would have had to set to build your own custom Firefox that had that capability. They left it in the main production release. So anyway, just a very cool hack. Again, I've got the links in the show notes for anyone who's interested.

And I should mention that the link farm page, for anyone who hasn't looked, it's where I'm continuing to put things: GRC.com/linkfarm. And it's growing. I've got a bunch of static things and a list of the puzzles and toys that this community has been exposed to and loves. And then based on various podcast numbering, I pull a bunch of relevant links out for each podcast. So don't forget to check GRC.com/linkfarm. And the show notes also have specific links to this.

Also I wanted to point out that we've been talking about the Ubiquiti EdgeRouter, and before that the Cisco SG300, and also pfSense. PC Perspective dotcom, PCPer.com, P-C-P-E-R dotcom, put together a very nice write-up with very nice diagrams explaining all of the evolution of the three-router network isolation approach, the so-called Three Dumb Routers solution. And even the comments after that posting were useful. So I've got a link also in the show notes, and also in the link farm, on the link farm page, because I think for some people that makes sense. That is, you may not want to get all into the

sophistication required to set up individual subnetworks on interfaces on something like the Ubiquiti EdgeRouter X, or deal with a system with multiple interfaces running pfSense. You just want the solution. Or you may have a few old and retired blue box consumer-grade routers sitting in a closet, where you can just plug them together and get the equivalent strength.

So I really appreciated the write-up. And I also note that we need the same thing for the Ubiquiti EdgeRouter X and for pfSense. So if anyone is interested in detailing their IOT network segregation solutions, write it up and bring it to my attention, and I will share it with our community because I know that people here would appreciate a bit of a how-to guide.

Leo: And that PC Perspective piece was probably written by Allyn Malventano; right?

Steve: Isn't he Perspectives Plural?

Leo: No, no, that's him, PC Perspective. And it's PCPer, they host This Week in Computer - Brian Shrout, the publisher and editor-in-chief, hosts our This Week in Computer Hardware.

Steve: Yes, It wasn't written by Allyn. It was written by somebody else there because I did look [crosstalk].

Leo: Okay. You and Allyn had corresponded, I know, over that.

Steve: Yeah.

Leo: Okay. Yeah, that's the same site.

Steve: Cool. Anyway, very nice piece of work.

Leo: Good, good, good, good.

Steve: So it turns out we talked last week about this very worrisome Windows SMB - the so-called Server Message Block - credential leakage, which allows, if you are using a Microsoft browser, IE or Edge, allows a malicious party to put an SMB Windows filesharing-style resource on a browser page. Or I forgot to mention also Outlook. You can receive - that's also supported there. So if they send you email that has a blob, and you're using Outlook to open it, it will do the same thing. These Microsoft clients will initiate an outbound connection across the Internet,, and part of that is your username and the hash of your password. So if that's not - if the password is not super strong, I mean, and I mean super strong, then that can be a problem.

And of course LAN Manager passwords, as they're called, have notoriously been weak, sort of fundamentally weak. And ages ago we dealt with lots of problems with those. And

this is a concern because people are now using their Windows credentials to log onto Microsoft properties, so it's not just your own machine that you'd be logging onto. I think this is why Microsoft didn't worry about this traditionally is that there wasn't really anything somebody remotely could do. Now there is.

The point is that I got a tweet from someone, Donn Edwards, who noted that he'd been playing around with this mitigation, and it has got showstopping side effects. His tweet said: "Hi Steve. The 'fix' [in quotes] for IE usernames mentioned here" - and he refers to the BleepingComputer.com page that I referred to last week - "causes more problems than it solves." And there's a registry setting change which I referred to last week, "RestrictSendingNTLM" - that's NT LAN Manager - "Traffic," and you set that to two.

And he says: "This effectively isolates the PC on the LAN" - so not even the WAN, but the LAN - "so that it cannot see or connect to any other PC or fileshare on the LAN." Yeah, that would be a deal-breaker for me because my Drobo would go offline and become inaccessible. "You have to enable connecting with every other PC or server as an exception" to that policy. He says: "It also interferes with Remote Desktop connections. So even if you connect with Remote Desktop and use saved credentials, you still have to input the credentials again. Please could you alert users to this issue, since the article itself is not particularly clear or explicit. Keep up the good Security Now! work."

Okay. So I wanted, yes, to let everybody know, but also to note that, first of all, if your ISP is blocking access to those ports, certainly Cox does, and a lot of ISPs do because of the traditional problems with Windows. On the other hand, I would argue that those have largely been solved with firewalls which are now running since XP SP2. Or was it SP3? It's been a while. I forgot. Might have been 3. Oh, no, SP2. Yes, because I just recently upgraded to SP3.

Leo: Oh, yes, I see.

Steve: Ah, yes. In order to get SHA-256 signature awareness. So, yes, SP2 with XP brought the firewall on by default. Of course everybody's now behind a router, which is providing protection. So the fact that the problem no longer exists means ISPs could be forgiven for dropping those port filters. The right solution is for us to filter them. And so this is another perfect application for a smarter router. That is, if you were using the Ubiquiti EdgeRouter or pfSense, you could absolutely firmly block ports 137 through 139 and 445. That range of three, 137 to 139 - so 137, 138, 139 - and 445, those are the ones where all of this server message block NT LANMAN stuff happens. If those are blocked at your own interface between your network and the Internet, then you're protected, and all of your goodies inside can function, and any query that any Microsoft browser or client in your network attempts to make, it'll absolutely fail at the border. So, and then if you had some need for specific outreach, you could permit that.

And so digging into this a little bit more, that led me to an interesting page because I was kind of curious about where pfSense stood with this. And of course Universal Plug and Play is a constant concern from a security standpoint. So I got a kick out of what they said. They said - this is the documentation for pfSense at pfSense.org - "UPnP is short for Universal Plug and Play and is commonly found on Windows, BSD, and Linux systems. NATPMP is short for NAT Port Mapping Protocol and is similar to Universal Plug and Play, but found more commonly on Apple devices and programs. A growing number of programs support both methods. pfSense supports both, and the service may be configured at Services > UPnP & NATPMP." So that's all available for pfSense users.

"UPnP and NATPMP both allow devices and programs that support them to automatically add dynamic port forwards and firewall entries. The most common uses are in gaming systems (Xbox, PlayStation, et cetera) and BitTorrent programs like uTorrent, which both rely on allowing inbound connections to a local service."

And then they have a bar and warning flashing. And they said: "WARNING! Potential Security Risk! If UPnP or NATPMP are enabled, use only devices and programs which are trusted. These mechanisms will allow these entities to bypass the firewall to allow incoming connections with no additional control or authorization." And I love it, they said: "Do not be surprised when this happens. Access permissions for the service may be crafted in the options on pfSense. The format of these is shown in the GUI at Services [and again] UPnP & NATPMP in the user-specified permissions boxes. Using these, access could be restricted to a specific workstation or device."

And so that's an advantage, an example of the kind of power that one of these more capable routers gives you is you could enable UPnP specifically for your Xbox, just because you want to do it that way, that is, use UPnP and have it configure the network itself, but still have it not generally available. And in this world of Internet of Things, this is something I'll be very surprised if the IoT devices aren't just saying, oh, I'm going to open myself a port so that China can access the light bulb whenever it wants to. It's like, okay. Bad idea.

And one last link here. There was specific information about UPnP with the Xbox, and I've got a link in the show notes which, for people who don't want to use Universal Plug and Play, but want to statically map some ports, that provides some guidance for doing that with pfSense.

And finally, before we get into a little bit of miscellany, yet another way of exfiltrating data from a computer. Yes, the hard drive's head movement sounds. It turns out that the same guys we covered a couple months ago, who were extracting data through a wall, remember they had like an antenna on one side of a wall and a laptop sitting on a table on the other side. And they were able to get the data. They were able to extract a key in use from that machine. It turns out that we now have DiskFiltration, which of course is hard drive information exfiltration. They call it that because it uses acoustic signals which are emitted from the hard drive, which is also known as sound to non-geeks.

Leo: Acoustic signals, yes.

Steve: Yes, an acoustic signal...

Leo: That's what we've been doing [crosstalk].

Steve: ...from the hard drive. Make sure I heard - I'm sure I heard something. And so what they're doing is they've got software in the machine which is manipulating the hard drive head actuator to generate sounds that can transfer passwords, cryptographic keys, and other sensitive data stored on the computer to a nearby microphone. So in practical terms, it's worked at six feet distant to a phone sitting on the desk. They were able to achieve 180 bits per minute, which, yes, you're not going to be sending any large texts or exfiltrating a database. But that's generally not necessary these days because we've become increasingly dependent upon encryption. And all you need is the key. And 128 bits per minute would allow you to exfiltrate a very strong 4,096-bit key in about 25

minutes.

So again, not super practical. But you could imagine a situation where, in a super secure environment, if there were some way to infect a computer that was believed to be safe because it was air-gapped, no WiFi, no network connection, it's sitting there, and only it knows what it has, if there were some advantage to be gained from surreptitiously obtaining 4K bits of data - and again, these days 4K can be enough. If it's a cryptographic key whose secrecy absolutely must be protected, this potentially allows you to break that wide open. So just, what have we got? We've got fan noise. We've got of course the speaker. We've got the hard drive. Basically anything that you can do on a computer to in any way affect the environment, through any means, can be used to send data. Really not a big surprise. But it's sort of fun to see this stuff applied in practice.

Now, okay. I have in the show notes a picture of this thing, which is the most beautiful piece of engineering I've seen in a long while. It is a Kickstarter that someone pointed me to. And unfortunately, it was instantly sold out because of its popularity and the fact that they're just unable to mass produce them. It's a five-page wooden book where each page is an intricate, beautiful wooden puzzle. And you have to solve each page successively in order to open that page of the book, to unlock that page to get to the next page of the book.

And anyway, it's just - I just wanted to show it to people, just for - I know we have a large following of puzzle lovers. None of us can get it, unfortunately, because every single version, the build-it-yourself, the buy a bag of toothpicks and make one yourself, every one of the Kickstarter variants is completely sold out. And I would have immediately grabbed one because they just look beautiful. And in their notes they note that, if it weren't for laser cutting, this would never be possible.

Leo: Oh, yeah. Can you imagine?

Steve: But it's the reproducibility. And it's just - it's beautifully, I mean, beautiful wood, it's just a wonderful-looking thing. So for what it's worth, all we can do is lust after it.

Turns out the mysterious ARP generator which was CUJO'ing my system, actually two of my three machines at Level 3, was the Intelligent Platform Management Interface.

Leo: Of course it was.

Steve: Of course. Unbelievable. This is that baseband processor that we talked about a few months back. I think actually it was the week that you were in Newport Beach, Leo, and I discussed it with Father Robert, but then we talked about it the following week, as well, essentially the backdoor that exists in all of these machines. And fortunately, those servers, they're beautiful Intel 2U servers with six hot-pluggable drives each, and dual Xeons and dual redundant power supplies, I mean, it's a beautiful box, and I have three of them. They have dual NICs. And this is the takeaway. Only the primary NIC is hooked up to the platform management. I verified there is no way to shut it down. Nothing in the BIOS lets me do it.

Those are all running FreeBSD. FreeBSD has an IPMI driver and something called IPMI Tool. So I was able to dynamically load the kernel driver and then use IPMI Tool to probe the IPMI, look at the status, see the fan spinning, you know, all the kind of things that

this technology allows you to do in terms of managing the underlying hardware. And then I got a clue because there was an option to use a secondary NIC as a failover from the primary. But there was no option to use the secondary instead. And I thought, oh. I wonder if that means that it's not being hooked by this. So I went into RC config and changed the interface to the secondary one, moved the plug, and all is quiet. Problem is solved. It was just that one NIC of the two. So that's another tip.

There was a lot of dialogue after we were talking about this, about just plugging in a third-party interface, which you can certainly do, into the motherboard if you wanted a network connection that would not be risking having this platform management as a potential backdoor if there were any vulnerabilities found in it. And we know how difficult it is to make this stuff perfect. But if your motherboard has multiple NICs natively, it may just be as easy as switching to the secondary or tertiary and so forth. The newer servers I have have five NICs.

Leo: Wow, wow.

Steve: So I don't know why.

Leo: All my new computers have dual NICs on the motherboard. It's interesting, yeah.

Steve: Yeah. So, and it's weird because one of them, the one that's running the oldest version of Free - oh, I verified it wasn't FreeBSD. I stopped it at the little devil screen before BSD was running, before it had booted, before it had loaded the kernel, before the network was up. And sure enough, that ARP noise was occurring there, and it was stealing from another IP in the system for itself, even without FreeBSD running. And so I thought, okay, well, it's not the OS. Which is nice because I didn't want it to be the OS, although it would have been fixable if it were. But I don't need two interfaces there, and now I'm only going to use the secondary one.

I got an interesting tweet that I thought I would pose as a puzzle for our clever listeners. Jimmy G., tweeting as @TheRedTech, he sent actually a tweet both to @Naked Security and @SGgrc. He said: "I just bought a USB" - and he means a thumb drive - "from a friend, but he's a hacker of sorts. How would you safely format it, in case of a joke or worse?" And so I thought about that. I mean, because that could be deadly.

Leo: There's always BadUSB, too; right? I mean, he could have firmware stuff on there.

Steve: Yeah. So a Chromebook could probably do it safely, and Chromebooks will format USB thumb drives natively. They don't need anything added to them. And you could then power wash the Chromebook to, like, flush it from anything that might happen.

Leo: Just in case, right, right.

Steve: Yeah. And I would think that Unix would probably be a safe choice since it doesn't

have any of that autorun nonsense that mainstream OSes have.

Leo: You can format it without mounting it on a Linux system using DD, which is just...

Steve: Ah, that's exactly right.

Leo: Yeah, or your format. But you still wouldn't fix, if there's a firmware bug, I mean, if the guy's a mean hacker and put BadUSB on it...

Steve: Yeah, but it wouldn't be able to do anything. It would...

Leo: Well, yeah, but you couldn't use it anywhere else. You'd format it...

Steve: Oh, I see what you mean. Ah, very good point.

Leo: Doesn't make it clean.

Steve: Very good point. That's very true, Leo, because it is a little computer, as we know, that could get up to some mischief, yeah.

Leo: Right, yeah. To fix that I guess you'd need a EPROM program or something.

Steve: And I also ran across, just, again, miscellany, an excellent USB explainer. You were just talking about USB on MacBreak Weekly. And of course we know you're a fan of the Type C connector.

Leo: Love it.

Steve: LogicSupply.com is a nice site that actually is one of Mark Thompson's favorite sources of well-built, sort of above-consumer-grade, industrial-grade PC hardware of various sorts. They did a really nice explainer for the USB standard, taking us from USB 1 all the way through, not only 3.0, but 3.1, first and second generation. And talk about the differences in the physical connectors, the power delivery capabilities. I didn't realize that USB 3 could go 20 volts, which allows it to deliver as much as 100 watts of power. So that's substantial. And also the various data rates and so forth.

So again, link in the show notes, and I think on the link farm page. I think I put a copy of the link there for anyone who just wants to sort of just, if you haven't had a chance to focus on that, and you're interested, it's a bunch of good stuff. And then finally a tweet from John Adams. I was interested because his Twitter handle was @netik. And that's sort of a good Twitter handle. Turns out he's one of the early Twitter employees. And in a short bio he said, "I helped build this thing."

Leo: Wow.

Steve: Anyway, someone retweeted a tweet of his that I got a kick out of, and I knew our listeners would appreciate. He said - and my point is he's got some cred. He said: "There is no Internet of Things. There are only many unpatched, vulnerable, small computers on the Internet."

Leo: Yeah, that'll give you chills.

Steve: And that's exactly, I mean, that's really what it is. It's just like, we're in a wild wilderness at this point. And I got a nice note from a young student, David L. in Omaha, Nebraska. The subject was "SpinRite saved me once again - and our high school yearbook." I guess he's getting an early start on the yearbook. Or maybe it's photos from the summer or something.

He said: "Hello, Steve. I'm a high school student who really enjoys listening to you and Leo on Security Now! every" - well, he wrote Wednesday. Oh, I guess he listens on Wednesday because we record on Tuesday.

Leo: Yeah, yeah, that's when he gets it.

Steve: He said: "Love the depth you go into in your topics. Anyway, I was working on a paper due the next day on my desktop, and everything on my computer had been running very smoothly, as if nothing was wrong. I noticed that some updates were waiting for me to install. I didn't have time to install them right away" - probably because he was working on that paper - "so I decided that I would let them install when I head off to school. The next morning it boots up just fine, but when I come home" - or he says: "In the morning it boots up just fine. But when I come home, I'm greeted by a screen that read 'No active partition found.'" Ooh.

"Normally," he says, "I need to confirm to restart the computer, as it was sitting on the Windows Update screen when I left. I began to panic, as most of my schoolwork had resided on that computer, along with some photos that I needed to copy to a flash drive for the high school yearbook. Trying to figure out what to do, I was googling the error message. It turns out that others have fixed it through the Windows 7 install DVD. I didn't want to do that as I was afraid it might corrupt the drive even more. So I ran SpinRite. I selected my drive and ran it on Level 2. I let it run. About an hour or two later, I was greeted that SpinRite found no errors. I was prepared for the worst then, as I did not back up this computer," he said, parens, "(even though I should) as we do not have the best Internet speed." He's in Omaha, Nebraska.

Anyway, "I was afraid I was going to lose everything and let our yearbook staff down. I rebooted the computer, only to find it prompting me to select 'Last Known Good Configuration or Profile 1.' I selected Profile 1, and the computer started right up into Windows. I am in the process of backing up my data to a safe place and giving the important photos to our yearbook staff like nothing ever happened. Thank you, Steve, for your amazing work. David L., Omaha, Nebraska." And, David, thank you for sharing your experience.

And for what it's worth, this is one of those ounce of prevention issues. It's difficult to sell something that no one needs. But the other thing I routinely hear is from people who listen to the podcast, who purchased SpinRite and use it preventatively, you know, for preventative maintenance, just running it on their system to prevent it from getting to a point where it's in real trouble. And one thing I've never mentioned in all the time I've been talking about SpinRite, believe it or not, is that there's absolutely no need to run it in that mode all at once.

One of the things that I added in SpinRite 6 is a percentage complete that is accurate to four decimal places. And I did that specifically so that it would be, no matter how big the drive, it would be sufficiently sensitive to represent where it was. So, for example, if you have a drive that's so large that SpinRite's going to take a long time to run, and it's just inconvenient, it's completely feasible to start it at the beginning at the end of the day, let it run overnight or until you need it. Then you just hit Escape, and up pops a screen saying, like, "Pausing SpinRite," and showing you, it'll say SpinRite is 37.39246 percent in. You make a note of that and terminate SpinRite and then use your computer for the day. And then you are completely able to pick up exactly where you left off.

And that's one of the options when you're starting is by default it starts at the beginning, but you can also enter the exact percentage where you want to begin. So you'd put right back in 37 point whatever I said, and whatever you wrote down. And you know me, I'm a perfectionist, so I always - I round forward when I'm reporting where we are, and I round backward when you're restarting, so that you're always guaranteed of an overlap between where you ended and where you restart, no matter what happens with the math. So you're absolutely guaranteed, then, to start from where you were and be moving forward. And you could run SpinRite over the course of as many individual sessions as you want it to in order to get the whole job done. And doing that gives you full SpinRite preventative maintenance protection.

So in all this time I had never mentioned the fact that it doesn't have to be all done at once because there is a very nice suspend and resume capability built into it.

Leo: If you don't mind, let's take a break for one last commercial before we get into memory and micro kernels.

Steve: Yes.

Leo: Some topics that we brought up Sunday on TWiT.

Steve: That's what triggered both of them.

Leo: Steve wants to weigh in. You know you could always just call. I can put you on any time. But it's nice. Do it on your show. Yeah, it's nice. All right, Steve.

Steve: Okay. So there is a new memory technology on the horizon.

Leo: Yes.

Steve: And it is a true breakthrough.

Leo: Allyn Malventano had just come back from the FMS, the Flash Memory Summit, I guess it was, and was talking about this.

Steve: Right.

Leo: Yeah.

Steve: And this is not flash memory. This is something different. This is the result of years of work from a joint venture between Intel and Micron, working on this technology together.

Leo: Both known for their solid-state drives, by the way, and flash memory.

Steve: Yes.

Leo: Yeah.

Steve: Yes. This blows flash away. I mean, completely. It is one thousand times faster than flash memory, than so-called NAND, as it's called. It's got a thousand times more endurance, so it doesn't have that - we've talked about the fatigue problem with NAND often because essentially we're stranding electrons on a little island floating with an insulator underneath it. And the way you write is you create enough of a electrostatic field that you overcome the insulation and drive the electrons, you tunnel the electrons through the insulator.

The problem is that fatigues the actual physical properties of the insulator. That's why writing is hard on flash memory. That's why writing causes damage - tiny, incremental, but still something, for example, that hard drives don't suffer from. This, no sign of that kind of an endurance problem. And it is 10 times the density of dynamic RAM, of DRAM. And I have a picture here in the show notes that's called 3D XPoint Technology. And it's almost impossible to imagine a more simple solution.

In static RAM, static ram is so-called SRAM. That's like the registers in a processor. And essentially static RAM are cross-coupled inverters, inverters that are coupled to each other. And if you think about it, we've talked about this in the past. If you pull an input low, the output of that inverter will be high. And that goes into the input of the second inverter. That'll mean its output is low. So that continues to put a low into the input of the first one. So two inverters connected to each other are stable. And if you yank one of those lines in the other direction briefly, it'll flip. And that's why it's called a flip-flop.

And the problem is that requires a bunch of transistors. And transistors take up space. And they also take up energy. And they produce heat. So, and by "a bunch," it's like six or so transistors. The advantage is it's super fast. The problems are it's volatile, that is, it's only the fact that you keep those inverters powered up that has them knowing what way they were last set, that is, their memory is a function of this dynamic process. So

that's why static RAM is volatile. Dynamic RAM was a major innovation where all of that complexity was reduced to one transistor and a capacitor. And there the concept is you store the state in the charge of the capacitor. And the downside is the capacitors have to be itty-bitty in order to get a lot of them in a small space.

Well, itty-bitty capacitors tend to leak and can't store much charge. That means you need to read them periodically, before they've had a chance to discharge, in order to top them up again. And that's what refreshing is. Dynamic RAM, DRAM, needs to be constantly refreshed, meaning that there's a highest priority of all down in the hardware which is going through the entire DRAM of the system, reading it and then rewriting, basically checking the fill state of every one of the little capacitors, and topping it off before it has a chance to discharge down into the - enough that you can't tell whether it was a one or a zero. So, again, one transistor, one capacitor, much smaller because it's so few components. But with it comes the obligation of needing to refresh it. And of course it's also volatile.

Now we come to this so-called 3D XPoint technology. It uses essentially a phase-change technology. HP and SanDisk partnered up on something called memristors years ago. And it looks like that effort is failing. It doesn't look like that's going to happen. This one is similar. It is happening. Here, imagine a horizontal set of conductors, and you put little dots of stuff on these conductors horizontally. And then you lay on top of it a vertical grid of conductors such that you've got the horizontal array of conductors intersecting with the vertical array of conductors in XPoint, thus the name.

And at every intersection, what separates those is this physical substance. And what they have found a way to do is to run a current through this special substance which changes its resistance property permanently. So you send a pulse of current through it in one direction, and its resistance drops. You send a pulse of current in the other direction, and its resistance rises. And it's known as a bulk change, that is, the whole thing changes property. So it is very stable. So it is nonvolatile. And, as you can imagine, all it is is a grid, an overlapping grid of conductors.

And you can see why it's called 3D, because I just showed one layer, but then you put dots on top of that one and put a grid in the other, perpendicular to the grid, and dots on top of that, and another set of connectors perpendicular to it, and you can stack these things up so your efficiency goes up very high. What you end up with is - and it's just, it's hard to get your mind - it's hard for us to get our mind around this and what it means because it's such a change from the way we're used to thinking.

We have always been thinking in terms of bulk storage, mass storage being blocks, or being sequential. Or being slow access. The idea being that the registers in the processor are static RAM, superfast. The L1, L2, and L3 caches are superfast memory. Then we go outside the motherboard to DRAM, which is much larger, but much slower. Which is why we have a hierarchy of increasingly fast, but also smaller caches. And then connected to this whole system is comparatively much slower memory. Fast as it is, hard drives or SSDs, it's still dramatically slower. And we're used to thinking in terms of fetching a block. You know, go ask for this sector, or read this block of sector into memory, transfer it into memory, and only then can we use it.

What's mind-blowing about this technology is that it is random access, high density, and nonvolatile. I mean, it's like core. It's like the return of core memory, where we started in the early days, where you could turn the computer off - I did this when I was 14. You turned the computer off, and then you come back in the morning, you turn it on, and it's still there. I mean, it doesn't have to boot. It doesn't have to do anything. It's just - because the memory itself was nonvolatile.

So their first chip, and they're thinking later this year, I mean, it's working. I have some links in the show notes for anyone who wants to see. There are a couple of very nice YouTube videos and a lengthy 45-minute Intel, joint Intel-Micron presentation that I watched a couple years ago, whenever it was that it happened, because I was fascinated by this. And this is actually - this is not blue sky. This is not, oh, yeah, like supercapacitors, let's hope someday that they figure out how to do it. This year, they're saying 2016, and we don't have a lot of months left in 2016, we're going to get this.

Their first delivered product is a 128Gb die which is 16GB on a chip. So it's 16GB of mind-blowing, random access, nonvolatile memory. I mean, and again, we're so used to the concept of essentially storage being sort of semi-offline. I mean, you just can't randomly address it. Now for the first time I'm glad for 64 bits. Until now it's like, eh, who needs 64 bits; 32 is fine. Except it's a little bit like the Internet, okay, 32 turned out not to be enough bits for the Internet. With 64 bits, imagine that you just - all of the mass storage, all of the memory in the system is just there. And you can access any byte of it that you want to like it was DRAM, except 10 times faster than DRAM and nonvolatile. So also it is your storage. It's just, it's weird to, like, to think...

Leo: Wait a minute. It's faster than DRAM? I thought it was faster than NAND but slower than DRAM.

Steve: I'm sorry, 10 times more dense than DRAM, you're right, you're right.

Leo: And faster than NAND memory.

Steve: Right.

Leo: Like a thousand times faster than NAND. But it's not quite as fast as DRAM.

Steve: Correct.

Leo: Yeah. But, I mean, Allyn hypothesized machines that didn't even have DRAM, that this would, you know, you could live with slower DRAM if you didn't have to transfer stuff from storage into RAM.

Steve: Correct. It changes, I mean, what's so weird is that we are just so used to thinking about there being a delay and needing to, like, go access something in a block. Here it's like 21st-century core. It's just you can access any of it that you want to. And it's fast and nonvolatile and just goes.

Leo: So you imagine machines that, you know, you no longer say, "I have 8GB of RAM." You just say, "I have 20TB of..."

Steve: Of storage.

Leo: "...of XPoint storage." And it's all available, and at near RAM speeds. Wow.

Steve: Even an iPad, you know, there we see a single device where we know it's got flash memory and RAM. But that division dissolves in this case. So it's just, to me, it's just so cool. Over time we've developed abstractions for the way we access memory. Back, I mean, a long time ago, I mean, memory was a challenge in the early days of computing. There were actually data and programs stored acoustically in a mercury delay line because it took time for acoustic waves to propagate through a tube full of mercury. So there were transducers at each end and an amplifier, and this thing would recirculate. And the pattern of acoustic waves moving through this column of mercury was the storage. I mean, that's how clever people were becoming.

And then of course there was drum memory, where you had a spinning drum, and it was kind of random access. But it turns out that there was an art to writing programs that were inherently synchronized with the rotation of the drum so that, when the program needed to fetch its next instruction, that had been placed on the drum the proper distance upstream for how long the previous instruction took to execute so that it would be there, available to be read in. I mean, you think about the pioneers of computing and what they went through with the tools they had at the time.

And then of course all the sci-fi movies showing the spinning mag tape, where magnetic tape was the way you did bulk storage. Think about the challenge of sorting files on mag type. An incredible thousands of hours of programmer time went into designing algorithms to, efficiently as possible, sort and extract data from multiple mag tapes. You'd have scratch tapes and sort of mount them on machines and then run a deck of cards through the card reader in order to load that up into core. And then that would run a program that would do your data processing, if you were a big insurance company, where all of your customer records were on mag tape, where you can barely get to them. But somehow they made this work.

And so of course we've moved slowly forward with storage getting much faster and much less expensive, but always fundamentally remaining kind of inaccessible, block accessible, where you would address a block. And this represents such a big change in thinking that it's just sort of hard to get your mind around. So I just, to Intel, I say, and Micron, bravo. I don't think we can guess yet where this is going to go. I imagine it'll be very expensive in the beginning. I don't think it's going to threaten hard drives just in terms of cost per bit.

But again, if we've seen anything, it's that these are exponential curves. And I think there's no doubt that a decade from now this will have replaced SSDs easily, well, probably sooner than that, I think, given the pressure for, especially in enterprise and big cloud storage facilities, to access data more quickly. It's just fabulous. And I'm done.

Leo: You don't want to do micro kernels?

Steve: Let's talk about it next week.

Leo: Yeah. Okay, good, yeah. We're out of time, so that's fair enough. I'm excited about this memory thing. This is the kind of breakthrough technology that doesn't

happen every [crosstalk].

Steve: And it's just, it's elegant. It's just elegant. I mean, you look at it, it's like, okay. How do we simplify this? You can't. It's just, you know, it's intersecting addressing lines in a little blob of goo that somehow remembers...

Leo: They're resistors.

Steve: ...the last pulse it received.

Leo: They're resistors; right?

Steve: Yeah.

Leo: That's what's kind of interesting to me, yeah. Fascinating.

Steve: Yeah. They're resistors whose resistance can be changed.

Leo: Variable, yeah.

Steve: Yes, based on the history of the current that flowed through it.

Leo: It's awesome.

Steve: It's just so cool. It's just, it's like, you know, that's why I sort of painted the picture of static RAM with complexity. Dynamic RAM, way simpler. And now this, this is it. This is just cut to the bone.

Leo: I guess you'd still have some sort of high-speed cache RAM lying around.

Steve: Yes, yes.

Leo: To keep the pipeline full.

Steve: I think you still need that. Even with DRAM, although DRAM has its limitations because it wants to be read out. I mean, this, well, I was going to say, the organization, the physical organization of DRAM requires some constraints. The reason, you know, DRAM has a natural size at which it can be read, and it's called a cache line. And so a line is read into the Level 3 cache. And then the DRAM typically needs to get refreshed.

Because in order to read it, you need to transfer the charge of that line into a buffer. And once you've done that, you've discharged that line, so you need to rewrite that line.

So there is overhead associated with DRAM that we just don't have with this technology. This is just, I mean, the only thing I can imagine is some other sort of like holographic crystal, where we're able to zap it with lasers to, like, flip molecules around inside. That's, again, and probably a slower access mass storage. I think this is going to be sitting here as the king of high-speed, nonvolatile storage for quite a while.

Leo: Well, you told us about ZIP disks. You told us about flash memory. Now you're telling us about Optane. Optane.

Steve: Nice.

Leo: Steve Gibson. This is the show to listen to if you want to keep up on what is going on for real in the world because this is our most sophisticated show, I think. You've got to tune in every Tuesday, 1:30 Pacific, 4:30 Eastern, 20:30 UTC. Or Wednesday on demand, after the fact. We'll get the show out in a couple of hours. Steve also a little later will get written transcripts of the show. He's going to have an audio version and the transcripts at his site, GRC.com slash, well, whatever. Just go to GRC.com. It's in the menu. He'll also have show notes there. He always does, so if you want links and so forth.

We have the audio and the video at our site, TWiT.tv/sn. And you can also subscribe, and that way you won't miss an episode, you know, whatever podcatcher you prefer. While you're at GRC.com, though, consider making a yabba dabba doo sound in Steve's lair by buying a copy of SpinRite, world's best hard drive maintenance and recovery utility.

Steve: And you don't have to run it all at once. You can run it over the course of as many little sessions as you want.

Leo: Once in a while. A little bit at a time.

Steve: Yup.

Leo: You can also find out more about SQRL, the Healthy Sleep Formula, and all of that. It's all at GRC.com. Steve, we'll see you next week. This is our last episode in this studio, by the way.

Steve: Yay. And we're going to see what you come up with for your studio next week.

Leo: Well, eventually it's going to look exactly the same.

Steve: But probably not by next week.

Leo: Not by next week. Next week we'll be sitting at the roundtable, in all likelihood. But the week following, I'm guessing it'll only take a couple of weeks to get this recreated.

Steve: I think it'll be good.

Leo: And you won't notice the difference except I won't be alternatively freezing and sweltering.

Steve: Are you guys going to take photos of your current setup so everything can be placed in the same location?

Leo: Oh, that's a good question. Where does everything go? You know, it's a shame we don't have any video of what it looks like or...

Steve: Yeah, no record.

Leo: No record of how it's changed over the years. We tried when we built this to make it look like The Cottage. And I think we came pretty close. But this time we're taking everything with us. There's nothing to build. We're going to keep all of this, including my desk and everything. So it shouldn't change at all.

Steve: Cool.

Leo: I think we'll have a better monitor for you. Don't have to adjust it every time. Thanks, Steve. We'll see you next week in the new studio, in the Eastside Studio, for Security Now!.

Steve: Thanks, Leo.