SECURITY NOW!

Transcript of Episode #539

Listener Feedback #226

Description: Leo and I discuss the week's major security events and discuss questions and comments from listeners of previous episodes. We tie up loose ends, explore a wide range of topics that are too small to fill their own episode, clarify any confusion from previous installments, and present real world application notes for any of the security technologies and issues we have previously discussed.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-539.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-539-lq.mp3

SHOW TEASE: It's time for Security Now!, the last episode of 2015. And you think, you know, it's Christmas time. The hackers must be relaxing. No, no. There's all sorts of news from the front, including a horrific story about security on Juniper routers. Steve will talk about hotel and other public WiFi access points and the risks therein. And we'll answer some great questions from our audience. Security Now!, the last episode of 2015, is next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 539, recorded Tuesday, December 22nd, 2015: Your questions, Steve's answers, #226.

It's time for Security Now!, the show that protects your security and privacy, with this guy right here, Steven Gibson. He is the man in charge of Security Now! and GRC.com, the Gibson Research Corporation, where he purveys the finest hard drive maintenance utility known to man, SpinRite, and lots of other free stuff. Hello, Steve. Happy holidays.

Steve Gibson: Yo, Leo, the last podcast of the season.

Leo: Yes.

Steve: So I get a week off. You're going to go off on vacation, which is cool.

Leo: Yeah, so exciting.

Steve: And we'll come back all bright-eyed and bushytailed and do our next - the first one of 2016 on January 5th. This week, lots of news. We're going to reprise our attempt at a Q&A.

Leo: This time we'll do more than one, I think.

Steve: I expect we'll get to more than one. But there are some juicy deep things for us to cover. Also, of course, the Juniper router catastrophe. Really, really interesting story there. And I want to talk about Oracle getting smacked by the U.S. Federal Trade Commission. What happens if you press backspace 28 times in a row when you're trying to log into Linux? I wanted to talk about WhatsApp's run-in with Brazil. A little note on some misreporting of Hillary's call for a Manhattan-style effort to break encryption, as unfortunately Ars Technica misreported it. Some update on web privacy via an interesting automated audit. Microsoft has increased the controversy over Get Windows 10 yet again. We have some miscellany stuff. And then questions. And I didn't even try for 10. I put eight in because I figured, well, we'll just use them as filler with whatever time we have remaining after we deal with all of the main show content. And I did get your mail this morning.

Leo: I was going to ask. I didn't mean to add new content.

Steve: No, no, no, yours is so good I want you to kick off with that because I did not have this in the notes, and I heard you talking about it over on MacBreak Weekly, so I know you're up to speed on it.

Leo: Yeah. And I wanted to get your input on it because, A, I wanted to see if this actually made sense. It is a year-old article from a Dutch journal, and I am not going to attempt to pronounce the Dutch.

Steve: So for what it's worth, for our listeners, before you explain what it is, I listened to you explain it, and it is 100 percent accurate.

Leo: Okay.

Steve: Everything that you said makes sense. And it is a perfect snapshot into the vulnerability of any public, non-encrypted WiFi.

Leo: And then I also want to ask you about my solution. But let me show you this. What's nice is this was translated to English from the Dutch De Correspondent. And the author of it, here I go, I'm going to ruin this, is Maurits Martijn. But the title is probably good: "Maybe Better If You Don't Read This Story on Public WiFi."

"We took a hacker to a caf, and in 20 minutes he knew where everyone was born, what schools they attended, and the last five things they googled. And he did it with a hardware device." And I'm not sure what the hardware device is. They simply describe it as a small black device about the size of a pack of cigarettes with an antenna on it. And he did really kind of some clever stuff with it. He got the password of the WiFi network, had the hardware device join the WiFi network, then had his computer, probably using, I don't know what software, but some software on the computer, to then see all the devices on the open WiFi access port. But more than that, to also see which other WiFi networks the devices were previously connected to. I thought that was interesting.

Steve: Yup.

Leo: And then using that to do a man in the middle by spoofing those preconnected networks, their home networks or whatever. It just was fascinating. I presume it was a man-in-the-middle attack. It's hard to, you know, this is an article designed for kind of a general public, so it's not very deep in the technical details.

Steve: Right.

Leo: But essentially, as far as I can tell, and I guess you've read it, and so you can confirm this if it's true, it looked like he made a man in the middle using spoofed WiFi access points and then was able to see a whole lot of stuff, including people's previous searches, you know, and scanning for names, passwords, sexual orientation.

He found a woman who had just recently moved to Holland, and she was searching for health information, all sorts of stuff, where she was born, where she studied. She has an interest in yoga. She's bookmarked an online offer for an anti-snore mantra, recently visited Thailand and Laos, and shows a remarkable interest in sites that offer tips on how to save a relationship. So I read this, and I got scared enough that I said, we're going to be on a boat. Normally I just - I tether or hotspot to my phone, and it's LTE, instead of using an open access point. Usually that's faster. And that's safer; right?

Steve: Right, absolutely.

Leo: So I tether the phone, and I use that. But on the boat I'm not going to have a choice because we're at sea. We have to use the satellite Internet provided by the boat. That is a viciously open WiFi network because it's 5,000 other people on it. So I immediately went out, and I probably won't get it in time, and ordered a tiny hardware firewall. I've used these before. Hotspot VPN offers them. They're free if you subscribe for a year for 91 bucks. That seems like a good deal. And it's just one of those little Marvell or some other small processor-based device that will join the WiFi network, or an ethernet network.

Steve: Establish a VPN tunnel out to the Hotspot VPN.

Leo: And then create either a WiFi hotspot or another ethernet connection. But it also can run a Tor connection through that, as well, which seemed like a nice - this is newer hardware and so can do more than the old hardware I had. And so that seemed like a good idea. Yes?

Steve: Yeah, I think so. So the problem with WiFi is that it's like a hub in the traditional wired ethernet sense where the way a hub worked is that any data coming in to any port of the hub was rebroadcast out of all the other ports of the hub. So it was very easy to do so-called "promiscuous sniffing." Basically, any person who was just sucking in the traffic on their port of the hub was seeing all of the network's traffic.

Now, this is not as easily done when you change to an ethernet switch because it intelligently learns, on the fly, what MAC addresses of devices are connected to which ports. Even if it's a few switches downstream, it still learns, oh, down this wire is this collection of MAC addresses. So ultimately, only one device is on a given switch port, and that switch, the ethernet switch, has learned who to send that MAC address's traffic to. And so the beauty of that is that they are, while they're not super secure, they're way more secure than a system which simply broadcasts everything on the network to every other person on the hope that someone, somewhere, is the recipient. So moving from hubs to switches was a great step forward.

Well, in that sense, going to open WiFi is a great step backwards because we're back into radio, which is inherently promiscuous, so that everything anyone sends is in the air. And where there may be some segmentation, I assume on a sufficiently large yacht or boat or chip, that they'll have multiple...

Leo: They've got to have some VLANs or something, yeah.

Steve: Yeah, well, they'll probably have multiple access points, and probably have those connected with a switch. So you may not be able to see the traffic of somebody at the other end, you know, if you're on the bow, you may not be able to get the traffic from someone on the stern, but you would certainly - and this is certainly the case in a single access point coffee shop scenario - have access to all the traffic of everyone there. And what I liked about your stumbling over the story is that it takes us out of sort of this theoretical, oh, well, yeah, you know, ARP spoofing, maybe that can happen, or access point spoofing, maybe. And very much the way Firesheep, we remember, back in the days before Facebook, to name one prominent website, went to TLS HTTPS connections, where Firesheep was simply using promiscuous sniffing on open WiFi to collect all the unencrypted traffic from everyone there.

And what Firesheep was doing was grabbing the cookies which were being sent in the clear after the person had logged into their service. So somebody would, you know, back then, Facebook would bring up a secure connection only to protect the username and password. Then Facebook would respond to the browser's query with a cookie, setting a cookie in the browser, and then drop security, switch back to unencrypted connections, because it was believed, and there was some truth to it back then, that computers were not fast enough to do encryption in addition to everything else, and that it would slow things down, and especially at the server end because all of these encrypted connections were being concentrated into Facebook servers. They said, eh, we only want to do that, you know, only when we really need to.

So that was sort of old-school thinking, which increasingly has been abandoned because we now have hardware coprocessors that are able to handle the overhead of the crypto establishment. We have faster crypto, the elliptic curve crypto, which is the term we're going to be hearing later today about the whole Juniper mess. That's way faster than the traditional RSA-style handshake establishment. So we've had progress in crypto. We've had progress in machine speed. Power is going up; cost is coming down. Now it's more economically feasible to encrypt everything all the time. But even in that environment, it is still possible in a WiFi environment to send a user's machine a disassociate packet, which is the normal connection shutdown for a WiFi connection. And so if a disassociate packet is sent, that breaks the wireless connection.

And from reading between the lines of what is, what was published in that story, the guy was able to acquire from the various users' broadcasts of the SSIDs that they know of, a list of their SSIDs. Most people had their systems set to automatically connect to known, to previously known access points, like when you're back at home, you don't want to be harassed to need to do anything, so you just say, yeah, fine. So he was able to, with this little bit of hardware, to spoof the SSID of the beacon of a known access point, having obtained that from a client's machine that he had disassociated with the coffee shop system, and it connected without complaining. Now he had a direct connection to it and was able that way to establish himself as a man in the middle.

Now, it's still the case that, without being able to force a certificate on the user, he wouldn't be able to crack their encryption. But he may be able to do things, for example, like do an HTTPS or TLS downgrade, where there would be some way to get that user not to use encrypted connections, and then get nonencrypted traffic on the fly and start gathering data.

Leo: Yeah, because sniffing a certificate, there would be evidence of that.

Steve: Yes, yeah, exactly. The user would be notified that the site they thought they were connecting to was trying to use a certificate that was not known to their browser. And browsers used to let you push past that. Now it just says no.

Leo: Nope. Not going to do it.

Steve: They're just, sorry, go somewhere else. So anyway, this is the danger of current nonencrypted WiFi when there's really a malicious actor present. You can do the Firefox, the old Firesheep tool, it was a passive sniffer. It was just sniffing the traffic that was more or less nonencrypted at the time. But if you step up your game to an active attack, and this guy certainly was, but it's wireless so no one knows what he's got in his backpack, that he's able to do this.

So I agree with you completely, Leo. In an environment where you're going to be spending a lot of time, I mean, a ship like you're going to be on is very much like a hotel. A hotel, as we've talked about in the past, is another very high, you know, a target-rich environment, if the hotel was using hubs. Or now, of course, hotels are often using WiFi now. And so there once again they're creating a problem, if they're not an encrypted WiFi.

Leo: Yeah, and I feel a little bit bad because I have been saying to people, yeah, you know, it's a pain in the butt to use VPN. So what you should do on an open WiFi access point is, A, only do stuff you don't mind if somebody sees what you're doing; and, B, if you're doing something like using a credit card or banking, as long as it's HTTPS, a secure link, you're probably safe in that regard. And you want to make sure most importantly that your email password is secure, and your email transactions are secure.

Steve: Yeah, and you know, that's a very good point. It may well have been that email was the way in for this guy because email is still, if you're connecting to SMTP, more often that not, it's not encrypted. And once you see a username and password, then he's able to get in. And if you're using IMAP to pull however many, you know, like current repository of undeleted email is there, and rifle through it, and find out all kinds of things; and, as we've seen, even obtain password reset links which are incorrectly coded and aren't expired and allow them to reset the password for something that - because, I mean, people are typically forgetting their password all the time because they're still not using password management to the degree that they should. So we're in a mess. There's just no two ways about it. Right now the industry is in a mess.

Leo: Do you think, I mean, I'm also going to assume that I'm a target more than, you know, just some stranger on the street, that it seems like I should maybe pursue something more aggressive than just being careful about what I do online in public. So that's why I got the hardware device. I don't know if it's necessary for everybody to do that.

Steve: Well, and remember, too, that if you were to use a hotspot, I'm sorry, a VPN, on your various machines, then that solves the problem, too. So anybody who's in, you know, the non-hardware approach is simply use proXPN.

Leo: Right, right.

Steve: Install the client on your machine, and when you're away from a secure environment, simply bring up a VPN tunnel, and at a little cost of performance, because everything has to go through there and then disseminate, so there will be a little bit of tradeoff. But the tradeoff is absolute local security, so that your traffic is encrypted out of the frightening environment. Now, of course the problem is, to be really secure in your environment, you've got to have, like, what, your camera and five different phones and a couple of MacBooks and some iPads and things. And so the beauty of that...

Leo: This device becomes a WiFi access point...

Steve: Yes.

Leo: ...because it allows four devices to connect. So I'm going to tell Lisa and Michael, use this as your WiFi access point, not the ship.

Steve: Right, right.

Leo: Probably won't do anything.

Steve: In fact, my best friend travels a lot and was worried about his security when he was recently in a hotel. And I can't remember now the device I found for him. But he said, you know, what do I need? And it might have been a TP-LINK. I don't remember. But it was only, like, 20 bucks or so.

Leo: Yeah, these are made by TP-LINK, some of them, yeah.

Steve: Ah, good. And it specifically supported the comment you made on MacBreak Weekly, and that is the problem of a captive portal. Because with some of these, you somehow need to say yes, I agree to your terms of service. And that needs to be done before everybody else connects to it, without needing to independently authenticate themselves. And this device specifically knew how to do that. I will find the note and talk about it in two weeks because I know that our listeners will be saying, oh, well, what was it, what was it? And I don't have that in front of me. Maybe I can find it.

Leo: And I'm sure everybody's going to say, what was the device the bad guy was using? And I think many of our listeners can figure it out. Some in the chatroom already have. I'm not sure we want to really talk about how.

Steve: No. No, we don't want to enable people to do that. The good news is it's not simple. At this point it's not a turnkey thing you can get. It was just - and these are available. It was a wireless WiFi transmitter that gave the computer low-level promiscuous access to all of the radio. Most receivers in laptops won't do that. But you can buy some that are little USB dongles. I have one because I've been curious in the past, and I've poked around, sucking in all of the traffic at Starbucks. And it was an eye-opening experience back in the day. Not so much anymore because there is more security, and you need to go active in order to do this. But there are, in the gray areas of the Internet, all of this stuff is open source and free and available for someone who wants to pursue that.

Leo: But it's kind of a multilayer, multistage thing that uses some skill, it looks like.

Steve: Oh, you've got to know what you're doing, yes.

Leo: Got to know what you're doing, yeah.

Steve: Yes, yes.

Leo: Thank goodness.

Steve: So our Picture of the Week on the first page of the show notes, we'll loop back around to this when we talk about this interesting audit that was made of web privacy. But the numbers just make you just shake your head. This is, of the top 1,000, so it's 1,000 most popular websites. This is a histogram of the number of cookies which were given to a spidering test browser that was written in Python, so they created sort of a pseudo browser in order to crawl the top 1,000 sites, and for it to then act sort of as a database also to record and count the number of different assets that it received. So, for example, almost half of the sites, half of the top 1,000 only gave up to 49 cookies. Zero to 49 cookies were offered by 473 out of the top 1,000 sites. But if we slide down the curve, 47, for example, sites gave between 200 and 249 cookies.

Leo: What? What?

Steve: And there were 26 sites, not of those 47, but another 26 sites that were over 350 cookies.

Leo: Wow.

Steve: So, I mean, so this shows us how out of control this has gotten. I mean, yeah, 26 sites out of the top 1,000. That's not a lot. But 350-plus cookies from those 26 sites among the top 1,000? That's insane. This is the feeling that I've been trying to articulate all year is it's just, it really - and this is what happened, I mean, this is why Apple finally said, okay, fine, we're going to put some filtering into, you know, we hear our listeners. We hear our users. We're going to put some filtering into Safari because of what has happened to the mobile web. You try to surf these sites on a bandwidth-constrained or bandwidth-metered mobile device, and one of 47 of the top 1,000 sites wants to give you up to 250 cookies, it's like, ow. That's going to slow things down.

So anyway, it's a great audit. I'm not going to go through it in great detail even when we get to it later in the show. But I do have a link for people who are interested because it's just eye-opening. Again, the problem has been, as I have said before, one of perverse incentives. There was zero cost, virtually zero cost, and some minimal incremental benefit for sites to put more crap on their pages. And users bore the brunt, and the sites got little snippets of revenue from all these various parties that said we'll pay you, we'll make a micropayment to you if you'll put this little bit of JavaScript on your page and provide you with who knows what, instrumentation about what's going on.

And so they said, okay, yeah, fine, you know, we're trying to support ourselves from traffic to our site. And this is what happens because there wasn't any pushback. It was increasingly slowing things down. But sites were making a little bit more money from each additional one of these things that they said yes to. So the good news is we're really beginning to look at it now, so there's some pushback.

Okay. So our network-aware listeners will know the name Juniper. Juniper is right up there with Cisco as providing the major Internet infrastructure backbone. When I go into my datacenter where GRC's servers are, it's racks of, like, just crazy optical fibered Cisco and Juniper gear. This is the so-called "big iron" routers and switches that route the traffic of the Internet. So that's, for people who aren't familiar, who aren't as familiar with the name Juniper, it's because they're not a consumer device. And really Cisco wasn't until they bought Linksys, that much. Cisco also was more of the old-school Internet plumbing sort of equipment.

Okay. So very quietly last week Juniper published or posted the news of an out-of-cycle important security update. They just sort of said, um, this is important. We would like everybody to update their equipment, their Juniper equipment, as soon as possible. And there's two vulnerabilities that this update fixes, thank you very much. And that's all they said. No details. Just sort of, here, we changed something. Fix it.

Leo: Always makes me suspicious when that happens.

Steve: Oh. So that's exactly what happened was that a number of the names we talk about, Adam Langley was involved, Matthew Green, HD Moore, it's like everyone's like, wha-what? Because, I mean, this is not like a camera sold by Ali Baba like for $4 has an update. This is Juniper. And this is the so-called ScreenOS, which is like their - it's like, actually I was going to say it is actually called IOS, Cisco's, it's the Internet Operating System, which is completely separate from Apple's iOS. So this is their ScreenOS. So this is big news. But they didn't say anything.

So all that we were left, we the security community, left to do was the standard reverse-engineer the patch, which is get the firmware from the previous release and the firmware in this release and do what's called a BinDiff, a binary difference, of the two. That'll give us some idea what bytes are different. Then you use something called IDA, I-D-A, the Interactive DisAssembler. You have to tell it what the processor is. And this thing takes the machine language and disassembles it into the equivalent assembly code.

Now, it needs to be interactive, though, because what happens in firmware is everything is mixed together in a pile. That is to say, the data and the instructions are comingled. When you see the source code, they're typically separate, or at least they're clear. But when it's been compiled down, it's just a blob of hex. So you need to go through, and this IDA tool is - it's evolved over time. It's very good about helping you. So it's able to build essentially a flow graph with blocks of code that jump to each other, showing calls to other blocks. Typically things are not labeled because the symbols which normally are left in to help with debugging, those have been stripped out.

But there are still interesting little tidbits. For example, it turns out that the ScreenOS uses OpenSSL for its security. And OpenSSL has macros which expand to common sequences of bytes which previous people who have reverse-engineered compiled code using OpenSSL have already figured out, oh, look, it's left little signposts all over the place, just as sort of a side effect of the way the code was put together. So people who are really good at this know how to take an updated firmware and dig in.

So what they found is phenomenal. And so, okay. So there were two different things found, both of them backdoors. I forgot to say that Juniper did say something else that certainly raised people's curiosity. Juniper said we have found unauthorized code, that is, unauthorized source code, in our master source. Meaning that what Juniper confessed was somebody, without their authorization, somehow got into Juniper's master source code repository and made some changes.

So this all happened in '09. So this is six years ago. And some sort of a review that Juniper did, they didn't say how this came to light. But so that other key phrase, that unauthorized changes were made to their source, that was just too good for the security community not to pick up on. So I want to share the beginning of Matthew Green's blog because he puts it just beautifully and, like, frames this right. And I'm only going to read the first, like, third of it. But it almost reads like a crypto sci-fi novel.

So Matthew Green, of course Johns Hopkins cryptographer, writes: "You might have heard that, a few days ago, Juniper Systems announced the discovery of 'unauthorized code' in the ScreenOS software that underlies the NetScreen line of devices. As a result of this discovery, the company announced a pair of separate vulnerabilities." And then he enumerates them, and he says, "and urged their customers to patch immediately. The first of these CVEs (#7755) was an authentication vulnerability, caused by a malicious hardcoded password in SSH and Telnet."

Okay, now, normally that would be a deal - that would just, like, that would stop the wheels. Wait a minute. Someone added a malicious hard-coded password into the secure shell and Telnet of our equipment. That would be a big deal. Anyway, no, that was, I mean, it is a big deal, but it wasn't sufficiently juicy.

Okay. So Matthew continues: "Rapid7 has an excellent write-up of the issue." And I have a link in the show notes for anyone who's interested. Matthew continues: "This is a pretty fantastic vulnerability. If you measure by the impact on security of NetScreen users. But on the technological awesomeness scale it only rates only about a two out of ten, maybe a step above 'hit the guy with a wrench.' The second vulnerability is a whole different animal."

Leo: Uh-oh.

Steve: "The advisory notes that CVE-7756, which is independent of the first issue, 'may allow a knowledgeable attacker who can monitor VPN traffic to decrypt that traffic.'" Matthew continues: "This is the kind of vulnerability that makes applied cryptographers cry tears of joy. It certainly did that for me." And the he quotes in his blog posting his December 18th tweet from four days ago, where he said: "I'm really invested in the idea that this Juniper encryption vulnerability is going to be amazing. Like, Flame-level amazing."

Leo: Oh, boy.

Steve: That's what Matt tweeted four days ago. And this was, like, back then we didn't know anything. Now we know everything. So he continues: "And while every reasonable person knows you can't just drop 'passive decryption vulnerability' and expect the world to go on with its business" - so let me just make sure we understand where we are at this point. What Juniper said was that a knowledgeable attacker could passively monitor Juniper's VPN traffic and decrypt it. So that's, well, first of all, how is the big question because this is not man in the middle. This is not spoofing anything. This is, well, we know, for example, the NSA has taps, and they're sucking all of the traffic of the Internet in. And so if there were some incredibly obscure flaw that would allow them to passively decrypt it, that's big news.

So he says: "While every reasonable person knows you can't just drop 'passive decryption vulnerability' and expect the world to go on with its business, this is exactly what Juniper tried to do. Since they weren't talking about it, it fell to software experts to try to work out what was happening by looking carefully at firmware released by the company.

"Now," says Matthew, "I want to be clear that I was not one of those software experts.

IDA (Interactive DisAssembler) scares the crap out of me," he wrote. "But I'm fortunate to know some of the right kind of people, like Steve Checkoway, who I was able to get on the job, mostly by encouraging him to neglect all of his other professional obligations. I also follow some talented folks on Twitter, like HD Moore" - of course HD Moore is the original father of Metasploit, that whole Metasploit framework - "and Ralf-Philipp Weinmann. So I was fortunate enough to watch them work, and occasionally, I think, chip in a helpful observation. And, yes, it was worth it because what Ralf and Steve and the rest found is beyond belief. Ralf's excellent post provides all of the technical details, and you should honestly just stop reading now and go read that." And I have a link in the show notes.

"But since you're still here, the TL;DR is this: For the past several years, it appears that Juniper NetScreen devices have incorporated a potentially backdoored random number generator, based on the NSA's Dual_EC_DRBG algorithm." Which we've talked about in the past, but I'll remind our listeners what that is in a second. "At some point in 2012" - so there was a reference to '09 and now here 2012, so it may have only been - only - the past three years - "the NetScreen code was further" - oh, okay, right, there were two changes. There was the '09 problem and then 2012.

"The NetScreen code was further subverted by some unknown party, so that the very same backdoor could be used to eavesdrop on NetScreen connections. While this alteration was not authorized by Juniper, it's important to note that the attacker made no major code changes to the encryption mechanism. They only changed parameters. This means that the systems were potentially vulnerable to other parties, even beforehand. Worse, the nature of this vulnerability is particularly insidious and generally messed up."

Okay. So enough of what Matt wrote. So here's the deal. We talked about before, previously, and for several weeks running, the revelation about the so-called "Dual_EC_DRBG." This was one of four, I think it's four, might have been three, maybe four, official NIST, formally approved, National Institute of Standards and Technology. These are the algorithms you should use if you want random numbers. And the listeners who've been following the podcast for a long time will remember the controversy, and it must have been, was it post-Snowden? It may have been Snowden-related revelations that the company, RSA, received a chunk of money, I want to say a million dollars, it's been a while, I don't remember the dollar amount, from the NSA to weight - this was never proven, but the allegations were that they received this money a long time ago to make this the default for the RSA sort of industry-standard crypto toolkit that RSA sells. And I have one.

Leo: You have it on the shelf behind you?

Steve: Yeah, I do.

Leo: Wow.

Steve: I can't - BSAFE, it was called BSAFE. And I bought it from them years ago, and actually after this was changed. So mine was safe. But I wouldn't have used the defaults anyway. You know me, I would go through and pick exactly what I wanted. So this algorithm was the default. In '07, some security researchers discovered a vulnerability, a fundamental vulnerability in the algorithm. The algorithm has two constants. It uses two constants, P and Q. And they are supposed to be chosen at random. And those represent some points on the elliptic curve such that, when you use those, this DRBG, the Digital Random Bit Generator, is a pseudorandom number generator.

So we know what it means. It means that from a given starting point it then follows an externally unpredictable path, as long as you don't know these secrets, as long as you don't know what those constants are or the internal state of the algorithm, that is, its starting condition, the idea being that you're not supposed to be able to reverse engineer its state from what it puts out.

What they discovered is that, if P was related to Q by a secret constant, that is, if they weren't independently chosen at random, but if they were deliberately chosen so that there was a - technically it's a finite field multiplicative relationship between the two, where you multiply a secret number E by P, and then do modulus the field size, and that becomes Q. If that's the relationship, that is, if they're not random, then what they figured out was, taking a very small bit of the output, an attacker who knew there was a special relationship between P and Q could very quickly, with a tiny bit, 30 bytes is the number that has been recently quoted, they could figure out the state of the algorithm and then predict all future random output. So that was known about DRBG.

What Juniper did was just change Q. And so when the reverse-engineer guys in the last four days saw that Q was changed of the DRBG, they said, what? Wait a minute. So what had happened was somebody got in, in 2012, and changed that one 32-byte constant to something else. And the only way this makes any sense is if it was somebody who understood this vulnerability, who knew that Juniper was using this now, I mean, completely deprecated algorithm. No one would be using this.

And in fact, back when this came to light, Juniper did acknowledge their use of this, but said that it was being used with their own arbitrarily arrived at constants, and not in any place where it could cause a problem. So it was sort of a nonissue. Still, it's like, okay, why is this still in there? But as we know, the industry tends not to change things that it doesn't really have to. So they left it alone.

So then, looking at this, what they realized was that this pseudorandom number generator was - its output was not being used directly, but rather it was being used to seed a second high-quality pseudorandom number generator, another standard in the industry, which happens to use 3DES as its pseudorandom mixing function and uses three different 3DES functions with some XORing and outputs fed around in circles and things. And it does produce, it's well known to produce, to be a very good source of pseudorandom numbers.

And so for a day the industry, the security researchers were puzzled because the exploit, for this to be an exploit, you have to know, you have to be able to determine the state of the DRBG pseudorandom number generator, which is then being used to seed the actual pseudorandom number generator, which generates the data which is available, like it's the raw crypto keying material that would be sniffable over the wire, or derivatives of it would be. So they couldn't figure out, okay, how that's a problem.

Then another researcher found a very convenient mistake in the code where a pointer to a buffer is in a "for loop." When the code is first loaded, the buffer pointer is set to zero. The for loop increments it by eight, checking to make sure that it is less than or equal to 31. So it runs that loop four times, moving this buffer pointer forward by eight each time, to essentially copy 32 bytes from one place to another. The mistake is it is never - oh, and I should say, and that data is coming from, that is the output of the DRBG into the reseeding of this next pseudorandom number generator. The mistake is it is never again reset to zero. So all reseeding of this fails. And the result of the second pseudorandom number generator isn't used. Instead, the direct output from the DRBG is used.

So essentially what happens is there was a bug in the code such that they were never actually, except after initially, when the system was first started up, no reseeding of the secondary PRNG, pseudorandom number generator, was ever done. So that the DRBG data was - it was pseudorandom, but now we know that, if an attacker chose the P and Q constants and had access to just a little bit of its direct output, not obscured by the intermediate PRNG, they would have decryption. And that's what was found, a bug in the code that allowed this bad and completely deprecated DRBG algorithm to have its data exposed in VPN and presumably other connections, all of the various entropy that the router was using such that somebody who knew the change, knew of the bug, knew that this was there and what was happening, would be able to do passive, just sniffing, decryption of all the VPN traffic. I mean, it's like crypto sci-fi. I mean, it's just like, you know, we never see this. And so...

Leo: And how long has this been going on, until now? They patched it now; right?

Steve: Patched it now, for three years.

Leo: Three years.

Steve: This constant was changed in 2012. Somebody...

Leo: So nobody has known about this for three years. But if you're using Juniper equipment, people could have been watching you for three years.

Steve: Yes.

Leo: Wow.

Steve: Yes. And we don't know who. We don't know if it was a foreign state actor. We don't know if it was our own intelligence services. There's no record of it. No one knows.

Leo: Hence the issue...

Steve: As far was we know, Juniper doesn't know.

Leo: ...as Rene pointed out, of having a backdoor in something is a backdoor can be used by anybody who discovers it.

Steve: Yes. And in fact Matthew, toward the end of his blog post, and I have a link also in the show notes for anyone who wants to read the whole thing, notes exactly that. He says that he's been running around all over Washington, trying to explain to politicians exactly this kind of thing, why, if there's a backdoor, there is just no way to keep it secret. There is no way to prevent it from the possibility of abuse. And so the nice thing is that this just fell in our laps as an early Christmas present for all of us who want to make that point.

Leo: The counterexample.

Steve: And the question is, who made the change? Somebody made it, and there's just, you know, it was a little annoying back when we were first discussing this Dual_EC_DRBG because the evidence was so circumstantial. It was like, oh, and remember, I remember somebody saying, oh, well, a million dollars isn't much to RSA. Oh, but RSA of that day, that was one quarter's worth of their revenue back then. So that was in the early days, when they were much smaller. And they never changed the default after presumably, or allegedly, receiving a payment from someone, we believe the NSA, to make it the default. And so this thing still lives. If these people used OpenSSL, I wouldn't be at all surprised if they also used the BSAFE package from RSA, and built it in, and left with the defaults.

Leo: And it might not be malicious. Yeah, it might have just been, you know, they used the default. Well, you can trust RSA; right?

Steve: Well, except that the change of the parameter, that's the difference. And thank you, because I hadn't finished my thought. It was annoying back then because all of this was circumstantial. It's like, if P and Q - oh, and I forgot to say that it is known that those constants came, the original DRBG constants from RSA did come from the NSA because the NSA, back again 20 years ago, we believed they had our best interests at heart. The DES, the Data Encryption Standard, that's an NSA cipher. And we believed that the NSA had, and maybe at some point it was true, the best cryptographers in the industry, I mean, in the world. And they may still today.

So there was no documentation of where that constant came from. For example, a perfect solution would be to take a passage from the Bible and hash it, and that be the P constant, and a different passage and hash it, and that be the Q constant. Because what that would enforce is that P and Q were not deliberately chosen. If known texts were hashed, we know that the action of a hash is to produce an unpredictable output, that is, no one has any control over it. That would prove no one chose them. But we have no such proof in the original DRBG constants. They were simply, you know, from on high they were given to the NIST standard, without comment or explanation of where they came from by the NSA.

So that's why, when the researchers then discovered that, if they were deliberately chosen, if there was a special knowledge of their relationship, that would allow this pseudorandom number generator to be cracked. And again, same rule applies here. If it were possible for only the NSA ever to possibly know that, then, okay, there's a backdoor that only they could have access to. Except that we've just demonstrated it's math. It's not possible for these secrets to be kept. And so the point was that there we never had any formal proof of exploitation. All we had was strong suspicion.

Now, what's different today, is that Q was changed, four years ago, three years ago, by probably a malicious actor. And other bugs, other subtle bugs in the code allow the direct output, basically bypass an intermediate security stage, which is that inline secondary pseudorandom number generator to directly expose the DRBG's output. And that's the prerequisite for being able to reverse engineer its state and then decrypt everything that that piece of hardware does until it's rebooted. And then you have to wait a little bit, and then you can start again. Just incredible.

Leo: Yeah, no kidding.

Steve: Really, really cool. So I can imagine, I mean, this is just like, as Matthew said, this is what applied cryptographers live for is just a cool super mystery like this.

Leo: Well, we'll never know, I don't think. I mean...

Steve: No, we won't know more. But we do know that here's another example. Basically a backdoor was found in a core Internet backbone device. And then, you know, how was this done? Because, I mean, it's been assumed that the NSA may be infiltrating deep sleeper agents into key Internet-oriented companies, and they have their own agenda which just never becomes public. I mean, unfortunately, this is the world we live in. It sounds like science fiction, but somehow a constant got changed in their source. Wow.

So Oracle was smacked recently by the FTC over, believe it or not, Java's security. There was a complaint that was settled, essentially, because the FTC, the U.S. Federal Trade Commission, was charging that Oracle deceived consumers about Java software updates, that is, like the security of them, and that the company will now be, as a part of the settlement, will be required to notify consumers of risk, of the true risk of Java, and provide tools to uninstall insecure older versions. So the FTC had a release to the public and the press when this settlement went public.

And they said: "Oracle has agreed to settle Federal Trade Commission charges that it deceived consumers about the security provided by updates to its Java Platform, Standard Edition" - that's the Java SE software - "which is installed on more than 850 million personal computers," which of course they brag about whenever you have to update Java and turn off all the extra crap they're trying to install on your system. "Under the terms," writes the FTC, "of a proposed consent order, Oracle will be required to give consumers the ability to easily uninstall insecure, older versions of Java SE."

Jessica Rich, who's the director of the FTC's Bureau of Consumer Protection, was quoted: "When a company's software is on hundreds of millions of computers, it is vital that its statements are true and its security updates actually provide security for the software. The FTC's settlement requires Oracle to give Java users the tools and information they need to protect their computers." So then the release goes on, saying: "Oracle's Java SE provides support for a vast array of features consumers use when browsing the web, including browser-based calculators, online gaming, chatrooms, and 3D image viewing.

"According to the FTC's complaint, since acquiring Java in 2010, Oracle was aware of significant security issues affecting older versions of Java SE. The security issues allowed hackers to craft malware that could allow access to consumers' usernames and passwords for financial accounts and allow hackers to acquire other sensitive personal information through phishing attacks.

"In its complaint, the FTC alleges that Oracle promised consumers that by installing its updates to Java SE, both the updates and the consumer's system would be 'safe and secure' with the 'latest security updates.' During the update process, however, Oracle failed to inform consumers that the Java SE update automatically removed only the most recent prior version of the software, and did not remove any other earlier versions of Java SE that might be installed on their computer, and did not uninstall any previously released versions prior to Java SE version 6 update 10. As a result, after updating Java SE, consumers could still have additional older, insecure versions of the software on their computers that were vulnerable to being hacked."

And I'll just stop for a second and say we talked about this years ago, the fact that the older versions are explicitly addressable by code running in browsers that can say, "I want this version of Java," and the browser will use that version of Java if it has multiple versions to choose among. So this was really being exploited on an ongoing basis, and nothing about what Oracle was saying made this explicit.

So finally they said: "In 2011, according to the FTC's complaint, Oracle was aware of the insufficiency of its update process. Internal documents stated that the 'Java update mechanism is not aggressive enough or simply not working,' and that a large number of hacking incidents were targeting prior versions of Java SE's software still installed on consumers' computers." So basically Oracle was just saying, eh, you know, implicitly saying we don't care about that. We're updating Java, but oh well.

So, "Under the terms of the proposed consent order, Oracle will be required to notify consumers during the Java SE update process if they have outdated versions of the software on their computer, notify them of the risk of having the older software, and give them the option to uninstall it. In addition, the company will be required to provide broad notice to consumers via social media and their website about the settlement and how consumers can remove older versions of the software. The consent order will also prohibit the company from making any further deceptive statements to consumers about the privacy or security of its software and the ability to uninstall older versions of any software Oracle provides." So, smack. Yikes.

You know, Java was a major topic, and not in a good way, of this podcast for years. And it was sort of like the pre-phishing email topic. We've gone through phases of the podcast over the last decade, and Java, you know, it was just a constant whipping boy because it was just a constant source of problems. And I do remember that there was some overt reason why older versions had to be kept around. Now, remember that the distinction is what your browser has access to. It's the browser plugin that interfaces your browser to Java installed on your system. That's the problem, that bridge between the Javas, maybe plural, on your system and the web browser. That's the source of the vulnerability.

And unfortunately, what Java has traditionally done was to offer up its services as a plugin, just like Flash. And unfortunately, just like Flash, Java is a huge massive old library written in the dawn of the Internet, pre-security focus days that is just riddled with areas in the code which work, but which can be exploited. And so it was just - it was a cat-and-mouse game for years of bad guys finding problems and exploiting them until it became public. People got hurt. Then Java, or Oracle, originally Sun, then Oracle, would update it and push out a patch.

I think, though, that there was a reason, as I recall, there was something about an earlier release that some people had to have. It was like, you know, their corporate software depended on this version of Java, and they could not update, or it would break some critical corporate software whose programmers had long since left, and they'd lost - they took the source code with them, or they deleted it, or they lost it, or no one knew how to recompile it. They couldn't maintain it. And so it became a fragile system that they couldn't change.

So I can kind of see a little bit of that problem. But it is the case that, if Oracle's own internal documents are saying, you know, we really ought to be doing a better job of this, that they probably should have been. And now they've got a big slap by the Federal Trade Commission, and they're going to - it'll be interesting to see what the next Java release looks like for us. I have several things I need it for, but there is no presence of Java, which is in my system, there's no presence in my browser. There's no way for anything, any site I run that I access, to run Java.

Leo: That's key because a lot of people want Java. If you're into Minecraft, you need Java.

Steve: Yes, yes. Perfect, perfect example.

Leo: As long as you don't have, yeah, as long as you don't have it launch in the browser, you turn it off in the browser, you're safe; right?

Steve: Yes.

Leo: And don't download, obviously don't download JAR files and run them, random JAR files.

Steve: And say, oh, I wonder what's in here.

Leo: Wonder what's going on?

Steve: And left-click instead of right-click.

Leo: Actually, that does happen because there are a lot of Minecraft mods out there. And I see Michael, who's, as with all 13 year olds, he's really into it. He'll just download anything from anywhere. And there are modifications for Minecraft. But of course they could be anything. We don't know. I mean, it's just...

Steve: Yup. And as it becomes more popular, it becomes a greater, a larger target.

Leo: Yeah. That's why I made him a limited user in Windows.

Steve: Nice.

Leo: Yeah.

Steve: So, okay, this one was a - this was fun. The industry got a kick out of this. The good news is it's not a remote exploit for Linux. It is a local exploit only. But Ubuntu, Red Hat, and Debian all immediately released emergency patches. Okay, so Grub is the abbreviation for Grand Unified Bootloader, G-R-U-B. And we're at Grub2 now, which is used by most Linux systems to boot the OS. And so what you get is a password prompt from Grub saying, you know, what's your password, and then we will go ahead and boot your Linux.

Well, it turns out that there was an innocent buffer or integer underflow in the Grub code. It had been in there since Grub v1 point - oh, my god, this is where I got 2009 in my head. Sorry. It was 2012 for Juniper. It was December of 2009 for Grub v1.98. That's when this problem, it was introduced in a single code commit. And since 2009, anybody who, when presented with that Grub password bootloader prompt, if instead of guessing, they simply pressed backspace 28 times...

Leo: [Laughing]

Steve: It's all you have to do.

Leo: Oh, that's funny.

Steve: It crashes out of the prompting and launches the Grub rescue shell, which allows unauthenticated, no authenticated access needed to the computer, with the ability to load another environment. Researchers have verified that, from this shell, which you access by hitting backspace 28 times, a local attacker, somebody at your machine, could gain access to all the data on your computer and can misuse it to steal or delete all the data, install persistent malware, rootkits, whatever they want.

So not good news. Again, only vulnerable at the keyboard, and the various major Linuxes, the distributions have all said whoops and updated to Grub 2.03. I think that's it. So under Grub v1.98 to v2.02 this has been present. And so 2.03 or later this will not be a problem. But, so if you forget your password, just try hitting backspace 28 times. Maybe you'll get in.

And I heard you talk about this over the weekend, Leo. It was sort of interesting. And also I got a kick out of the fact that Zuckerberg was making a comment, apparently from the nursery, after this happened. And that is that WhatsApp was banned by a judge in a lower court in Brazil for 48 hours, that is, for two days, as punishment for refusing to allow law enforcement access to some encrypted communications that they wanted. So this is interesting in the dance that we're beginning to see start in 2015, that I have very little doubt is going to be playing out next year, through 2016.

So Reuters picked up on this. There was an update posted to the original story because the blockage or the ban was overturned by a higher court, by an appeals court, after only 12 hours. Yet some interesting numbers came out of this. So after about 12 hours - I'm paraphrasing from the Reuters report - a Brazilian appellate judge on Thursday ordered the lifting of a 48-hour suspension of the services in Brazil of Facebook Inc.'s WhatsApp phone messaging application, overturning an order from a lower court. The interruption of WhatsApp's text message and Internet telephone service caused outrage in Latin America's largest country, where the country estimates it has 100 million personal users, and led to angry exchanges on the floor of Congress.

WhatsApp is installed on 92.5% of Android devices in Brazil, making it the single most installed app in the country, according to an Internet intelligence and marketing company, SimilarWeb. By comparison, the rival messaging system, Telegram, noted on Twitter that it had received one million downloads in Brazil in one day due to the outage. Telegram was installed on 2.35% of Android devices before the blackout of WhatsApp, and Facebook Messenger on 74%.

So that's pretty much the story. A judge in an industrial suburb of Sao Paolo had ordered the suspension of WhatsApp services from midnight on Wednesday for 48 hours. And that order was made after WhatsApp, despite a fine, failed to comply with two judicial rulings to share information in a criminal case. Then the judge who overturned the lower court order said: "Considering the constitutional principles, it does not look reasonable that millions of users be affected as a result of the company's inertia to provide information." However, this overturning judge then recommended that a higher fine be imposed on WhatsApp. So it's not like they were off the hook, but rather don't do it this way, let's just fine them more.

So the Reuters story ends by saying: "The incident highlighted growing international tensions between technology companies' privacy concerns and national authorities' efforts to use social media to recover information on possible criminal activities." And then Zuckerberg said: "Until today, Brazil has been an ally in creating an open Internet. I am stunned that our efforts to protect people's data would result in such an extreme decision by a single judge to punish everyone in Brazil who uses WhatsApp."

And then, finally, according to BandNews TV, the criminal case involves a drug trafficker linked to one of Sao Paolo's most dangerous criminal gangs. The trafficker allegedly used WhatsApp services while committing crimes, and the court wants access to his communications with others. And then WhatsApp has said that it was unable, not unwilling, to comply. So of course this is smack dab in the middle of the huge controversy about this. And in fact...

Leo: At this point, though, WhatsApp should just ignore anything because they know there's no punishment that Brazil can mete out. What, are you going to turn us off? Go ahead.

Steve: Yeah, see how your people like it.

Leo: Ninety-three percent of all Brazilians use WhatsApp because of the predatory pricing on SMS messages in Brazil.

Steve: Yes, yes, exactly.

Leo: Sorry.

Steve: Now, I listened to the Democrats debate as I listened to the Republicans debate just after our last podcast last Tuesday. The Democrats debated on Saturday. And I was a little annoyed by Ars Technica's coverage. And I understand that oftentimes people who write the body copy of the column don't put the titles on. I was often upset when my

TechTalk column, back in the days of InfoWorld, had a title which was really misleading. It was like, wait a minute, that's not, I mean, sometimes it actually was the reverse of what my column said. And I wondered if whoever it was on an edit desk somewhere had actually read my column.

But anyway, Ars Technica's headline read: "Hillary Clinton wants Manhattan-like project to break encryption." And then the subhead was "Presidential candidate Hillary Clinton has called for a Manhattan-like project to help law enforcement break into encrypted communications." And so many people retweeted this that I just wanted to say, no, she didn't say that. Not at all. What she said was, and this is I expect what we're going to be hearing next year, she called for - or maybe 2017. Maybe everyone's just going to punt until we have a new President, who knows.

She called for the tech community and law enforcement to work together to solve this apparently, and I'm paraphrasing, really hard problem. And so her allusion to the Manhattan project was of course the codename given to the U.S. effort to develop the atomic bomb. And so she's saying, you know, we need, like, all of the really smart people in Silicon Valley and the encryption community and the tech community to work with law enforcement and come up with a solution. And it's like, okay. We all believe there isn't one. And we've talked about how people are now understanding that maybe the word "backdoor" is too loaded or not what we want. But they're just saying, okay, yeah, well, fine, give us an answer.

And of course those who understand math appreciate that there isn't a way to do this without incurring obligation. And I can certainly understand companies like Apple - and I also, thanks to you, Leo, who turned me on to that, that he was going to be on "60 Minutes" on Sunday, where Tim Cook was interviewed - I can understand that they're not wanting the responsibility of having the ability to decrypt this. So it's better from a business model standpoint for them to be able to say no. And WhatsApp said, I'm sorry, we would be happy to comply, but we can't. Our technology doesn't let us. But wow.

Leo: Which is interesting because I don't - is WhatsApp, do they even pretend to be encrypted?

Steve: Yeah, they do. And they have pretty good encryption.

Leo: I guess they do, okay, good encryption. Oh, never mind.

Steve: Yeah, way better than Telegram, at least, because we know how Telegram's encryption is.

Leo: Yeah.

Steve: So I already spoke about this Web Privacy Census. I won't dig into it any deeper. For anyone who's interested, the link's in the show notes. It is really just eye-opening to see - they also show over time, over the last couple years, the massive growth we've seen. So it's not like people just a few months ago finally got tired, although we did. But what happened is over the last few years there's been exponential, I mean, literally the curve is exponential shaped, explosion in the use of tracking and loading hundreds of kilobytes, often multiple hundreds of kilobytes of JavaScript, just because some site says, oh, we're going to get a little bit of money if we put this little bit of code, or we add this tag to our pages. And they do, and users suffer the consequences.

So anyway, this group put together, as I mentioned at the top of the show, wrote a Python bot which spiders the 'Net and acts like a user. They had what they call - they had two different levels of depth, a deep exploration where the bot randomly clicked on two links to acquire, like, sort of to simulate what a user might do if they were clicking on links on pages, and then just the top level bot. And the stats I showed were just the top level because I don't think that the deeper bot is as representative of what, you know, just going to a page on one of these sites gives you. But, boy, it really has gotten crazy.

Meanwhile, Microsoft has upped the ante yet again. Woody Leonard, who is Woody on Windows is his InfoWorld column, the title of his column was "Microsoft narrows Win10 upgrade options to" - and this is just great. The dialogue that comes up now has two big buttons on it. The left one says "Upgrade Now." The right one says "Upgrade Tonight." Instead of "No."

Leo: Either now or tonight.

Steve: Yeah, you've got a - now we're giving you a choice.

Leo: Oh, man.

Steve: We took away "No, thanks." Now it's "Upgrade Now" or "Upgrade Tonight." And so he says, "Microsoft's nagging 'Get Windows 10' campaign has hit a confusing new low, and user backlash is vocal." And he wrote: "It's hard to imagine any marketing campaign worse than Microsoft's ongoing 'Get Windows 10' debacle."

Leo: Yeah, I agree.

Steve: "Microsoft is pushing hard for Windows 7 and 8.1 customers to upgrade to Windows 10, and the backlash from users has been vocal and very negative." He notes that: "Paying Windows 7 and 8.1 customers have been subjected to surreptitious installation of a potentially unwanted program, GWX, starting way back in April," which of course we talked about when people were worried it might be malware because they'd never seen Microsoft offering an upgrade, and they were suspicious, rightly so.

Then, "Incessant nagging by a balloon notification in the system tray that 'Your upgrade is ready.'" And remember "Get your reservation," even though it's like, what? You're going to run out of bits? And then they were forced to download three to five gigabytes of unwanted installation files, which was done behind their back, again surreptitiously. And then "Accidental" automatic launching of the upgrade program.

Finally, last week, Woody notes: "PC World's Brad Chacos detailed the evolution of the nagging GWX balloon notification into a full-fledged, and nearly full-screen, Get Windows 10 window with these two options: 'Upgrade now' and 'Start download, upgrade later.'" And so Brad at PCWorld wrote: "To be fair, you can still simply close the window with the X in the upper-right corner; and, if you click through the itty-bitty inconspicuous chevron" - the little menu icon, those three little bars - "on the upper right-edge of the window, there may be a 'Nope' prompt somewhere further down the line."

And he says, "I closed the prompt before exploring the auxiliary pages. But having the only two large, clearly actionable options on a pop-up page both lead to a Windows 10 download feels inherently icky, like Microsoft's trying to trick less-savvy computer users into downloading the operating system with tactics often used by spammers and malicious websites." So that's what it's come to. They really are determined to get people onto Windows 10. And you've got to wonder why. I mean, I do. I mean, it'd be - it's a pain to have to support older operating systems. The problem is, a lot of people don't want to go.

Leo: Yup.

Steve: Okay, miscellaneous. I know we're not doing any Star Wars spoilers. I have not yet seen it. I'm seeing it with Jen on the 28th. And I know you guys saw it, and you're biting your tongue. How long do we have to bite our tongues for? Like how...

Leo: Forever. Forever.

Steve: Forever? Okay, at some point...

Leo: Yeah, it's ridiculous. It's ridiculous.

Steve: Okay. What we'll do is we'll wait a few more weeks. And then at the end of the podcast, I think that's - I was thinking about this.

Leo: Yes, say "Tune out if you haven't seen it yet."

Steve: Okay. So if you - yeah. Leo and I and the listeners who have seen it, we need to be able to talk about this. We can't be held hostage forever.

Leo: Right.

Steve: So at the end of the podcast we will then have a Star Wars discussion. But not now.

Leo: I can't wait. I m dying to talk about it.

Steve: We did note, and I'm sure everyone knows, that it broke all first weekend sales records in history. Nearly a quarter billion dollars in the first weekend, and worldwide I think it was at half a billion. So, and still going strong. It's all sold out around me until the week after Christmas.

Leo: This is a movie you're going to want to see at least twice.

Steve: Neat, neat, neat. And, boy, Leo, Season 6 of "Homeland" finished. I'm sorry, Season 5 of "Homeland" finished. God, I've got my notes all tangled up. Are you a "Homeland" watcher?

Leo: I watched the first two seasons. I haven't seen it since then. So I have some catching up to do.

Steve: Oh, boy, you have. I mean...

Leo: I liked it a lot.

Steve: I know you're done with, or maybe you're not done yet, maybe you're not...

Leo: Finished "Fargo," yeah, yeah, yeah.

Steve: Okay. So I would argue this is even better than "Fargo." And this fifth season ended, it was just - it's probably my absolute favorite series on television. It's just so well done.

Leo: Got to watch it.

Steve: And the good news is there will be a sixth. My number one question two nights ago, last Sunday night's final episode, is okay, is - because kind of like they left us in a way where they could have, like, finished the series. But apparently it's just too successful. Showtime is just knocking it out of the park.

Leo: Was Season 3 not great?

Steve: I think that's what it was. I think that it hit kind of a rough patch. But, boy, it really did recover.

Leo: Okay.

Steve: Four was really good. And I wasn't even that crazy about Season 2. It kind of continued some of the character line from the first season. It was like, okay. But it has really - it hit its stride. Just these latest seasons are just fabulous. And you don't have to watch them all, for anyone who's interested. But it's better to have the back story of the characters because the character development is really deep and really rich, and so it just would mean more. And, boy, for anyone who likes to binge watch over the holidays, let me fully recommend "Homeland" on Showtime, five fabulous seasons that I think people will like.

And I got an interesting tweet to a link to someone's posting about how SpinRite saved his DVR. And we've talked about this before, so I won't drag everyone through it. But I did want to mention something by way of DVRs that have always made me uncomfortable. And he actually posted the SpinRite screens from the scan, which did repair - it was a Humax 1000 or something DVR. And so he showed, like, he took it apart, opened it up, pulled the drive out, hooked it to a PC, ran SpinRite. SpinRite found problems, and now everything's working fine, you know, the standard SpinRite recovery story.

But people have talked about recovering the problems with their webcam or security camera recorders. And DVRs are the same. The problem is, if a drive is not actively recording when the power fails, it's probably okay. But, for example, most DVRs don't have a power switch. My TiVos have no - no TiVo I've ever owned, I've owned Series 1s, and now I've got the Premieres or, I'm sorry, the Roamios. No power switch because it's just - it's supposed to be always on. It's got a power cord. You plug it in, and it boots and says hello.

But the TiVos are always streaming several most recently chosen live channels. The Roamio will record six things at once. And if I step through the most recent channels that I've selected, my 30-minute playback buffer is live for all of those. Meaning it's busy all the time. Yet there's no way to turn it off. You pull the plug. And this is the same thing for security cameras that are recording to hard drive. And so the point is the reason that - and this has always made me uncomfortable. And I've never really been able to understand it. The reason that SpinRite is finding problems on security camera recorders and people's DVRs, I mean, we talk about SpinRite recovering DVRs relatively often. It's because people treat it, not like a computer, which is like right in the middle of working, but rather a consumer device.

Our computers, we shut them down. And when we shut them down, they're very careful to stop everything that they're doing and carefully power the drives down. But when you just pull the plug on a box that doesn't even have a power switch, it's in the middle of recording data on the hard drive, as fast as it can, for multiple security cameras or TV cable streams, and it's going to wreck the sectors. It's going to destroy the low-level format in those areas. So I guess the calculation must be, oh, well, we want this to be a consumer device, a consumer product that people don't have to shut down or anything.

So I'll just - here's a tip for any TiVo or other DVR user. Reboot the device and then pull the cord. In the UI, you can do a reboot. There is no - you can put it in standby, but it still records in standby. So you have to initiate a reboot. And shortly after the reboot starts, pull the cord. And that gives it the equivalent of a shutdown, and it is in the process of getting started from ROM and hasn't yet had a chance to bring the streams up. And that's a safe time to then, for whatever reason, unplug it to vacuum underneath it or move it to a different room or whatever. But reboot it, and as soon as it starts the reboot, pull the plug. That's a way of preserving the surface of the hard drive.

They must figure that, oh, well, we'll just skip over that bad spot. The problem is, depending on where the damage is, it can damage recordings that you care about, which SpinRite is able to fix, or it can cause deeper damage. Because we talk about where there have been instances where the device, you know, the DVR was restarted, and then it didn't come up. We talked about that just a couple weeks ago. So just a little tip for everybody.

Leo: Very nice. You want to do questions?

Steve: I think we ran out of time again.

Leo: Not really. I mean, there's no newscast at 4:00 anymore.

Steve: Okay.

Leo: So if you want to go a little longer, I don't mind.

Steve: Well, yeah, let's go till 4:00 because we're at an hour and 35 at this point. So we'll give everybody about two hours' worth of...

Leo: Nobody ever said I want less Security Now!.

Steve: Yeah.

Leo: Bill Fink, he was a programmer, worked for Microsoft, lived in Belleville, Illinois, passed away, I'm sad to say. But I have a feeling that he had a great sense of humor, and he must have written his own obituary because this is how it reads:

"William Ralph 'Bill' Fink, 46 [sad to say], of Belleville, Illinois, encountered an unhandled exception in his core operating system which prematurely triggered a critical 'STOP' condition on Wednesday, December 16th. Diagnostics indicated multiple cascading hardware failures as the root problem. Though his hardware has been decommissioned, Bill's application has been migrated to the cloud and has been repurposed to run in a virtual machine on an infinite loop. <END OF LINE>"

Steve: That's wonderful.

Leo: We salute you, Bill, and your sense of humor.

Steve: I think he must have had some notice that there was a pending Blue Screen of Death, yeah.

Leo: Yeah, yeah. Beautiful, though. I mean, really, that's what I want mine to say. That's awesome.

Steve: That is neat.

Leo: All right, now, questions. Let's get to them. Remember we got one in last time. We're going to do better this time.

Steve: And I have to say I loved your intro to last week's podcast, where you said, instead of "Listeners questions and Steve's answers," you said "Listeners question and Steve's answer."

Leo: It was a good one. I really enjoyed it.

Steve: Perfect.

Leo: The first one comes to us from Funny Gazelle on the Twitter. And I know why he calls himself Funny Gazelle, because his Twitter handle is ridiculous. Is it hex? I don't even know what it is. @edool9Chah5fe2a wants to circumvent his provider: Steve, is it possible to circumvent the provider-level SSL decryption, where you have to place a certificate from the ISP on your machine, can you get around that with a VPN? When they decrypt one secure connection, can't I just place another secure connection inside that one, like kind of nested?

Steve: So, okay. I've seen questions like this, and I want to make sure I haven't confused people. Because as far as I know, unless your provider is your corporation...

Leo: Yeah, businesses do this.

Steve: Yes. And of course we heard that Kazakhstan, which is sort of a super provider...

Leo: They want to do it, yeah.

Steve: Yeah. They were considering it. But to my knowledge there is today no ISP who is requiring that you accept their certificate in order to use security on the Internet. I mean, it's frightening, and that's why I've been talking about it, because it seems like in some future Orwellian universe that may happen because it does represent an obvious choke point where law enforcement could require ISPs to install monitoring taps. And then the way to do that would be for the ISP to take responsibility on a per-subscriber basis for users putting certificates in their machine. That's going to be a day I hope we don't see for quite a while.

But the answer to Funny Gazelle's question about a VPN is that, well, it's sort of two ways. First of all, a VPN done right cannot be intercepted. Now, that's Juniper's VPN done wrong notwithstanding. But a VPN done right will pass through, like, any connection that's being made. And it just - a VPN looks like a standard network connection. So anything you do over it, whether you then add your own encryption, like with HTTPS, TLS, or not, or just email, everything is encrypted by that tunnel. The issue will be whether ISPs will then start blocking VPNs because it could be read as a means for their subscribers to avoid ISP-based taps.

And so a VPN would definitely work. The problem is it would be encrypted, and it could very well be that, again, in some horrible Orwellian future, that subscribers at the user level are denied use of VPNs by ISPs. Again, that seems like a stretch. It's hard to imagine. And until then, even if encryption were what the ISP was doing, a VPN would solve the problem. Maybe law enforcement would attack the VPN providers and say, okay, if you're going to provide VPN services, you need to allow us to set up monitoring on your exit node, you know, on the VPN server.

Leo: And that's why you use an internationally-based VPN; right?

Steve: Correct.

Leo: Preferably run in a nation that doesn't do that kind of thing, although I don't know how many nations will be around that don't.

Steve: Boy.

Leo: Yeah, after reading that WiFi thing, I thought, you know, I'm not worried so much about some Joe trying to get into my stuff as I am about a day when a government really wants - once a government really decides to spy on its citizens, it's going to be really tough, frankly. It's going to be very hard. And now is a good time to start listening to Security Now! and develop the skills you need. I'm not kidding. Seriously.

Steve: Yeah.

Leo: And I also told Lisa, I don't know if she's going to go for it, I want to get a gun and learn how to use it. But that's another story.

Steve: Oh.

Leo: She's always teased me about, oh, I've got a Glock somewhere in the closet and that kind of thing. Then as soon as I said, "I want to get a gun," she said, "No, you don't. What are you talking about? Stop. No, you can't." So I think it was all just a little bluster. Vern Mastel with the Bismarck Public Library, Bismarck, North Dakota - I love that town because of public broadcasting in Bismarck, North Dakota. He oversees a large public WiFi network: Steve, I administer a public wireless system at the Bismarck Public Library. It consists of a Meraki router, 17 access points. It connects to the Internet via a cable modem connected to our local cable ISP.

I installed the Meraki system three years ago in 2012, to replace a system that was built with consumer grade Netgear Access Points and a Smoothwall router. Hey, I like this guy.

Steve: Yeah.

Leo: He obviously knows what he's doing. I had become accustomed to the existing system, which could only tell me how many logins there were per month. I was not prepared for the wealth of data the new system would supply. Now the system's logging, and I can see almost in real time how much traffic is flowing, what devices are connected, what traffic is being generated by whom. Devices are identified by MAC address, machine name and operating system. Many have an easily identifiable machine name, like TommyBoy or CheezEater. Obviously from Wisconsin. It will tell me what access point a device is currently associated with. At the end of every month I have a nice detailed report listing traffic by type like BitTorrent and Google Video, traffic by device and user, and by access point.

Steve: Wow.

Leo: Yeah. When I discovered the extensive data gathering features, I asked Meraki, how do I turn that off? I didn't want to have user identifiable data in case the authorities ever came calling with a court order. Meraki told me, not possible, the data gathering cannot be turned off. And I receive a DMCA warning letter at least once a month warning me about copyright violations on the cable connection. What a surprise. Given that BitTorrent alone accounts for about 10% of our monthly public wireless traffic, the real surprise is I'm only getting a letter a month. What, kids go to the library now to share music? Wow. So in my limited experience, my WiFi is not an anonymous communications system.

Finally, I'm astounded by the number of WiFi hotspots that exist. I have the system configured to warn and react - oh, this guy's good - when anything broadcasting as a hotspot comes within range. As you should because a bad guy setting up spurious hotspots; right?

Steve: Yup.

Leo: Guess what? The system logs 500 to 700 every month. Wow. Thank you.

Steve: So I just really liked this little report from the field, that here is, as you said, obviously a tech-savvy Security Now! podcast listener who is running a major WiFi installation, 17 access points. So this is a substantial public installation with lots of activity. He's tried to be proactive in saying, no, I don't want access to this information. He doesn't want to have to serve information on people who use the library, assuming that it's an access-friendly location. And this information is being generated, like it or not.

Leo: Wow.

Steve: Wow.

Leo: That's too bad. Michael O'Rourke, Cambridge, England poses a question about assuring privacy: From a privacy perspective, how can an ordinary law-abiding person determine if they're being placed under surveillance - oh, this is good - in their ordinary PC usage? Within reason, are there any telltale signs to be aware of in ordinary, day-to-day usage of their PC? I bet the bad guys already know. So what should the good guys be aware of, within reason?

Steve: You know, this was, as you said, a great question. And the problem is that what we're telling law enforcement, you know, we the security community, since we're saying no, you cannot have backdoor access to encrypted connections, we're saying you need to get access before it's encrypted or after it's decrypted. So this of course is why, back when the Snowden revelations first occurred, people like Google and Facebook and others were immediately implicated because that would be the decryption on the far end, where the encryption has been removed. And the question was, you know, are these major companies complicit in helping law enforcement to have access to the post-decryption data?

The other side is the pre-encryption, which is the famous keystroke logger. And we know that law enforcement has malware, or what we would call malware - they don't, I'm sure they call it surveillance-ware - because we know, for example, that recent company that had its whole database breached, it was selling its surveillance software to U.S. law enforcement companies through shell corporations so that it wasn't obvious who was buying it. And so they were arranging to install things on people's computers. And so that's the get something, get a shim installed before the encryption in order to monitor what a person's machine does.

The problem is these are stealth things. They're rootkit technology. They are very good at hiding themselves. They're typically undetectable by antivirus systems. We know that our - we've been talking the last few months that there are now BIOS-resident technologies that get themselves installed even before the machine boots. So the fully honest answer is it's a great question without a great solution. How does a law-abiding person know if you're under surveillance? I think, unfortunately, in this day and age there probably isn't a way to know, and you for that reason should stay law abiding.

Leo: Assume surveillance.

Steve: Yeah.

Leo: Denny in Portland, Oregon worries about knowing someone's password length: Steve and Leo, listener since Episode 1. Whenever I've had conversations about online security, the topic invariably turns to password length. Usually someone will mention the length of a password. For example: "My Amazon password has 20 characters." I cringe when this happens. It seems to me revealing the password length would allow bad people to focus their brute-force attacks because they could ignore all the other lengths. Saying a password has a maximum of 20 characters seems very different from saying it "is" 20 characters. Should I be keeping my password length secret? Or is this a non-issue since it would effectively save 100 years from a 10,000-year brute-force attack? What are your thoughts? Thanks, Denny.

Steve: Well, he answered his own question. I agree that, I mean, it's always the case that you want to provide the least information possible. And of course, as we mentioned before, the reason some people think that the math is wrong on my Password Haystacks calculator is that they assume that I'm only calculating, in this case, the length of time to crack a 20-character password. But I recognize that attackers don't know the length. And so what the calculator does is to sum the number of combinations of 20 characters, 19, 18, 17, 16, 15, and all the way down, to get a substantially larger number. But if your password is 20 characters, and if it's high entropy, if it's 20 random characters, then you have nothing to worry about. So be proud that it's 20. I agree with Denny, it's not good to say it. On the other hand, he says that this is someone mentioning it. Well, the attacker...

Leo: Don't post it on the Internet.

Steve: Right.

Leo: But saying it to someone, I could tell you my password length. I just don't want to announce it to the world.

Steve: Exactly. Exactly. And so the spoken length over coffee isn't going to represent any decrease in security because a bad guy scanning the 'Net and finding a machine isn't going to know anything about this particular guy and can make no assumptions.

Leo: Right.

Steve: So but I thought it was an interesting question. And I think...

Leo: It is a good question, yeah.

Steve: Yeah.

Leo: Question 5 comes from an anonymous listener. And I figure he was representative of maybe 58,000 emails you got about this subject?

Steve: Uh-huh, exactly.

Leo: Steve, you call for a law compelling Apple and presumably others to have the ability to decrypt on a phone-by-phone basis. You are assuming Apple would not have a Snowden to leak that ability, in whole, to the world or bad guys. If the NSA could have a Snowden, what makes you think Apple wouldn't also?

Steve: Okay. So you're right, Leo, I got, of course, my saying that I thought Apple's ability to respond to a court order was probably the compromise we were going to see.

We'll see. I hope I'm wrong. But I'm acutely aware of this tension. And the tension is not going away. And we do not have - I also am acutely aware of Congress often making bad decisions. The DMCA is a bad idea. I mean, so bad things can happen. And this would be - I'm not saying I wish things were different than they are now. I love the way they are now, that Apple can say "We cannot decrypt."

So, but he did mix two things. And so the reason I wanted to mention this is there is a difference between a backdoor and what I said. And in his question is the essence of the confusion. He said, "You're assuming Apple would not have a Snowden to leak that ability." No. I'm not suggesting that they add an ability. And that's the difference. If there was a secret, one secret, which we would call a backdoor, which if the government knew the secret they could access anyone's phone, that's a backdoor. What I was proposing is that Apple maintain a database for which there is no algorithm, where on a person-by-person, on a phone-by-phone basis, a high-entropy random value is generated, and Apple stores it. And that's the master key for that one phone.

Now, there is a huge obligation that Apple has to protect that database. So, yes, it creates a vulnerability. There again, that's - there's no way to do this without creating a vulnerability. And so all I'm saying is it's not a backdoor. I'm not saying that I think it's a great idea. I've been a proponent of what Apple is doing and this kind of security. And it's because I wasn't sure I would be allowed to have this kind of security that I killed the CryptoLink project that I had already made a big investment in. So, again, I watched Congress do dumb things. And this seems like a compromise that I wouldn't be surprised to see happen. But again, it isn't something where a single secret unlocks the keys to the kingdom. It is explicitly not that. It is individual phones would be kept in a database. And if Apple had to respond to an individual court order to unlock a phone that had been captured, for example, after an attack, they could do so. So it is a compromise, and maybe one that we're going to be forced as an industry to make. We'll see. We may not have to wait long.

Leo: Yeah. Norman Davie of Vancouver, BC, Canada brings us our "You're Doing It Wrong" faux pas of the week. You're doing it wrong: Microsoft recently sent an email telling people to upgrade their Windows Live Mail with a link to click on to perform the patch, instead of just telling them to go to Windows Update. So it took no time for virus writers to copy the content and point the link to a TeslaCrypt infection site. Norman Davie, President, Exceptional Computer Services, Inc. And he sent us a link. Oh, doesn't work anymore.

Steve: Okay. Well, I did capture, I captured it from his Facebook page link. And this is what it looks like. And, I mean, it's 100% official Microsoft looking. It's in the show notes here. It says Outlook.com, "Important information about your email service." And, you know, Dear User. So it's Microsoft's letter. But they actually sent it out with a link to click here. And in this day and age I hope everybody and their spouses and significant others and kids have been told over and over and over, never click on links in email, no matter who it comes from. If it comes from Grandma at Christmas time with a Christmas card, just do not click on it. Go to Outlook.com and follow from there. Again, the best wisdom of all is, if you did not go looking for it, do not do it.

Leo: Joe in Cleveland wonders about the future of NAT routers: Steve, will IPv6 make NAT routers obsolete? Long-time listener, first-time questioner: You've mentioned many times how NAT routers are important to IPv4 after the address shortage, and for security. But what's going to happen when we shift to IPv6 and everything has its own IPv6 IP? We won't need NAT to separate public and private. And without NAT we'll lose all the protections we have from it today. It seems like we'll need a new type of home firewall appliance. How do you think we'll handle this in the future? Thanks for all you guys do.

Steve: So it's a great question. And I've been sort of following loosely the various discussions going on. With 128-bit IPs, again, this is one of those things where our brain doesn't get around it very well because 128 bits only seems like, well, that's only four times more bits than we have now. We have 32. If you double that, you get 64. You double it again, you get 128. It doesn't seem like that big a deal. Oh, my lord. I mean, it's a crazy number of IPs. So much so that there is discussion, and this isn't all settled yet, and there may be some ISP-to-ISP variation, but it may be that individual subscriber networks like a person at home would have their own 64, wait, no, 32-bit chunk of the Internet. I think ISPs get 64 bits, and they might be then allocating 32-bit chunks. Which means an ISP would have 4.3 billion subscribers, each with an Internet size with 4 billion IPs. I mean, that's how many we have. When we go to IPv6, it's crazy.

So as we know, the reason NAT is being used in consumer routers is that right now we get one IP from our ISP. And because this sort of hails from the day that we used to have one computer on the DSL dialup or something, you know, that was what we did. And then NAT routers allowed us to use network address translation to get many more. It looks like when we move to IPv6 we will at least get, you know, the number doesn't matter. You'll get more IPv6 IPs than you could ever possibly be using at the same time. So they will all be present.

The good news of this is that there are problems that NAT creates. And we talked about this. The Universal Plug and Play problem where, for example, UPnP was created to allow a device like an Xbox to be able to tell the router to open ports so that unsolicited packets could get to it. Increasingly, we are seeing peer-to-peer network systems. Peer-to-peer networks have a problem because - it's called "NAT penetration." The fact that the term "NAT penetration" even exists is testament to the fact that NAT has to be penetrated somehow. I mean, and the users want that. But it's a technical problem. The good news is that goes away.

So there will probably still be a NAT router, or a router, which probably does IPv4 NAT, but is also IPv6 aware. So it'll be more sophisticated. And the Xbox, for example, will get an IPv6 address. And the router, without even being told, will automatically send incoming IPv6 packets on that IP, on the Xbox's IP, to the Xbox. The point is that right now, with normal consumer NAT, all the traffic outside has that one IP. So it's up to the NAT to then distribute them inside.

When we go to IPv6, what we'll start seeing is, from this one consumer, a myriad of IPv6 packets emerging from the border, from the consumer's network, going out onto the Internet, and IPv6 addressed return packets will come right back through and back to the equipment. So there may still be a stateful firewall, where, for example, unless it's told it may, unsolicited incoming traffic still gets blocked. So that it's - and that's the way I want to run, once this happens. I absolutely don't want my internal network to be scannable, if that's what we're talking about. We're talking the same way that the Internet can be scanned by Shodan to find problems, individual consumer networks could be scanned by Super Shodan to, like, find all their light bulbs and other nonsense that they've got hooked up in their network.

So we absolutely don't want that kind of lack of protection. So I think we'll still see a stateful firewall feature, yet devices inside the network will have unique IPv6 addresses from a big block that the ISP has allocated to them. And incoming traffic that is expected, solicited by having some outgoing traffic first, it won't have to, like, wonder what device it goes to. It just goes back into the network. And then the network, as any network does now, routes it to the proper device. So, wow, you know? If IPv6 ever happens. It's still struggling. Here we are at the end of 2016. What, we were talking about the death of IPv4 a couple years ago. And, yeah, we're all still using it. But ultimately, as we know, the industry only makes moves when it's exhausted every other possibility.

Leo: And there are, I presume, some issues with IPv6, you know, things that, problems that have to be solved and stuff; right?

Steve: Yeah, that's the problem is compatibility. It's like no one wants to have any problems, so they just don't use it.

Leo: Right. Don't blame them.

Steve: Yup.

Leo: Eric Sarratt in Sylva, North Carolina wonders how he can determine whether his SSL email truly is secure: Steve and Leo, is there anyway to tell if my IMAP (port 993) SSL email connection is secure? I often send email from my personal laptop from my place of work, and I want to make sure my employer is not intercepting and man-in-the-middling my SSL connection between my email client, which is Thunderbird, and my email server. How can I tell if the connection is being tinkered with? Yeah, that's a good point because within a browser you could see the certificate. Is there some easy-to-use website that will check the connection for me? If not, an explanation of "the hard way" will also work just fine. Thanks, Eric.

Steve: So the problem is that you want to know whether the connection from your laptop to a remote server is encrypted. So, and that's a problem. You can't really use a remote site because that's going to be coming from a different direction through a different network to your email server. I poked around because I was curious about this myself, because I thought it was interesting. And it turns out that you can use OpenSSL to establish a connection to your - I don't know if he mentioned IMAP. I think he did somewhere because he was talking about...

Leo: Yeah, 993 is IMAP, yeah, for SSL.

Steve: Yeah, yeah, his IMAP port 993. Right. So, again, this is the hard way. But OpenSSL is available safely for download from, like, legitimate repositories like OpenSSL.org, where you could load it in Windows or Mac or whatever you're using. It's a command line tool. And in the show notes I gave the two commands that you could use to cause OpenSSL to connect securely to an IMAP - in my example I used imap.remote.com. See, the problem is you want to see the certificate because maybe you would know if your personal laptop was bearing a certificate from your employer, in which case there's a chance they could be, and this is exactly what Eric's worried about, man-in-the-middling his connection.

So the only way to really know - because, for example, a browser you can examine the certificate. You click on the URL, and you say show me the certificate, and you can see if it's like your employer, or if it is a legitimate certificate issued by a legitimate root certificate authority. You don't have the ability to do that in an email client. It just does it magically behind the scenes. So what you need to do, you need a way to see the certificate which a remote server is returning to see if it looks like it's not from somebody who might be in the middle.

And what these two command lines I gave, for example, it's openssl s_client -connect, and then the domain :993, telling it you want to connect to imap.whatever, you know, your IMAP server name is on port 993. OpenSSL will make that connection and then dump the certificate which the remote server provides. And then you can examine it and determine exactly what - because this is exactly what your own email client making the same connection would receive. So definitely the hard way, but it's the way to get the information.

And it's, you know, kind of cool to mess around with OpenSSL. You hear us talking about it all the time. It's both a library which can be built into code for its own internal use, but it is also a command line, an amazing toolkit that allows you to examine certificates as they fly around the 'Net.

Leo: How did this happen, Steve? We got through all eight of them.

Steve: Oh, my god.

Leo: Steve Gibson is at GRC.com. That's where you can find SpinRite, the world's finest hard drive maintenance and recovery utility. It's Steve's bread and butter, so go buy a copy, even if you don't need it. You will. You will. He also offers a lot of free stuff there, including this show, 16Kb versions of the audio, 64Kb as well. And something unique to Steve, a full, beautiful, human transcription of the entire show, so you can read along as you listen. You can also search it, which is really the chief value of it.

Steve: I use it all the time in order to refer back to previous podcasts.

Leo: And Google indexes it, so you can use Google to search it, yeah.

Steve: Yup.

Leo: We have audio and video, as well, of the show at TWiT.tv/sn. You can also find a subscription pretty much everywhere podcasts exist. I can't think of a single place that wouldn't have Security Now!, one of the oldest, best [crosstalk] podcasts.

Steve: It's funny, in the early days of the podcast, I used to google it, you know, in the way that some people google their name, because I could sort of see it spread. And now it's like, now it's not even - I haven't - it's been years since I bothered.

Leo: Not necessary. Not even necessary. It's everywhere. Steve, lots of fun. Next week we're going to do a Best Of, an episode we haven't heard in, I can't believe it, almost five years.

Steve: Six.

Leo: Six.

Steve: Yes, it was '09.

Leo: The Vitamin D episode. And it doesn't have anything to do with security, but it really does have to do with your health. And I think over the six years intervening, the medical community has absolutely come around to your point of view.

Steve: Yeah, we were early on this. I read a couple books, and I - I think people, even if you're not, if you don't think that sounds like it's interesting, I surprised everybody because I take a deep and sort of historical biochemical approach to define and explain the power of that hormone. It's not a vitamin because our body makes it. It's actually a hormone. And almost everybody is unfortunately deficient because we're no longer running around naked in South Saharan Africa near the equator, hunting and gathering during the day. We've got clothes on. We're behind glass, which blocks UVB. And the only real source is our skin, which synthesizes it. But it's crucial.

So I think people will actually like it. And it'll be a refresher for people who've been listening since before '06, but I know we've got a ton of people who have maybe heard us refer to it briefly from time to time. But, yeah, I think it's really important.

Leo: Great stuff. And then we'll be back, as you said, on January 5th.

Steve: Yup.

Leo: And every Tuesday thereafter at 1:30 p.m.

Steve: For the rest of time.

Leo: For the rest of time, 1:30 p.m. Pacific, 4:30 Eastern, 21:30 UTC, right here at TWiT.tv. Thanks, Steve. Have a merry Christmas.

Steve: You, too. Have a great Christmas vacation, and I'll talk to you in two weeks, my friend.

Leo: Happy New Year. See you in 2016.

Steve: Right-oh.

Leo: On Security Now!.

Steve: Bye-bye.