
# Listener Feedback #227

**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-542.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-542-lq.mp3

---

SHOW TEASE: It's time for Security Now!. Steve Gibson is here with an update on all the security news, including a new low in "You're Doing It Wrong." Plus, of course, 10 of your questions and his answers. It's all coming up next on Security Now!.

**Leo Laporte:** This is Security Now! with Steve Gibson, Episode 542, recorded Tuesday, January 12th, 2016: Your questions, Steve's answers, #227.

It's time for Security Now!, the show that protects you and your loved ones from the dangerous environs of the Internet. Here's your captain. He's kind of like the captain on the Jungle Boat at Disneyland. He's got a pistol; and, if a hippo attacks, he's ready.

**Steve Gibson:** I have a cap. If I was going to be the Jungle Boat captain, I should have my captain's cap.

**Leo:** That's Steve Gibson. Hey, Steve.

**Steve:** Hey, Leo. Great to be with you again in our 11th year of the podcast.

**Leo:** Whoa. Amazing. Just amazing.

**Steve:** Yeah, it is. So we have a Q&A. I dumped the mailbag and sorted through it and found a bunch of interesting stuff. And we've got a bunch of news. Trend Micro has,

perhaps more drastically than anyone else yet, lowered the bar on how to do it wrong. It's just unbelievable what they have done, and what is currently deployed in all of their customers' systems...

**Leo:** Oh, no.

**Steve:** ...is horrifying.

**Leo:** That's an antivirus that's widely used.

**Steve:** Yes, yes. And large corporate adoption. Anyway, so our friend Tavis Ormandy found some problems that we'll be talking about. Tavis, of course, is at Google. Symantec made some mistakes of issuing SHA-1 certs after the beginning of the year, which is a big no-no. But the way it happened is kind of a classic bug that we'll talk about. Firefox decided that they were going to stop accepting certs issued after 2016, then had to back off for a really unsuspected reason.

This is a special day in the Windows XP ecosystem, which I am shedding a tear. This is a significant day. It is also Patch Tuesday, but they just came out. And so always now on Tuesdays, when patches don't come out early, I have no chance to go through them. So everyone should just be aware that it's Patch Tuesday and make sure you are patched. I'm sure you'll want to. I promised to talk about how LastPass's new Emergency Access Feature that was introduced in v4.0, how it can be TNO, how they can do this without lowering security. So I will explain that.

I had an eye on Mark Russinovich's new version of his SigCheck utility we'll talk about. And we've got a bunch of miscellaneous stuff, some media stuff, some sci-fi stuff, a fun SpinRite story. I even had a tweet from someone saying - who already read the notes that I posted online, saying "I really liked that SpinRite story." So we'll do that. And then we've got 10 questions and comments from our terrific listeners. So another great podcast.

**Leo:** Wow. Great show ahead.

**Steve:** In our 11th year and going strong.

**Leo:** I just received a box from Wayne in Florida, one of our regular listeners. I'm very, very excited. He sent me a brand new BlackBerry Enterprise Edition. He says, "Never been used. It's okay for the Rogers and the AT&T network." Unfortunately, I don't know if I'll be able to use it, Wayne, because it says, "To charge, drop it into your BlackBerry cradle." And I'm afraid I don't know if I have a BlackBerry cradle.

**Steve:** That was the one with the scroll wheel on the side?

**Leo:** This is the classic. Look at that.

**Steve:** Yeah, where you scroll on the side, and then you push in, in order to select.

**Leo:** Right. That's the click. And you know what, if I could charge this sucker, I would use it. Although probably that network is no longer compatible. I don't see anywhere to put a SIM card in it.

**Steve:** No. I think, well, but now, here's the problem. See, I have every Atari machine ever made. All of those, the early ones, the later ones, the Apples and so forth, because I keep stuff. And you guys were talking about Treos.

**Leo:** Right, right.

**Steve:** Might have been like last TWiT or something. And I've got my Treos, I've got all my various Palm Pilots just because, for me, I bond to them in some strange way, I mean, even the PDP-8 computers that are flashing behind me. But now you're in this new, if it doesn't spark joy, drop it in the round filing cabinet.

**Leo:** Yeah, yeah. This sparks joy, though, wait, I've got to tell you, it sparks a lot of joy, even though it's totally useless.

**Steve:** It's nostalgic.

**Leo:** Yeah. Wow. And you know what, this you can type with, this keyboard. This is not like...

**Steve:** Oh, that original keyboard was so good.

**Leo:** Yeah, yeah. Wow. I used to have one of these on my belt. Remember? Because they had that hard plastic clips, the belt clip, too. Man.

**Steve:** Yup, yup. In fact, that was the early messaging platform. That was, more than anything else, people were using both email away from the office, strong corporate buy-in, and also it had BBM. The BlackBerry Messenger was really strong there.

**Leo:** Oh, yeah. Somebody said, "Call your member of Congress. They probably have a BlackBerry charging cradle lying around." This is a Black...

**Steve:** And it's funny because I've seen, because I've been watching a lot of news, with all this crazy election stuff going on, I've seen there are still reporters carrying BlackBerrys.

**Leo:** Oh, yeah. Absolutely.

**Steve:** That's still the phone that they have.

**Leo:** I think you go in a lot of boardrooms, a lot of government offices, this is probably something like you're going to find there.

**Steve:** Because of the keyboard.

**Leo:** The keyboard.

**Steve:** It's funny because we had, I think it's toward the end of our Q&A, maybe, I think, oh, I think I put it at the end, something that we never talk about is we don't do predictions on this show except, well, I guess I do predict, like, okay, from a security standpoint, what's clearly going to happen based on the conditions that have been set up. But we don't, like someone asked for our feelings about what the next five to 10 years has in store, more broadly. And I thought, well, that's kind of fun because we don't do that. So we're going to today.

And one of the things that I noted was that keyboards have finally succumbed on mobile to touchscreens. Even though BlackBerry isn't giving up, they qualify now as an outlier because it's like, well, okay, fine, but that's not what actually anybody's using anymore. Even I gave up. When I give up a keyboard on a mobile device, then that's a really - that's a harbinger of new things.

**Leo:** That was the thing I learned with the BlackBerry Priv is I don't want to go back to a physical keyboard. I've gotten used to the onscreen, as you have. So you got rid of your, what was it, Ashton Kutcher, the keyboard?

**Steve:** There was a - who was that guy?

**Leo:** Greg Kinnear?

**Steve:** Oh, some clown.

**Leo:** I conflate all of these clowns, yeah.

**Steve:** Yeah. You know, a big VC guy who fell in love with it. Well, it was only for the iPhone 5. And they did have to go to the big screen, and they did not follow it up into large-screen territory.

**Leo:** Right, Steve. Let's launch into the news.

**Steve:** Okay. So this is just gobsmacking, as they say in the U.K. Trend Micro, who as you noted is a major AV supplier, had a really interesting back-and-forth with Tavis Ormandy of Google, who has been responsible for finding a number of vulnerabilities in major Internet things in the past. We've talked about him from time to time. For whatever reason he was looking at Trend Micro's antivirus for Windows product, which by default includes and installs a password manager component, which is also set to start up at launch time. So it's just - it's there and running. That product is primarily written in JavaScript using Node.js, and it opens multiple HTTP RPC ports. RPC stands for Remote Procedure Call. And so that's sort of a term used for network-based automation, where you're able to either effect changes by sending data or request that data be returned to you, RPC (Remote Procedure Call), for handling API requests that the system uses.

So Tavis writes, and there's a log that Google had private while this was going on. When Tavis created the entry, it started Google's famous 90-day countdown where we're notifying the company, they've got 90 days to deal with this. And whether or not they have dealt with it, this goes public at 90 days. And we've talked about that in years past where that's been sort of a mixed blessing. Google themselves have sometimes not met their own deadline and so forth. But that's what they're doing.

So now what went public is the dialogue that had transpired where Tavis first writes, he says, as a first posting in this blog, in this off - what was originally a private log, which is now public. He says: "It took about 30 seconds to spot one that permits" - meaning these RPC calls - "to spot one of those calls that permits arbitrary command execution." The call is named openUrlInDefaultBrowser, which eventually maps to the ShellExecute, which is the way Windows runs anything. So he says: "This means any website can launch arbitrary commands" on the user's machine.

And he gives three lines of code. The first one creates an XMLHttpRequest object. That's newer JavaScript, which is the way that scripting is able to initiate outbound connections. In this case, it makes a connection to its own local machine, which is famously 127.0.0.1 is the small range of IPs that are reserved for so-called localhost, that is, for the machine sort of to use its network stack to talk to itself, which turns out to be very handy. But it can also be referred to using the term "localhost."

So in this case he gives the example localhost:49155. So that's the port which this poorly written code has opened and is listening to for commands. And what this means is that a browser can issue commands to the Trend Micro code in order to get it to do whatever it wants. And in this case it's as simple as saying openUrlInDefaultBrowser? and then URL= and then c:/windows/system32/calc.exe. Of course, calc.exe is sort of now the default classic app that you - it's always present in Windows. And that's the benign way of demonstrating to someone that you're able to run something on their computer because suddenly the calculator pops up, and they didn't start it, which means you did. And of course that's bad because you don't want people anywhere in the world to be able to run code on your machine.

And he notes in this log that you cannot read the response due to the same-origin policy, but that doesn't matter because the command is still executed. So then he also added, as sort of an afterthought, Trend Micro helpfully adds a self-signed HTTPS certificate for localhost to the trust store, so you don't need to click through any security errors.

**Leo:** So much easier, yeah.

**Steve:** Yeah, just, eh, we don't want to confuse people. We'll just make it transparent. So we'll put a self-signed HTTPS certificate, add that to the trust store. So then some back-and-forth ensues with screenshots between Tavis and someone at Trend Micro, and they send him an updated version, which he takes a look at. And so in the log, having looked at it, he posts: "Trend Micro sent me a build to verify they had fixed the problem. It looks like they're no longer using ShellExecute, so it fixes the immediate problem of trivial command execution." He continues: "I'm still concerned that this component exposes nearly 70 [seven zero] APIs to the Internet, most of which sound pretty scary."

So he says: "I tell them I'm not going through them, but that they need to hire a professional security consultant to audit it urgently." So then there's some more back-and-forth between Tavis and Trend Micro. And he responds: "Thanks, Jean. I ran this on top of a Trend Micro Maximum Security 10 installation." I guess Trend Micro Maximum Security 10 is the name of the product, which really then makes this extra sad. And Tavis continues: "And it looks like this fixes the most critical problem. Honestly, this thing still looks pretty fragile. I haven't looked through the dozens of other APIs you're exposing, and some just sound really bad. Look at some of these I noticed." And so he then enumerates some.

There's settings. There's settings/force. There's showCreateMasterPin, browserPasswordExport, getSessionKey, setProxyURL, clearSessionKeyData, exportBrowserPassword, emptyBrowserPassword, certPinningAddException, and then of course openUrlInDefaultBrowser, the first one that he found. And he says: "These are just the first few that jumped out at me as interesting from the list of about 70." He says: "I'm not planning to go through them all, but I would really suggest you get a professional audit of this."

So he posts then to the bug log, as he's continuing to look at this further, the fixed version. He says: "I happened to notice that the API/showSB endpoint will" - and "endpoint" is a term for RPC, technically they're called "RPC endpoints," basically a socket, or the actual code behind the socket - "showSB endpoint will spawn an ancient build of Chromium v41 with [the command line option] --disable-sandbox." He says: "To add insult to injury, they append (Secure Browser) to the UserAgent [string]." He says: "I sent an email saying, 'That is the most ridiculous thing I've ever seen.'"

So then he sends to Trend Micro: "I spent a few minutes trying to understand how the SB shell worked, and then realized they were just hiding the global objects. I sent this annoyed follow-up." And so he now he's beginning to get really sort of annoyed with Trend Micro because they're not getting, they're not understanding this. They're in typical CYA, do the minimum required, patch the one thing you show them is wrong, even though he's saying, "Guys, you have a complete architectural eff-up; and this entire thing is, like, seriously wrong."

So he starts: "This thing is ridiculous. WTF is this?" And then he gives them a link to a URL that demonstrates another way of spawning the calc.exe using this showSB. So they fixed the one he pointed to. He looked a little bit longer and said, okay, here's another one. How many more times do you want to do this? So he says: "You were just hiding the global objects and invoking a browser shell? And then calling it 'Secure Browser'?" And remember, he says, "The fact that you also run an old version with --disable-sandbox just adds insult to injury. I don't even know what to say," he writes. "How could you enable this thing by default on all your customer machines without getting an audit

from a competent security consultant? You need to come up with a plan for fixing this right now. Frankly, it also looks like you're exposing all the stored passwords to the Internet. But let's worry about that screw-up after you get the remote code executions under control. Please confirm you understand this report."

So the response from Trend Micro says: "Hi, Tavis. This is well noted." Whatever that means. "This is well noted. We have forwarded this information you have shared with our Product Team. Rest assured that this will be investigated thoroughly."

So Tavis then wrote and posts another working exploit. He says: "I noticed that there is a nice clean API" - unfortunately, of course, when he says "nice clean API," he means exposed to the Internet - "for accessing passwords stored in the password manager, so anyone can just read all the stored passwords." And then he gives them a demo link that does it. He says: "Users are prompted on installation to export their browser passwords." Now, by that he means whether or not they wish to export their browser passwords into the Trend Micro password manager.

He says: "But that's optional. I think an attacker can force it with" - then there's an API for that - "/exportBrowserPasswords API, so even that doesn't help. I sent an email pointing this out." He says: "In my opinion, you should temporarily disable this feature for users and apologize for the temporary disruption, then hire an external consultancy to audit the code. In my experience dealing with security vendors, users are quite forgiving of mistakes if vendors act quickly to protect them once informed of a problem. I think the worst thing you can do is leave users exposed while you clean this thing up. The choice is yours, of course." Trend Micro thanks Tavis for pointing these things out.

Anyway, the last thing is Tavis finally replies: "Thanks, Roy. I spent a few minutes looking into how passwords are stored if the user is using the password feature, or if they've exported all their browser passwords to Trend Micro. You're prompted to do that on installation; but it's optional, and you can decline. To be clear, you can get arbitrary code execution whether they're using it or not; but stealing all the passwords from a password manager remotely doesn't happen very often, so I want to document that."

**Leo:** Or the ability to do so. I'm sure it would happen every time, if you could do it.

**Steve:** Right. "This will get you all the encrypted passwords."

**Leo:** Good lord.

**Steve:** "For example, this will show the domain of the first encrypted password." Then he provides a link with basically, using this very powerful showSB API, he invokes JavaScript in order to run a JSON parser to pass the password data, which is exposed, just to show them. He says: "Then, you can use the decryptString API to decrypt all the strings, then post them somewhere else, meaning export them to anywhere else on the Internet that you would like in order to exfiltrate them from the current machine. So this means anyone on the Internet can steal all of your passwords completely silently as well as executive arbitrary code with zero user interaction. I really hope the gravity of this is clear to you because I'm astonished about this."

Then finally, again: "In my opinion you should temporarily disable this feature." Actually, all users should temporarily remove, actually permanently remove this horrible software.

**Leo:** Yeah.

**Steve:** Anyway, he says: "...disable this feature for users and apologize for the temporary disruption, then hire an external consultancy to audit the code. In my experience" - and anyway, he's just repeating himself there. He says: "Then the thread went public, and there was a bunch of back and forth" among various people who had continued to have fun. It turns out that one of the things they did was to enforce the same-origin policy, which does help this from being exploited by a remote website whose code has been running to issue this local HTTP request. However, people in the thread that then ensued note that there are ways around even the same-origin policy.

So what we're seeing is Trend Micro is not even providing robust fixes. They're desperate to hold on to basically a terminally irreparably broken design, just patching it where people find problems. This really bodes ill for it. So my recommendation upon seeing this is just anyone who has it should yank it immediately from their systems. It's just a hundred percent really bad code.

But Tavis is just very patiently saying, well, you know, you really should turn this off and get somebody, get a third party who, you know - because he recognizes that it's difficult for the company to fix this themselves. It's just you just cannot get the objectivity. They're no doubt committed on other projects. They're behind the eight ball and probably behind a deadline. They've got managers who have managers who don't understand what's going on, who just want to ship something or honor commitments that have just been made. Meanwhile, this comes in, and it's just, you know, it desperately has to get fixed. But they're trying not to have to do it. And so it's just an incredible example of doing it wrong, you know, squared. Wow.

**Leo:** Oh, well.

**Steve:** So we know that SHA-1 certs are no longer being issued in 2016. They may be living into 2016, but all CAs - and remember we went through the discussion, there was a thread that was launched in the so-called CA Browser Forum, the CAB Forum, where they were considering extending that because there were some companies that were saying, please, please, please, please, we really need to continue to issue SHA-1 certs in 2016. But it was never made clear why. Well, I think the next story may explain that.

But in this case, just as that thread had been floated, that was when that 80-round weakness in an internal component of SHA-1 was found, and that really scared everybody away from softening our determination that we really do need to start winding this down. So no SHA-1 certs issued in 2016. Except that Symantec did. In Symantec's own statement they said: "Symantec has identified a gap involving a limited use case on one of our platforms that allowed the issuance of five" - count them, five. That's not a lot, but still it's more than zero, which is what it should be. And they're a CA. They're a certificate authority. They're supposed to, like, you know, making mistakes is not okay. So they issued five SHA-1 certificates after the 31st of December, 2015.

Symantec says: "We have released a patch that addresses this issue and are in the process of notifying customers and revoking the affected certificates." The nature of the bug, though, is what's so fun here. They said under details: "Symantec maintains an enterprise portal in which customer administrators can approve or reject enrollments for certificates with domain and organization names that have already been pre-vetted by

Symantec, without requiring further Symantec review. We identified a gap in a specific use case where the code to block SHA-1 issuance after the 31st of December, 2015, was not in place, specifically" - this is, again, classic bug - "where the certificate was enrolled in 2015 but approved in 2016." So this was sort of - these certs were in limbo.

**Leo:** Oh, yeah, okay.

**Steve:** They had been technically kind of like preliminary issuance. It got partway through their system. And no doubt the approval process had the date check in it. I'm sorry, the enrollment process had the date check in it. But the approval didn't. So in these five cases the certificate - the total process of issuing a cert straddled New Year's so that they were approved in 2016 and had 2016 dates. So anyway, that was found, and they did the right thing. They fixed the bug. They revoked the certificates. And I'm sure, well, and they've told whoever issued them, sorry, you can't get SHA-1 anymore, so figure out how to deal with SHA-256.

Okay, but the next story is interesting because Firefox, in wanting to be as tight as possible, knowing that no certificates, no valid certificates, SHA-1 certificates, would be issued from 2016 on forever, in 43.0.3, that version of Firefox which was issued before 2016, it added code to reject SHA-1 certificates containing a not-valid-before date after 2016-01-01. And maybe they mean, yeah, yeah, 2016-01-01. Meaning that 2016-01-01, January 1st of this year, 12 days ago, was the first day that this certificate is valid. So, for example, those ill-issued Symantec certs would have had a "not valid before" in 2016.

Well, it turns out Firefox began having weird problems. People started reporting they couldn't connect. I mean, they were having problems. And they tracked it down. And, oh, boy, and this is another one of these - this is under the category of unintended consequences and side effects and further sort of demonstrates just how sprawling the technology has become and how difficult it is to give up things like IPv4 for IPv6, even though we know we're supposed to.

In this case, it turns out that there are man-in-the-middle platforms, like all of these corporate proxies, that are issuing - we know that they're minting certificates on the fly. Remember, the way they work is that they're a CA, with their CA that's been planted in the desktop and laptops of all of the corporate employees as part of the requirement for them to connect to the Internet. When they go to any website, this man in the middle intercepts that request, on the fly generates a certificate, signs it, and returns that as - basically it's impersonating the actual site remotely.

Well, these appliances, these man-in-the-middle HTTPS proxy for whatever reason, typically content control and AV filtering, they're still signing their fake certs with SHA-1. And so when Firefox said, with all good intentions, just never occurred to the people at Mozilla, they said, "We're not going to accept SHA-1 signed certs anymore because why should we? No valid CA is going to issue them." Except apparently, and now we know for sure, these appliances that have inserted themselves into users' connections and are doing this in order to crack the encryption and inspect the content, are signing with SHA-1. So browsers still need to tolerate this, at least on a local scope.

One of the Google groups had some commentary about this that just sort of followed up and basically repeats what I've said, so I won't go through it. But essentially what Mozilla had to do was immediately issue an update to Firefox so we are all now at 43.0.4, which removes this requirement that SHA-1 certs are only valid if they were issued before the beginning of this year. Now, what we'd like to see is the man-in-the-middle proxies not

be used, but that's not going to happen. So they'll have to be updated so that they're signing with SHA-256.

A short-term fix would be simply to fudge the not-before date to New Year's Eve of 2015. That way the certs would look like they had been signed at the end of last year. I mean, the whole thing is horrible and makes you pinch your nose anyway. So the fact that it's an invalid notBefore date seems sort of, you know, who cares because the whole idea of doing this is reprehensible. But it's the way the world is going to be moving forward. But anyway, sort of another example of why it's just hard to get this kind of forward progress in the industry.

And, almost predictably, it turns out that malware is being now hosted on malicious HTTPS websites.

**Leo:** Oh. I saw this.

**Steve:** Using the Let's Encrypt free certs.

**Leo:** That's too bad.

**Steve:** And, you know, it's...

**Leo:** Inevitable, though; right?

**Steve:** Yes. And it has nothing to do with Let's Encrypt. If we were ever going to have a means of encrypting for free, which we absolutely, I would say it's now proven, more than a quarter million Let's Encrypt certs have been issued. And I've received communications from many of our listeners who, for example, had a site that was using a self-signed certificate because, you know, which always gave a warning, and everybody who visited it got a warning, but at least they were able to get some encryption up, even if they didn't have a certificate authority to verify the site's identity. They've switched gleefully to Let's Encrypt. I mean, it really is that easy to do.

So, and this is really not, in my opinion, that big a story. What this means is that the content that wouldn't have been encrypted and so would have been more accessible to AV filtering catching it, is now able to run through an encrypted tunnel. I'm not saying that's a good thing. But as you said, Leo, it is absolutely inevitable that, if we're going to have an API-based system, notice that this is not - there was no defect found in Let's Encrypt. In the specific instance where the story was generated, some bad guys compromised a non-HTTPS website and added a domain and then installed Let's Encrypt, used the API, got the certificate for it, and then used that compromised website, with security now, to host their malware.

And what was interesting was some interesting back-and-forth about accountability. One of the things that has been always clear is that it is not a certificate authority's responsibility to police content. They're not content police. And there's been a very firm line drawn in the delineation of responsibility. They assert exactly and only what they say they are. In the case of Let's Encrypt or other domain validation certificates, they're only saying this is a certificate for this domain. Well, that's all it was in this case. Or, of

course, the CAs are still in business because in some cases you want additional levels of validation for certificates like extended validation certs, where you're actually saying this is a real - this corporate entity that has this website has been issued this certificate. So there's way more you can believe about them.

And this demonstrates it. Here we've got malware using now the free certificate ecosystem that's been created in order to run its stuff through an encrypted tunnel. Again, inevitable. But this is, again, this is also why the higher level of authentication of certificates is still useful because they're not using those because that would require much more vetting, and that's why they weren't using them before. But so I guess my sense is, yeah, it's unfortunate, but nobody who's been watching this, after we saw it, was that surprised. And it's going to happen. Okay, Leo.

**Leo:** I love it when you sigh like that. Okay what, Steve?

**Steve:** It's January 12th, 2016.

**Leo:** It is. I know that, yes.

**Steve:** Today, on this day.

**Leo:** Yes.

**Steve:** Windows XP Embedded SP3...

**Leo:** Oh, no.

**Steve:** ...Extended Support ended.

**Leo:** Oh, it's over. It's finally over. Our long national nightmare is over. So that was the trick you were using, right, to update Windows XP?

**Steve:** Yes.

**Leo:** Does that mean you're not going to get any more updates?

**Steve:** Yeah, correct. We were continuing to get them by adding a key in the registry that said we're an ATM or a credit card terminal or something. We're an embedded version of XP. That support continued until today.

**Leo:** You got an extra year; right? You got one more year out of XP.

**Steve:** Until today.

**Leo:** Aw.

**Steve:** So, yup.

**Leo:** Golly. Now what are you going to do?

**Steve:** Oh, I'm staying right where I am.

**Leo:** That's what I thought.

**Steve:** Yeah. I've got too much work to do. I would love to rebuild this system and build a new Win7 machine. I'm completely comfortable with Windows 7 because it's almost obsolete. So when I get caught up on projects.

**Leo:** Were there any, just out of curiosity, were there any updates that you got through the embedded...

**Steve:** Oh, yeah, yeah. Yeah, yeah, I have two laptops that are still on that.

**Leo:** But, I mean, were there some issues - in other words, I guess the real question is...

**Steve:** Oh, was anything important?

**Leo:** Yeah. Yeah.

**Steve:** No, no.

**Leo:** So even if you didn't update XP, there's no massive security flaw that you are now vulnerable to.

**Steve:** Nope.

**Leo:** That we know of, anyway. All right.

**Steve:** Yup. Correct. Correct. Now, the problem is, for example, well, actually Service Pack 3 does support SHA-256. But it is going to now become increasingly old. There will

be, well, for example, it doesn't support...

**Leo:** But, hey, so are you and I, so...

**Steve:** Yeah. I'll move to 7 before I die. Of that I'm sure. But that'll be my last move.

**Leo:** Never 10.

**Steve:** Never. No, no, no.

**Leo:** Never Windows 10.

**Steve:** I'm here to stay. I'm not doing flippy tiles. No, thank you.

**Leo:** No, thank you.

**Steve:** I got Paul Thurrott's icon in Twitter is now a flippy tile. I think he did that just to annoy me.

**Leo:** Just to drive you crazy.

**Steve:** Because, yeah. Okay. LastPass v4.0 Emergency Access. How can it be TNO? Everyone wanted to know. And this is beautiful. You'll get this immediately, Leo, because you're a big PGP person, or GPG, or, you know, that.

**Leo:** Yeah. I use GNU Privacy Guard.

**Steve:** Right. So it's just - it's elegant in its simplicity. And it's as simple as the fundamental RSA public key system. Everybody, all the players in emergency access generate themselves an RSA key pair, meaning that, well, actually LastPass users already have one, which is the way they're able to decrypt their stuff. But when they appoint other people to be their emergency access recovery contacts, each of those, any of those people they appoint generate an emergency recovery RSA key pair. By that I mean that, in their computers, both a public and a private key are generated. The private key never leaves their computer and their control. But the public key does. Through an API, that goes to LastPass. Or maybe it goes directly to the user. I didn't look down at the plumbing level because those are details, and I completely get it that Joe knows how to do this stuff right.

But the idea is that, when this emergency access enrollment occurs, the user, the main user who wants protection to enable controlled access to their vault, that their LastPass vault master key is encrypted using each of the emergency access contacts' public key. And LastPass holds that. So once the main user's vault key is encrypted under the

various emergency access contacts' public keys, the only way to decrypt it is with their individual private keys. So LastPass can securely escrow those blobs. And then we have whatever mechanism is put in place to then enable LastPass to supply those.

And so, for example, like the 48-hour notice where, under whatever terms and conditions, time goes by such that - and the user doesn't reply, then that enables the mechanism by which LastPass can supply those individually encrypted blobs to the individual emergency access contacts. They're then able to use their private key that they always kept to themselves. LastPass escrowing it got no benefit and couldn't ever do anything with it. All it could do is be the mediator and determine when the conditions had been met for emergency access.

So LastPass provides the blobs. They use their private keys to decrypt them. That gives them the decryption key for the person who wished to enable emergency access, making that person's vault available. So a perfect classic example of the way public key crypto works and how, basically in a three-party system, you can securely have an escrow agent that has a role to play, but in being an escrow agent doesn't gain any benefit. They're just, you know, there's no way they can. The only thing they could do would be to collude with some of the contacts to gain access. But assuming that the contacts are trusted, then there's no way for LastPass to unilaterally do anything with the blobs that have been encrypted using those contacts' own public keys.

**Leo:** I'm so glad, by the way, they did this. I mean, and I'm glad to hear that it's TNO.

**Steve:** Yes. And...

**Leo:** It's just a great feature, I think.

**Steve:** Yes. And you've been talking about it, about the need to have - and here we are, new year, 2016. Think about...

**Leo:** Your demise.

**Steve:** Yeah, unfortunately. Plan ahead: life insurance, medical insurance, and password insurance.

**Leo:** Yeah. No, that's really - this was a great feature. And I like the new LastPass 4. But as far as I could tell, that's really the new feature of LastPass 4. The rest is cosmetic. I'm just glad they...

**Steve:** So our friend Mark Russinovich released an update that I mentioned briefly, v2.4 of SigCheck. In his own little squib he says: "SigCheck is a command-line utility that shows file version number, timestamp information, and digital signature details, including certificate chains. It also includes an option to check a file's status on VirusTotal, a site that performs automated file scanning against over 40 antivirus engines, and an option to upload a file for scanning."

Then he says, sort of in his user notes, for example: "One way to use the tool is to check for unsigned files in your windows/system32 directories with this command." And then he gives the command, which is just - so you'd open a command prompt, and you'd say "sigcheck -u -e c:\windows\system32." What that does is that runs through every file and verifies the signature of every file. And he says: "You should investigate the purpose of any files that are not signed."

**Leo:** Boy, that's a great tool. That's nice.

**Steve:** Isn't that cool? Yes, I just think our listeners are going to love it. So you want to google SigCheck, S-I-G-C-H-E-C-K, and you'll get right to it. This is Mark who of course Microsoft hired from his Sysinternals, where we all loved the Sysinternals tools that he and his partner were creating. But the thing that put this on my radar, what he added - oh, and how cool is it also that, if there's a bunch of stuff that you want to just quickly check, it'll also traverse subdirectories. So it'll do a full multilevel directory traversal, and it will upload and verify things against VirusTotal. So it'll just automate that whole process. If someone wants to, like, run their My Documents directory, for example, through Virus Total, you can do that with one command.

But the thing that put this on my radar was, and what he added in 2.4, was that you can also use it to check for anything unexpected, well, or un-Microsoft, in your system CA root store using the -t, with options [u] and [v]. He says dump the contents of the specified certificate store, or you can use asterisks for all stores. And then, he says, you can specify -tu to query the user store, although the machine store is the default. And you can append a -v to have SigCheck - and "v" stands for verify. SigCheck will download the trusted Microsoft root certificate list and only output valid certificates not rooted to a certificate on that list, meaning that are not associated with the official Microsoft certificate list. And if you are offline, if for whatever reason right now at that time you cannot get to Microsoft's online site, then there is an authrootstl.cab file and an authroot.stl. Either of those will be used instead. So you can provide those locally.

Anyway, we've talked about tools for helping people discover unauthorized root certificates. Now we've got an official one from Mark, which downloads Microsoft's list on the fly and then performs a comparison against it. So that's, I think, that's the tool you want to use. And it does so many more things, and it's tiny because Mark writes good code.

Oh, a piece of errata. Last week Shawn tweeted me, I think he was the first person, although I found a bunch of this from our sharp-eared listeners in the mailbag. I mentioned that WiFi used CDMA. No, Steve. I actually said CDMA/CD. What I meant was CSMA. And so Shawn corrected me, saying it's actually CSMA/CD. But actually even that's not correct because it's not collision detection, it's collision avoidance. So what WiFi uses is CSMA, which is Collision Sense Multiple Access, with Collision Avoidance.

And remember I also explained how wired Ethernet uses collision detection, where it's listening as it's transmitting. And if it realizes that it just got garbled, that is, its own transmission got garbled, then that was collision detection. Well, wireless cannot do that because, as you know, Leo, from just having recently passed your ham license, there's no way that you could have a receiver on the same antenna as a transmitter that is, like, blasting out at however many watts of power it is. There's no way that a receiver could listen at the same time because that local transmission would completely swamp its ability to receive. So the WiFi guys realized this, and they use an algorithm to assign

essentially timeslots to devices and thereby avoid collisions rather than try to detect them.

And a little bit of miscellany. We are, I think it's four, or maybe it's five episodes now into the new series "The Expanse."

**Leo:** You know, I just started watching it. You think it's good?

**Steve:** It is a huge win.

**Leo:** Oh, good.

**Steve:** I wanted to wait until I could see where it was going to go because the very first scene was a little muddy, and I was a little weird. And there is a lot of...

**Leo:** Yeah. That's when I, by the way, that's when I stopped watching it.

**Steve:** Yes.

**Leo:** Lisa said, "I don't like this." And I said, "Okay, well, I'll watch it another time."

**Steve:** Yeah. And there is some - there's, like, three - there's Earth and the Martians, as they're called, only they're not aliens, they're just people who moved to Mars. And then there's the Belters out on Ceres. And so, as happens, because each group has different needs due to their local ecosystem, politics have become strained. And so there is sort of a bunch of, like, background politics going on. And I have no idea what that is. When I was reading the book, I could follow it because you had people's names, and they kept getting repeated, and got careful dialogue. Here it's just blah blah blah blah, and people kind of talking under their breath, and something hovers by and goes past the camera and you get distracted.

So don't worry if you're not tracking all of the political machinations between the Earthers and the Martians and the Belters because it sort of doesn't matter. I mean, it's just tension. But, oh, the effects are good. The production quality is good. I've liked a lot of like the last three weeks. And so I want to give that a recommendation. I thought it was really good.

**Leo:** Oh, good. I'm going to watch it, then.

**Steve:** Yeah, I think you should. Go at least five in and see if you agree. I think you'll get hooked by then. I can't wait. It's tonight, and it's like, oh ho, what's going to happen next? Actually, I know because I read the books. But still I want to see it.

Then there was another new Syfy show. And I just wanted to mention it. It's not started yet. But it is available for download, or somehow accessible streaming. And that's a

series called "The Magicians." Not sci-fi. And it's a little bubblegum-y. But I liked the first episode. I actually watched the first episode twice. So I wanted to let people know, just give you a heads up. You might tag it in your DVR, check out the first episode. Again, something's going on with Syfy because they seem to be, you know, they've really upped their game in terms of production quality. It's gone from "Gag me" to, "Wow, this is on Syfy?" So "The Magicians" looked like it may be fun.

**Leo:** Good.

**Steve:** And Leo, without any spoilers, I will just say that I've seen "Star Wars" twice now.

**Leo:** Oh, you liked it.

**Steve:** I really - it was just fun. And I know without even - just because we have a mind meld, I know what problem you had with it. And it's the problem other people have had with it, and I did, too. I found myself thinking that...

**Leo:** But you know, what I realized is that was the right thing. That was a smart thing for Disney and J.J. to do.

**Steve:** I think so, too. I think so, too.

**Leo:** Now we're real - this is starting to sound like "The Expanse." And another thing. No, I think it was - I understand why they did it.

**Steve:** Yup, yup.

**Leo:** And you know what, they have next year...

**Steve:** But it annoyed me while I was watching it.

**Leo:** ...and the year after and the year after and the year after. So there's much more to come.

**Steve:** Yes. While I was watching it, I was annoyed. I thought, wait, wait. And I have to say I really - I went with my best friend. We both enjoyed it the second time, I would say more.

**Leo:** Yeah, I want to - I haven't seen it a second time. I think I might.

**Steve:** Because we knew what was going to happen, what to look for. I just...

**Leo:** And your expectations had been, you know, you were no longer saying, oh, this is going to be a reboot. You kind of got, okay, I get it. I know what's happening here.

**Steve:** Yes.

**Leo:** Which is good. You know what, it really - what it is, it's very satisfying for Stars Wars fans, and that's really the bottom line. Right?

**Steve:** Oh, and we get two more; right?

**Leo:** Yeah. More. Two more? You're going to get one a year for the rest of your life.

**Steve:** Oh, you promise?

**Leo:** No, Disney has already promised.

**Steve:** I guess, given the money, how could you say no?

**Leo:** I mean, it made more than a billion dollars. They only paid four billion for this franchise.

**Steve:** Now, here's a question. I watched, as you did on Sunday, the Golden Globes. And of course "Star Wars" was nowhere mentioned.

**Leo:** Which is weird.

**Steve:** Do you think there will be any mention on the Oscars?

**Leo:** Never has been. Never has been. Don't know why.

**Steve:** No? Well, it just...

**Leo:** Not for this, for sure. Had this broken new ground in some interesting way, maybe.

**Steve:** Right. It's just a pure cash-generating engine.

**Leo:** Yeah. It's great, though. It's great. And it makes you feel good.

**Steve:** They got my money twice.

**Leo:** Yeah. It's not - you don't - I think that you channeled me exactly right, and I channeled you back, and I think we are on the same wavelength about it.

**Steve:** Yup. Yup.

**Leo:** But I understand why, and I think it's fine, and it's made everybody happy.

**Steve:** I do, too. I forgive it. And think about how the whole new generation who didn't see Episode IV, and first of all wondered, wait, IV? What happened to I, II, and III? Unfortunately, we found out later.

**Leo:** Well, when we saw IV...

**Steve:** Yes.

**Leo:** ...they didn't say Episode - or did it? No, it didn't.

**Steve:** Yes. Yes.

**Leo:** No, wait a minute. Now, see, I thought so, in my memory. But then I've been reading about the original. I don't think it said Episode IV when it first...

**Steve:** I just watched it.

**Leo:** It does now.

**Steve:** Oh. And they also added a bunch of new little critters running around.

**Leo:** Yeah, that was part - I think that was part of Lucas's revisions.

**Steve:** I - but what I do remember was...

**Leo:** I could have sworn it was, too.

**Steve:** Yeah. I'm absolutely certain that we knew, like, that he did this on purpose, that he dropped us into the middle. And lord knows why. But he did. And I'm sure we knew that then.

**Leo:** Chatroom is saying the Episode IV was added in the 1978 re-release. And that was my understanding, as well.

**Steve:** Okay, well, '78.

**Leo:** It came out in '77. And so what I'm saying is when we first saw it, we'd been waiting in line a long time, we were young, young guys, and we saw it, it didn't - there was no Episode IV. That came later.

**Steve:** But only a year later.

**Leo:** Yeah, not much later.

**Steve:** So that was to clarify his mistake, essentially.

**Leo:** Yeah. He added more over the years. But, you know, we interviewed the guy who's doing the specialized edition on The New Screen Savers. This is his attempt to get back to that original 1977 version. It was really - it's really interesting what he's doing.

**Steve:** Oh, cool.

**Leo:** Yeah, really interesting.

**Steve:** So I've said it before. I've already seen tweets from listeners who have thanked me for pushing people to go watch "Homeland." We just finished the fifth season. It was really good. But I also wanted to mention that there was a series that is just starting that also looks really good called "Billions." And it stars the actor, Damian Lewis, who...

**Leo:** Love him. From "Homeland."

**Steve:** Yes, who was in the first two seasons of "Homeland" playing the character Brody. And so it's Damian Lewis and Paul, how do you pronounce his name, Giamatti or something like that?

**Leo:** Giamatti, Giamatti.

**Steve:** Yes. Those two actors and the producer and director were all interviewed by - I'm blanking on his name.

**Leo:** I'm trying to channel your thoughts.

**Steve:** Charlie.

**Leo:** Rose.

**Steve:** Charlie Rose. Thank you.

**Leo:** This is sad. We're like an old couple now at this point.

**Steve:** I strongly recommend that you DVR Charlie Rose, just stick it in there.

**Leo:** I will, yeah.

**Steve:** Sometimes they're not interesting. But he has fabulous interviews.

**Leo:** I do, I TiVo it.

**Steve:** Good. Well, anyway...

**Leo:** The only thing that makes me mad about Charlie Rose is he doesn't say at the beginning of the show who's on. So I have to...

**Steve:** No, he does.

**Leo:** Well, no, no, there's a long prologue, and then he finally gets around to it. It's a little annoying because you can't, like, just look in the thing and say, oh, I don't want to watch that one; oh, I do want to watch that one.

**Steve:** Oh, you do have to watch the first few minutes.

**Leo:** You have to watch the beginning of the show each time.

**Steve:** That's - yes, yes. So I wanted to say that "Billions" is available on demand.

**Leo:** "Billions."

**Steve:** For me, Cox provided it. I watched it and loved it. It's a little adult, I mean, this is not for the kiddies. You'll know why in the first minute. But it looks really good. It's the story of Axe Industries' multibillionaire who is skating over the edge. And the U.S. Attorney decides he's going to prosecute. But lots of complexity. Again, I think it's another great Showtime series.

**Leo:** Good, I'll have to watch it. All right, good. Yeah, good.

**Steve:** And in the theater I saw "The Big Short."

**Leo:** Ah.

**Steve:** I hated it.

**Leo:** You hated it.

**Steve:** I hated it.

**Leo:** Everybody's raving about this.

**Steve:** Okay. Yes. The problem is you don't want a bimbo in a bubble bath holding a glass of champagne to explain what collateralized debt obligations are.

**Leo:** Yeah, probably not, okay. But that was to make it palatable.

**Steve:** Well, now, Jenny and her mom loved it. And for the first time ever, they say they understood what happened...

**Leo:** There you go. There you go, yeah, right.

**Steve:** ...in the whole housing collapse. I actually already knew, and had read several books and seen all the movies about it.

**Leo:** Right.

**Steve:** So it just wasn't my style. So all I'm saying to anyone who wants to save themselves the cost of a movie ticket, if a sort of a whimsical, I mean, there were good

parts to it. But, eh, I wasn't a big fan.

**Leo:** Looked a little like "The Wolf of Wall Street." I felt like they were remaking "The Wolf of Wall Street" sort of. But I haven't seen it yet, so.

**Steve:** Yeah, but it did - it also had, like, cartoon bubbles popping out of people.

**Leo:** Oh, that, yeah.

**Steve:** And it was that. It was the style. Stylistically it just - on one hand, it wasn't taking itself too seriously. But I think they were trying to pop it up in order to...

**Leo:** Right, make it fun.

**Steve:** ...to make it like [crosstalk].

**Leo:** Michael Lewis's books are great. And if you haven't read that or any of his other books, he's really a wonderful writer.

**Steve:** Right. And last piece is that I was an early contributor to the Dash earphones.

**Leo:** The Bragi. Did you get yours?

**Steve:** The Bragis. No. But they - I got email on the first day of CES, when they were all excited about going, saying that the first 5,000 were - that they had shipped, and they would be shipping them out through the course of CES and shortly after. So anyway, I'm hopeful. They do look nice. I paid a hundred dollars less than they want now as an early supporter.

**Leo:** You lucked out because they're expensive.

**Steve:** Mine were 200 bucks.

**Leo:** Yeah, they're expensive. Well, good. Let me know.

**Steve:** And I do like the idea of them not needing a phone, that they've got 4GB of music storage. And so you just stick them in your ears and tap them, and they work. So...

**Leo:** Hey, I have an update for you.

**Steve:** Oh.

**Leo:** Because we talked last week, and I said I had ordered it, but I didn't get it in time for the cruise.

**Steve:** Oh, yes, the hardware firewall.

**Leo:** The Tiny Hardware Firewall. And I did do a review, a full review, if you want to see it, on The New Screen Savers last week.

**Steve:** I watched it, yup.

**Leo:** This was the device that I got for free by buying a $91 year's subscription to tiny HotSpotVPN, which is one of, probably one of the oldest VPNs, hotspot VPNs out there, from WiFiConsulting. They've been doing this for many years, and I've recommended them for years. I can give you some more technical updates because I didn't mention on the review, for instance, I queried HotSpotVPN about logging. You know, the negative on this, and one of the negatives in the review that I said was you can't choose your server. You can't choose your country. So for avoiding geographic restrictions it's not ideal. But this is really a security and an anonymity device. They do log, but they log for 48 hours only, and they overwrite the log every 48 hours. They don't log any user interactions, they say, but they do, they say, for net sec and support reasons they log for 48 hours. So just something to keep in mind. We're pro no logging for a VPN solution.

**Steve:** Yeah.

**Leo:** Now, how does it work? Really great. I mean, it's been on now, it's booted up, so I have it as a hotspot in my settings here on my Windows machine.

**Steve:** Powered by your USB key.

**Leo:** Yeah. Right now it's powered by USB. But I was able to run it for, how many hours? I think two or three days off of a small smartphone recharger battery. Once you run it, oh, I guess I've never run it on this machine, so I'd have to - I won't do it right now. You can log in. You can turn on the VPN. You can turn on the Tor browser. It also does malware and adblocking built into the VPN through a proxy. Can't use that and Tor, as well. You have to choose your proxy. But it's fast. It's fairly fast, which is nice because a lot of times VPNs and Tor slow things down. It's usably quick and very convenient. Once you set it up on any given computer, I just carry it around in my backpack - my man purse, okay, let's be honest - and I have it any

time I want to open a WiFi access spot. And I feel like this is a really good solution for people who just want to be secure on an open WiFi access spot. I like it a lot.

**Steve:** Right. Very cool.

**Leo:** So that is a positive review. For the full-length review you can go to our YouTube channel, YouTube.com/twit, or watch The New Screen Savers on Saturday because I gave a much more thorough review of that. All right. I've got questions. You've got answers?

**Steve:** But I've got one nice fun...

**Leo:** A SpinRite?

**Steve:** A fun SpinRite story to share.

**Leo:** Do a SpinRite.

**Steve:** This is from - I found this at the very end of my mailbag, so he just sent it in, Mark Clark. The subject caught my eye, of course. He said: "SpinRite helped a retired vet and saved my wife's sanity." He's in Portland, Oregon.

He said: "Dear Steve and Leo, blah blah blah. My wife runs a nonprofit housing company, and one of the tenants tends to visit the office when he has nothing better to do. His computer was not working, so he was spending a lot of time in the office. I happened to be in the office setting up a new computer, and my wife asked me if there was a computer in the office she could give him."

**Leo:** To get him off her back.

**Steve:** Get him out of my hair. "I asked him what was wrong," wrote Mark, "and he said his screen was red, and Windows would not boot. So I told him to bring his computer" - like I guess he was upstairs - "bring the computer down to the office, and I would look at it. I got out my trusted copy of SpinRite and let it go to work. I did notice the screen was not getting all the colors, but I figured it was because I didn't have the video plug in all the way. SpinRite ran and had about 8 million ECC errors and 600 seek errors." But of course that's normal on new drives. He says, "But otherwise it didn't complain. Once SpinRite was finished, Windows booted up just fine, and I let him know he could have his computer back. It was working again. He took it back to his apartment; and when he tried to connect the video, the port fell completely off the motherboard."

**Leo:** That might explain the color problem.

**Steve:** That could explain the color problem.

**Leo:** SpinRite did not fix that.

**Steve:** "I told him to give it back to me, and I would fix it. I went to the computer store, picked up an inexpensive video card."

**Leo:** Smart man.

**Steve:** Yeah. "After installing" - this guy really wants him not bugging his wife. "After installing the video card, I reran SpinRite on Level 4 just to make sure. Now he has his computer back, which means he's not spending time in the office annoying the staff, and saving my wife's sanity. Thank you for a great product. Mark Clark."

**Leo:** Love it.

**Steve:** Mark, thanks for sharing your story with us.

**Leo:** Yeah. SpinRite as yet cannot save you from disconnected hardware ports. But at least the hard drive's working. That must...

**Steve:** And it runs even if the screen is showing all red.

**Leo:** Yeah, yeah. That must have been some banged up computer there. Steve Gibson, you should answer these questions from our fine viewers, questions we should say, just in the interest of full disclosure, Steve has carefully selected, but more to the idea to having the answer at his fingertips, not having to do any research or anything like that.

**Steve:** We got some really good ones this time.

**Leo:** As always. The best audience ever. Charles Anderson kicks things off from Columbus, Ohio. He wonders whether hard disk drives read by sectors or by tracks: Steve, I've been listening to the Security Now! podcast since the beginning. And that is a long time ago. I purchased SpinRite to fix my TiVo, and it worked. But every time I hear you talk about reading and writing sectors, I'm reminded of something I read years ago somewhere: "Modern drives no longer read and write individual sectors." Since they're not constrained by internal RAM as in the old days, when they want a sector, they just read the whole track into onboard RAM and return the part of the track that represents the data requested. That kind of makes sense. And then, if a request comes in for the next sector, well, they already have it locally in the cache. And when they write a sector they read the track, replace the data in the stream that is the sector, and write the whole thing back to the platter.

So, given that, do modern drives operate on tracks and just return the sectors requested? And, if so, how do your repeated requests for a specific sector over and over during data recovery relate to this? How is it impacted? Charles Anderson, Columbus, Ohio.

**Steve:** So it turns out that the truth is somewhere in between that. So a simpleminded description, which is what he remembers, is that they no longer read and write individual sectors. That's not true. But what happened was, in the evolution of drives, we went from a controller that had all the brains, and where the drive was really just dumb. There was no brain at all, no processor to speak of. Basically the controller told the heads what cylinder to go to and then selected which head to read from. And then the data streamed in, and the controller looked at the data and found the sector or sectors that it wanted. So basically the controller placed the head where it needed to, activated the proper head for which surface the data was either to be read or written to or from, and then just had a real-time transaction as the data moved under the head.

Everything changed when we went to so-called IDE (Integrated Drive Electronics) drives because now there was a processor in the drive. And over time drives got increasingly clever. They got more and more RAM. Of course, operating systems got much busier, too, so that lots is happening all the time. So if you think about it, if the challenge is I'm being asked to read a certain sector on a certain track, and it might be that, I mean, it's likely that other sectors may be asked for on the same track, then what the drive does is, when it arrives at the track, because it's got ample RAM, and it's smart, it starts collecting the sectors that are passing under it while it's waiting to get to the sector that's been requested because the chances are great, especially now that we've got so many sectors per track, that it's not going to just arrive at the track as the sector starts to pass underneath. It's going to have to wait a while.

Well, since it's not doing anything else, and it's on that track waiting for the sector that's been requested, it sucks all the ones in until it gets to the one that's been asked for. Then it starts reading - and typically requests are multisector requests. So it'll start at that sector and then send all of the other sectors in as it encounters them. And at some point it may lap itself and get back around to the sectors that it was already reading before it got to the one it was asked for, in which case it'll send those in at a much higher speed.

So I guess the way to think about it is that you've got this physical system with moving heads and spinning platters, with as much intelligence as is useful, so, like, massive intelligence. What would you do to make this thing always be as busy guessing ahead, reading ahead, getting data that it thinks you might want. And that's what modern drives are doing. Essentially they're just doing everything. And as a consequence, they're really fast.

**Leo:** But you, as SpinRite, still have the capability of going sector by sector, no problem.

**Steve:** Ah, yes, thank you, I forgot. The second part of his question was what happens. All of this comes to a grinding halt when a sector cannot be read. If it can't read the sector, if it's not a sector that's been requested, if it was one that it was opportunistically reading, it just shrugs. It doesn't care. It's not going to raise an error because it's not an error in response to a request you made. So there's no way it can complain if it could not

read an opportunistically read sector. But if you do end up coming around and requesting that sector, then it will say, oh, sorry, I haven't got that for you.

Now, if you ask for it again, it'll try again. And there are maintenance modes that SpinRite drops the drive into. For example, I can turn off, and I do, all of that read-ahead activity because I don't want that messing around with recovery. So there are things you can do that are not the normal way the drive works to sort of retard them a bit in how aggressive they're being. And that gives SpinRite much more accurate one-to-one control over what it's doing.

**Leo:** Nice. Perfect.

**Steve:** Cool.

**Leo:** Thank you. Brendan Sherwin tweets about PRNGs: Steve, I was - prng, prong. Shall we call them prongs? Maybe prong. I was hoping you could discuss how a pseudorandom number generator, or PRNG, works, and how a hardware PRNG might be more random than a software PRNG. I think prong. I'm going to call them a "prong."

**Steve:** Okay.

**Leo:** How is a hardware prong more random...

**Steve:** No one will know what we're talking about with the prong.

**Leo:** Yeah, I know, be very confusing. How is a hardware prong more random than a software prong? You know what, we should start a campaign, I'm telling you. This is the way to do it.

**Steve:** And if there was more opportunity to say "prong"...

**Leo:** Why not?

**Steve:** For example, it's too bad that there's no way to pronounce HTTP.

**Leo:** Yeah.

**Steve:** That doesn't really work. Prong, yeah. But unfortunately it's a low use case.

**Leo:** Yeah. PRNG, though, that's a lot of syllables.

**Steve:** Yeah.

**Leo:** He says: I don't know if a hardware PRNG is more random. However, it seems to be the rule of thumb with the people I've talked with. They don't seem to have answers as to why. So maybe you do. Thank you. Love the show. Brendan. That was a long tweet. Is he using that new 10,000-character tweet capability?

**Steve:** Oh, I'm sure. You wouldn't believe. I get books now sent to me.

**Leo:** Oh, lord. And you thought that was a good idea.

**Steve:** Well, I'll tell you, it beats having a book divided up into 144 characters.

**Leo:** That's true.

**Steve:** Because that has happened also.

**Leo:** I'll grant you that, yeah.

**Steve:** And that's really not good. It's like, what? I also get, of course, I have noises for everything. So all kinds of - it's like [crosstalk].

**Leo:** [Making noises] And it goes one, two, except it's backwards, isn't it, 443, 442.

**Steve:** Yes. Not good. So, okay. Here's the deal. What is special about a hardware PRNG, or prong...

**Leo:** You see?

**Steve:** I agree, Leo, it's catchy. What's special about a hardware prong...

**Leo:** Oh, you can't do it, can't do it.

**Steve:** No, I can't do that - PRNG is it uses something truly random. It could be, in some cases, for example - this is crazy. Well, okay. A classic that I loved was that Sun Microsystems used to have cameras pointing at lava lamps, looking at the wax moving around, undulating in the lava lamp, because that's a chaotic process. And they would digitize that and use it to maybe not directly generate random numbers, but to seed a software PRNG.

There are pure hardware random number generators which use some physical property.

That's the key. You can't just use math, or then you have a software pseudorandom number generator. So it's actually not the case. You wouldn't say "hardware pseudorandom number generator," you would say "hardware random number generator," and that would be a HRNG, an H-R-N-G. So, for example, there are a number of ways that electronics can be used. Diodes are known to be noisy. That is, when a diode is trying to resist current, it only allows current to pass in one way. When you push current in the other way, every so often an electron will wander across the PN junction and cause an event.

And so there are ways that you can use a reverse-bias diode to generate absolutely unpredictable noise. It's called "quantum noise" because this is quantum phenomena. And that's typically - it needs to be post-processed. For example, it might be mostly ones and zeros, and some zeros. So you need to whiten it in order to balance it. So, for example, there is a bias in individuals. But it turns out, if you XOR a pair, that automatically removes the bias, which is a really cool side effect of XOR in this case. So a software pseudorandom number generator, given a known state, always produces a known and predictable, all the way out into the future, sequence of pseudorandom numbers. If you don't know what the internal state is, then unless you get a large number of its output, it's difficult to reverse-engineer its internal state.

So what we see now is sort of hybrid designs, where real-world things, maybe not a reversed-bias diode junction, but how about the unknowable, unpredictable, precise timing of packets arriving at a network interface? Or packets coming over from a WiFi source? Or the number of branches not taken or mispredicted by the CPU. It's completely, technically, that's deterministic because it's all algorithmic based. But it basically condenses so many different, unknowable factors about the past of the processor since it was booted and everything that's happened to it since, mouse movements and packet arrivals and everything else, even the phasing of the display interrupts coming into the chip. So that all could be used to seed a software-based pseudorandom number generator.

The reason is that sometimes these hardware events are low bandwidth. You can't ask the hardware for a huge amount of randomness. It's just not producing it that fast. So what you do is you use a pool concept, where over time a pool of true entropy is slowly generated. And then, once it's full enough, you start using it, and then you start filling, while you're using that pool, you are filling a new pool, again slowly, trickle filling it with hardware event true entropy. Meanwhile, a software algorithm is using the entropy in the pool you've just finished filling.

The problem is you don't want to use it forever because, in theory, anybody who knew what the algorithm was and looked at enough output could eventually reverse-engineer the state of the pool and from then on predict the future. So way before that much output has been generated, hopefully your secondary pool has now filled up enough, and again you've swapped pools. So that's a means of having true, effective, sort of constantly reseeded - this process of swapping pools is called "seeding" the software pseudorandom number generator. And so this is a periodic hardware reseeding of a software pseudorandom number generator. And now you know way more than you ever wanted to about prongs.

**Leo:** Or prangs. We didn't consider prangs.

**Steve:** Prangs, or hrongs.

**Leo:** I think random number generator is actually really the best.

**Steve:** I think maybe we ought to just be clear, yeah.

**Leo:** We don't have to - we can leave out the "pseudo."

**Steve:** Yeah, because in the case of hardware it's true random.

**Leo:** Yeah, hardware's random. That's the biggest difference, isn't it. It's not pseudo.

**Steve:** Oh, I forgot hard drive access timing, too. Anything you look at with sufficient resolution is generating at least some least significant bits of absolute uncertainty, based on the timing of the request and the rotational position of the drive. Boy, hard drives are a great source.

**Leo:** How good is it when they say, okay, move your mouse, wiggle your finger or whatever, those kinds of things? It's pretty random; right?

**Steve:** I don't have that in SQRL because it always upset me that sort of the user had input. It's like, okay, how long should I do this? And if I don't do it long enough, you know, and it turns...

**Leo:** Did I hit random?

**Steve:** Yeah, exactly. And it also seemed sort of more like a pacifier, like it's very much like McAfee saying that he's using military-grade encryption. Yes, but it's John McAfee.

**Leo:** Right.

**Steve:** So I don't care whose military's encryption you're using, if John McAfee was involved. Similarly, it's like, okay, if I'm in charge of making sure my hard drive encryption is secure by moving my mouse around a lot, that just doesn't feel right to me. And it turns out it's totally unnecessary. If you are operating, as I am, at the machine level with SQRL, as I just enumerated, a bunch of sources of entropy, and we did that famous podcast years ago, couple years ago, called "Harvesting Entropy" [SN-456]...

**Leo:** Yeah, that was a good episode.

**Steve:** ...where I talked about my design of SQRL's basically hardware true random number solution.

**Leo:** And that's where we first talked about this, using a diode as a way to generate...

**Steve:** As a hardware noise source, yeah.

**Leo:** Yeah, yeah. Manuel Othenos in Santander had an idea for upgrading HTTP or HTTP [trying to pronounce it as a word]: Great show, guys. Been listening for ages. When websites upgrade to HTTPS, they get the benefit of secure connection and encryption, which is often used for logons. Some sites don't want to use HTTPS across the board due to perceived processing overheads for the encryption. Or, I might add, the cost and complexity of adding this, which is the only, you know, we don't have logins to our system. So we felt like, well, there's nothing we're securing. You just come to, you know, a lot of websites, you go to the website, you read something, and you leave. There's no personal credential. So do you need HTTPS? And, well, we decided to do it anyway.

But here's my suggestion: Why don't the powers that be - I wonder who those are? - implement an upgrade to HTTP so the connection setup process is secure, but the encryption isn't used when it's not necessary? It ought to be difficult for that connection to be interfered with by attackers. And the lack of encryption should keep the NSA happy. Huh?

**Steve:** Okay. So this was sort of interesting to me, only I guess from a geeky standpoint, because that's been done already.

**Leo:** Oh.

**Steve:** It turns out that, in the enumeration of existing SSL and, well, SSL, I'm not sure if it's even still in TLS, it's certainly been, I mean, no one uses it. But SSL, remember that the way that SSL works is that there are a set of suites of cryptographic modes where you say this is the cipher I want to use. This is the message authentication technology I want to use. This is the way I want to do key agreement establishment. And you sort of mix and match one from Column A, one from Column B, and one from Column C in all kinds of different ways.

It turns out that a valid cipher - and this is something we'd never talked about, which is why I thought it was a fun question to bring up - a valid cipher is null. There are, exactly for this purpose, as Manuel sort of posits, there are SSL cipher suites where the cipher is null. So you would get authentication using a certificate. You would even get authentication that the message had not been tampered with. So it's tamperproof, and it's - I'm only thinking of the word "authentication." That's not what I mean. When the server, you're verifying the server identity, it's...

**Leo:** Uh-oh. I'm losing the - I can't channel you. I'm losing the signal. I don't know what you're talking about. Don't know where you're going with this.

**Steve:** [Laughing] Anyway.

**Leo:** Verification? Authentication?

**Steve:** Had too much coffee. So the...

**Leo:** Authorization. No.

**Steve:** Uh, no. I guess you're authenticating the remote identity of the server, but that doesn't seem like the right word. Anyway, it'll probably come to me. But so the point is this is already built in. And of course what it - so it prevents tampering. You know that you're connecting to who you believe you are. No one can mess with it. Yet there is no encryption going on. There's no encipher...

**Leo:** Certification?

**Steve:** No. No. Maybe it's authentication. I'm just not...

**Leo:** It is authentication. I mean, that seems like the right word.

**Steve:** Yeah. So anyway...

**Leo:** Validation?

**Steve:** No. Where you know the identity, absolutely know the identity of the other party. Authentication? That doesn't seem like authentication.

**Leo:** [Muttering] Don't know.

**Steve:** Question number four.

**Leo:** Affirmation? Chatroom's coming up with every -ation ever invented. We'll come up with one. Todd Warner in Hookstown, PA wonders what the hell does ShieldsUP! actually test? No, he didn't say it like that, I did. He said: Listener since day one, and several issues multiple times. I don't recall this being addressed: What does ShieldsUP! really test? I ask because every computer in my house failed to achieve the TruStealth rating. My router was enabled for one address to be a server in a DMZ, but the address was not in use. I disabled the DMZ, and all computers became TruStealth. Is ShieldsUP! just testing my router, or is it testing the actual computers? My router is a Cisco Small Business RV220W. What gives?

**Steve:** So I know this is sort of, for some of our listeners, they're like, okay, we all know this. But there will be some people who don't. And this is an important characteristic of a

router, which is why everyone needs one. Even if you only have one computer in your home, you need to have a router that routes to it. The reason is that the router is the entire public presence of your home's network, that one device. Your home, until we move to IPv6, has one IP issued to it by your ISP. So if you have one IP, then there's only one IP for ShieldsUP! to test. And ShieldsUP! tests that one IP. And that's the IP of your router.

So when you had a DMZ set that was apparently going to a computer that existed, even though there was no server running, it was responding to a ping. Thus the TruStealth failed because your entire network was not hidden because one computer was responding to a ping that the router forwarded to it. But when you turned off the DMZ, now that incoming ping from ShieldsUP!, among the many things it's looking for, trying to get a response from your system, now it was just ignored, just got dropped by the router.

So ShieldsUP! cannot test your machines because they're using private IP addresses, these things, 192.168. It's not possible for me for me to send packets out from GRC with 192.168. There's nowhere for them to go because that range of IPs has been set aside for everybody to reuse inside each of their own homes. So your neighbor has a router, 192.168.0.stuff, and you have one, 192.168.0.stuff. So you're using the same IPs inside your home, but one public IP, which is what the world sees your network as. And actually that was kind of good, so I'm glad I talked about it.

**Leo:** Yeah. Yeah, that's why we have routers. Otherwise...

**Steve:** Just for clarification.

**Leo:** ...you couldn't have multiple computers on the same IP address.

**Steve:** Correct. And they make great hardware firewalls.

**Leo:** And just by chance, I actually talked about that on the radio show this weekend. It's funny.

**Steve:** Good.

**Leo:** Ironic. I'm going to botch this name, Egbert Douwes. I know that's not how you say it. It's probably pronounced Vincent van Gogh. I don't know. He's in Amsterdam, The Netherlands, poses an interesting question about encryption: Steve and Leo, I really enjoy Security Now!, have a question for you. AES-256 can have a key length of 256 bits, but it always has a cipher block length of 128 bits. Does that mean that multiple keys will give the same encrypted output?

**Steve:** Okay. This is a great question.

**Leo:** Wow. You'd better explain the question before you give us the answer.

**Steve:** Yes. So here's what he's thinking. He says, okay, the key has 256 bits. So we know that there are 2^256 possible keys, right, 2^256 possible keys because that's how many combinations there are of 256 bits where each bit is either a zero or a one, binary, so that's 2^256. So that is a number that has 78 decimal digits. That's how many AES-256 keys, possible keys there are. It's a number with 78 decimal digits, so a lot. But now he's saying, okay, but if the cipher block length, the number of bits you put in at a time which this thing encrypts only has 128 bits, are there more keys than there are possible encipherings? Okay, which is a neat question.

So now we have to answer the question, how many possible encipherings are there? We know how many possible keys there are, 2^256, which is a decimal number with 78 decimal digits. How many possible encipherings? So what's an enciphering? An enciphering is a one-to-one mapping between an input, a set of input bits, and a set of output bits, so that every possible combination of input bits creates some other - probably other, not necessarily, could go right through, but seems unlikely, but it might - of output.

So how many possible ways are there to map all 128 input bits to 128 output bits? And we know the answer because let's take the first input, all zeroes. Now, that's going to map to one of 2^128 outputs. So it's going to encrypt, if we put zeroes in, it's going to encrypt to some one of 2^128 possible outputs. Now we turn the first bit on. That's going to encrypt to - it can't encrypt to the same thing we just encrypted to, so it's got to encrypt to one of 2^128 minus one. And the next one, like the input value two, that's going to encrypt - it can't encrypt to either of the first two, so it has to encrypt to one of the remaining, 2^128 minus two. In other words, factorial.

What we have is the total number of possible encipherings is 2^128 factorial, meaning 2^128, times 2^128 minus one, times 2^128 minus two, times 2^128 minus three, all the way down to essentially zero, or to one. And you're multiplying all that because that's, as we know from combinatorial math, that's how many possible ways there are of mapping a set of inputs to a set of outputs. Now, this is where Wolfram Alpha comes in because, in order to answer the question fully, I went over there, and I put into this incredible computing engine, I put in 2^128, surrounded in parentheses, and gave it a bang, a factorial, an exclamation point. It chewed on it for a shockingly short time, and it told me that the answer was approximated as 10^10^40.112.

**Leo:** Crikey.

**Steve:** And then I said, okay, how many digits does that have? It has 1.296 times 10^40 digits.

**Leo:** Wow.

**Steve:** Yes. So the answer to our Amsterdam questioner is no. There are only 78 decimal digits in the number of possible keys, yet there's a ridiculous 2^128 factorial possible encipherings. So what that tells us is that we are selecting, using the AES 256-bit key, we are selecting a tiny, tiny, minuscule, almost unmentionable micro tiny subset of the

total universe of encipherings that can exist given 128-bit input and 128-bit output.

**Leo:** Wow.

**Steve:** Loved the question.

**Leo:** Yeah. And I think I found a web page - almost as hard as the answer to that question is pronouncing his name. This says Egbert Douwes is pronounced [electronic voice] "Egbert Douse." Egbert Douse.

**Steve:** Wow.

**Leo:** Okay? So just - I figured it out. And now Mark Black, which is a little easier to pronounce.

**Steve:** Much.

**Leo:** Much. Mark Black from Lancaster, United Kingdom, just encountered TCP stack OS fingerprinting.

**Steve:** Okay, now, Leo, prepare yourself for this, because as I was putting this together I was again channeling you. But go ahead.

**Leo:** Okay. Okay. I was doing a routine security check and came across a website which correctly identified my OS, even though I blocked all JavaScript, have a local proxy (Privoxy), double NAT, NoScript, uBlock Origin with all lists enabled, Disconnect, Random User-Agent Spoofer with no DOM or APIs allowed - nothing. I even disabled almost everything I could in "configuration mania." Heck, I don't even install Adobe or Java on the OS.

And then I was quite annoyed to learn that something called TCP/IP OS fingerprinting - or "tikipipos" fingerprinting - has been leaking my OS the whole time. I completely missed that one, and I'm sad to say it took the wind out of my sails a little. And now I am brainstorming on how to obfuscate this in my Mac. If you have covered this in Security Now!, I apologize, as I must have missed your discussion of "tikipipos." I really hope you see fit maybe to mention it so we can muddy the waters a little in this respect, perhaps on the show if you think it's a valid subject. All the best from Lancaster, England, Mark Black. What is that "tikipopos"?

**Steve:** So this is something we've never talked about, and so Mark is right. He didn't miss it. He's listened to every podcast, and I don't think I've ever, except probably in passing, mentioned basically TCP/IP stack fingerprinting. The idea is that there are a number of parameters within the TCP protocol definition which are left up to the implementation. They just don't really matter that much. Consequently, over time, different operating systems and different versions of the same operating system have

just chosen different defaults for these values.

A perfect example, for example, is the TTL, the Time To Live. As we know from our early discussions of how the Internet works, that's a little counter, it's eight bits, which every time a router forwards the packet, it decreases that count from what it received to what it sends. And at some point, if the packet just keeps bouncing around the Internet, that TTL, the Time To Live, will go to zero, meaning that the packet dies, thus its name, Time To Live. And this keeps stuff from getting stuck in loops where it just circulates around the Internet forever. It's one of the brilliant foresightful things that the original designers of the Internet came up with.

So different OSes have set over time the TTL to different values. Once upon a time they were like 16 because the Internet wasn't very big. Then the diameter, the router hop diameter grew, and packets that were launched with 16 couldn't even make it to the other end, to the other side of the Internet, without timing out. So then OSes said, uh-oh, and they went to 32 or 64, generally powers of two, or 128, and maybe 255. So that's an example.

So the point is that people who are really interested in looking for, I mean, we're talking sort of more high-end hackers. One of the tricks of the trade has been to fingerprint, through the network, fingerprint the OS and maybe even the version, just by the way its TCP/IP stack acts. What's the largest TTL seen? What's the initial packet size? That might vary. The window size is a parameter that allows packets to be in flight without being acknowledged. That might just start off at a different value, and often does. There's also something called "window scaling" because the window size was only 16 bits, so now there's an eight-bit scale factor that's been added more recently to allow larger window sizes since bandwidths have gone way up, and we might need way more data to be in flight without being acknowledged, and so forth.

So I just liked the question. Mark is upset because it turns out that it's possible for someone to know that he's using a Mac. And he disclosed that. So, okay, now everybody knows. Sorry about that. There was a service, net - boy, I can't remember. It was a service that sort of looked at the uptime of servers. I tried to find it last night, but I was unable to find it, NetWatcher or something. And I always got a kick out of the fact that it could never fingerprint GRC's servers because they're behind a custom thing that I created years ago which is sort of my own TCP/IP stack. And it just never tried, it could never figure out what OS I was using.

There are masquerading, TCP/IP stack masquerading programs. Unfortunately, not for the Mac. Mark, I'm sorry, they exist for Windows and Linux and that ilk. But as far - and maybe you could use - I'm trying to remember if I saw one for FreeBSD. I might have. And so there's a chance it might be - it would have to be compiled for the Mac. Anyway, what you want to look for in googling is OS fingerprinting, or maybe TCP/IP stack fingerprinting, things like that. You'll find a ton of resources on the 'Net because this goes way back to the beginning. And it's an interesting topic, you know, kind of wacky, but fun.

And I don't really know if it matters. I mean, the original idea was that OSes themselves had major flaws, and so fingerprinting was the first thing a serious hacker would do in order to attack an operating system that was on the 'Net. If they knew what OS it was, then immediately they would turn to the subset of things that might be wrong with this particular instance of the operating system and not bother with all kinds of things that are wrong with all the other operating systems. So it was a quick way of weeding things out.

**Leo:** Are you talking about Netcraft?

**Steve:** Netcraft, yes.

**Leo:** I didn't channel that one. Jaguar in the chatroom did.

**Steve:** Ah. Netcraft, yup.

**Leo:** Martin Grigg in Vancouver, Canada wonders about the safety of links in Gmail: I use Thunderbird for my email. And I understand that clicking on an email link could allow a malware script to run in my user account. We were talking about that the other day. We found that Windows just will randomly run any JavaScript, or Java, right, even Java. However, when I open my Gmail account using Firefox and click on an email link, isn't that safe because it's inside Firefox?

**Steve:** Alas, no. And so basically, when you're clicking on a link in Gmail, you're in a website. And we know that clicking on links in websites can be a problem. Now, maybe Gmail is providing some protection, so there's that possibility. But that's sort of not the question he's asking. So I wanted to make sure that Martin understood that it wasn't email clients, per se, like Thunderbird, that are the problem. It's links that are deliberately imbued with a lot of power.

Just ask Tavis Ormandy what he was able to do with some links at the top of this show, and you can see how powerful they are. So that's a perfect example. If one of those links that Tavis found was in Gmail, and you clicked on it, you would activate a query to the localhost running in that Trend Micro user's machine and export all your passwords to the cloud. And that would work just as well in Gmail, if Gmail themselves didn't filter it, as it would in Thunderbird. So it's the links that's the problem, not the client.

**Leo:** Angelica Landry in Arizona notes that she is a girl.

**Steve:** And you'll know why I chose this. This is heartwarming.

**Leo:** Hi, Steve and Leo. First introduced to Security Now! in a security class while working on my undergraduate degree. Our professor required us to regularly listen to an episode - oh, thank you, professor - and write up a summary, including our thoughts. At first I saw how long the episodes were, and I thought, "Gasp, how will I have time for this and all my other work?" But after a few episodes I was hooked.

Two years later, and I still listen to you two. And I get excited when I understand and can follow along with various subjects. When I started getting behind and missed a lot of episodes due to life getting in the way, I began making Security Now! episodes part of my morning routine, and I listen while in the shower and getting ready.

But one thing I've noticed on Q&A episodes, there are a lot more males sending in questions and feedback. I want to hear an occasional shout-out to your female fans out there. Even if you don't do a shout-out, just know I'm a female listener and a big fan. Thanks for what you do. I appreciate you both. Thank you, Angelica.

**Steve:** I liked that.

**Leo:** It's nice.

**Steve:** And she's right. I really, from her perspective, I could see where she's going, hey, you know, where are the girls? And we never have any. So now we do.

**Leo:** The network itself, TWiT itself is about 95% male. But there are women, lots of them, in chat.

**Steve:** Oh, you mean in terms of the demographics of your followers.

**Leo:** Demographics. Last time we did a survey, which was a while ago.

**Steve:** Right.

**Leo:** But different shows have different percentages. iOS Today and The Tech Guy show absolutely have far more female fans. This show would probably have the lowest number, and that's something I'd love to fix. It's just kind of the nature of, right now, the state of the [crosstalk].

**Steve:** Well, we'll fix it one Angelica at a time.

**Leo:** Yeah. Certainly, I hope we - and let us know, Angelica. I hope we don't do anything to discourage female listeners. I hope, I absolutely hope we don't do that.

**Steve:** I'm sure we don't.

**Leo:** Jared in Australia brings us the Uber-Geek Tip of the Week: Mozilla has more DIY config options than I ever knew existed. For user convenience, Firefox trims the HTTP(S) protocol designation off the front of the displayed URL, even though it's present if you copy it to the clipboard. But I wanted to see it. By the way, this is a trend in browsers I do not like. Safari does the same thing. In fact, Safari trims everything off except the root domain.

**Steve:** You can hardly tell where you are.

**Leo:** I hate it. Although, you know, Tim Berners-Lee, the creator of WWW, did tell me, "I never thought humans would read these things, or I wouldn't have made them so non-human-friendly." These are supposed to be machine-readable URIs, http://.

**Steve:** Oh, that's right, because he was originally burying them in hrefs.

**Leo:** Right.

**Steve:** And so they were readable links, yeah.

**Leo:** Machine-readable. He never thought about browsers, people typing in URLs, all that stuff. That just didn't...

**Steve:** Yeah.

**Leo:** Remember, when he was doing this stuff, Gopher was kind of the state of the art for navigating the web.

**Steve:** And WHOIS was an exciting adventure.

**Leo:** Yeah, ooh, finger me. So in Firefox, in the about:config page, which everybody should browse around, and we've talked about that a lot, search for the word "trim," and you'll find the option to disable that display-simplifying trimming: Set "browser.urlbar.trimURLs" to "false." Uber cool!

**Steve:** I really, you know, I have to say, this bothers me, or bothered me. It hasn't now for the last day since I found, or since last night I found Jared's note. I went to about:config, typed in "trim" into the search term. It pulled up two instances of the bazillion little config options that are there, and I double-clicked on it, and it flipped it to false. And now I can see http://. I see the whole thing.

And what was interesting is that I'm cutting and pasting links a lot as I'm doing newsgroup postings and sending things to Twitter followers and so forth. And I'm always sort of nervous because Firefox is not showing me the whole URL. And of course I verify that it does paste it into the clipboard. When I do a copy, it's in the clipboard, so it does then paste properly. But I'd rather see it than not. So Jared, thanks for the heads-up on that. And to all of our listeners who would like to know about that, I'm glad to we do now.

**Leo:** I'm with you on that. I think that's, well, because we're old-timers. But I think for your own safety it's really important to see where you're going.

**Steve:** Yeah, just as a double-check.

**Leo:** Yeah. So that's why I'm not crazy about that. Charles Turner has our last question, I'm sad to say, our last question. He's from Virginia Beach, Virginia, and he's wondering what we think the future holds: Steve and Leo, I'm taking advantage of the holiday podcast doldrums to get caught up on back episodes of Security Now!. Way back in Episode 231, that was January 14, 2010 - five years ago - you were rather down on Apple's looming announcement of the iPad, not knowing how the iPad would shape technology during the subsequent years. Granted, making predictions is a precarious business. But if you need fodder for upcoming Q&A episodes, I'm wondering, what do you see on the horizon that may shape computing, in the loosest sense of the word, technology for the next five to 10 years? So a belated Merry Christmas and a Happy New Year. Sincerely, Charles.

**Steve:** Sort of a fun...

**Leo:** Were you down on the iPad? I don't know if I was down on the iPad.

**Steve:** And I tell you, I mean, I...

**Leo:** You loved it as soon as it came out.

**Steve:** Yes. And I have said ever since, it is the number one device, aside from sitting in front of my whole workstation with five screens as I am right now, being illuminated in their glow, the number one device I use, I mean, second only to my main machine. I'm a completely sucker for it. So, yeah, I'm...

**Leo:** I will stand by my statement on the day of, which was actually, now that I think about it, January 10th...

**Steve:** You know, the name we were annoyed with. Remember we thought, iPad?

**Leo:** Yeah, that was a dumb name.

**Steve:** Yeah.

**Leo:** But I remember saying, after Steve's announcement, coming out and saying on our broadcast, this is the computer Apple was made to make. This is what Steve Jobs always wanted to make. This is the future of computing. I think I said that. In fact, I would say I was more bullish on it then than I am now. I don't think it's actually lived up to some of its promise.

**Steve:** Well, and I think it hasn't grown well. I loved the dialogue that you were having

at the end of MacBreak Weekly, talking about, for example, I mean, basically a lot of what you were saying was that - you were talking about the new version of iOS, the little incremental release that adds a few things. For example, the nighttime, the removal of blue from the screen is one that I think is very, very cool.

**Leo:** Interesting, isn't it, yeah.

**Steve:** Yeah. And, you know, all...

**Leo:** So what do you think? I mean, he's giving us a chance to embarrass ourselves five years from now. I don't like to predict technology futures because I think one of the things that makes technology interesting is that really it's driven by the discontinuities, by the paradigm shifts, by the unpredictable shifts. So you can predict, obviously you can predict, based on what's going on right now, where those things will go. But what you really don't know - and Bill Gates famously missed the Internet.

**Steve:** Yeah. And so you're right. So there are things we cannot predict. My feeling is that mobile technology is probably past its dramatic revolution stage, in the same way that desktop computing is. Microsoft basically is looking for stuff to do in order to create new versions of Windows. I mean, for example, it's why I'm using an OS that was released in 2002 just fine. Works perfectly for me. Nothing has happened in the last, what, 14 years that, I mean, I'm using newer applications, but not the OS.

So I think my feeling is that mobile technology, we're not seeing revolutions now, we're seeing evolution. So I think we're refining what we have, making tweaks to things, things like UI tweaks, like the force input that Apple has just added and so forth. And I think still unclear is whether watches will ever be more than just sort of a small market curio, maybe multifunction. And problems will get fixed. I remember the first laptops that were "luggables," as we called them, that bear almost no relationship to what we have now. So the early, you know, the first-generation watch problems, I'm patient to think that they'll ultimately get solved. And I think that stereo goggle technology, to step back and generalize all of that, the whole idea of creating more immersive gaming experiences, that's something that we're going to see.

And we really are seeing, as the cost of technology drops, new things possible. Like I also love my Amazon Echo, which is listening to me, and then the television sets off from time to time, more than I do. And so that kind of technology. And clearly we're going to see what's going to happen with the Internet of Things. Lots of ideas. Many are dumb. Some are good, like the Ring Doorbell. That's a win. That's an Internet of Things win. Whereas, you know, I'm not so sure that something that smells inside of your refrigerator and tries to send a signal outside...

**Leo:** That's a pretty funny one.

**Steve:** ...is going to catch on.

**Leo:** Yeah, it's...

**Steve:** And from a Security Now! standpoint, I think we're going to see the continued decline of third-party plugins, famously Java and Flash, and them being replaced with the ever more capable native JavaScript and HTML5 platforms. So those early sort of crutches to creating more capable web experience, that just - it's going native. It's going into the platform. And IPv6, I think it's - even though, I mean, we talked about it last week. Last week's Picture of the Week was that, not quite exponential, but it had an exponential liftoff; then it was going linear. But it was definitely happening after 20 years. In the last few years it's happened because we actually did start running out of IPv4.

So I think, you know, certainly what'll be interesting is IPv6 being offered by ISPs, and then routers sending them into our home networks so that we're no longer 192.168, we're actually a 128-bit IP. And then every device in our home, all of these IoT things, can have its own IP. So, yeah, there will be some changes. And I think a lot of refinement.

**Leo:** I can make one prediction. This show will be around in 10 years, and we'll still be talking about hacks and flaws. And, you know what, we'll still be able to do that segment, "You're Doing It Wrong."

**Steve:** Yeah.

**Leo:** Because that's not going away.

**Steve:** Nope.

**Leo:** And as stuff gets more complex, it doesn't get more secure. But I don't think the bad guys will win. I just think that it will be a continued seesaw back and forth.

**Steve:** Think it'll be, exactly, it'll be a simmer, like we have.

**Leo:** Yeah, yeah. Well, that's good news because we need Steve Gibson here, whether there's security flaws or not, every Tuesday at 1:30 p.m. Pacific, 4:30 Eastern, 21:30 UTC. You can count on that. Steve never misses an episode. And if you want to watch live, we would love it if you do, but you don't have to. You can always get on-demand audio and video after the fact.

Steve's got the audio and transcripts, too, so you can read along as you listen, really well done transcripts that Steve pays for from Elaine Farris. And those are at GRC.com. While you're there, do ShieldsUP! because it works, get SpinRite because it works, and all the other great services that Steve provides. The only one you have to pay for is SpinRite. Everything else is free. SQRL's coming along. I bet we're going to get a SQRL update sometime in the near future.

**Steve:** Yeah, we just, two days ago, I just keep thinking we're done, and then it's like, oh, wait, wait, wait, we need to fix this.

**Leo:** One more thing.

**Steve:** So we just changed, we're in the process of changing the storage format. SQRL used to allow you to change, to rekey your identify once, the idea being you may never need to, but if somehow it got away from you, you might need to rekey. The problem is that the system only supported one - it only supported the storage of a single previous key. And that's important because, when you went to websites that knew you by your previous key, you need to seamlessly update to your current key. All of that works. All of that's transparent. It's very elegant. But there was always some contention in the newsgroup where I've been doing all this, that we needed more.

And so, after around and around and around, we decided on four, four previous keys plus your current one. But doing that necessitates reengineering the identity storage for technical reasons. So I'll be doing that in the next couple days. But we're getting very close. Once that's done, there's one bug that was reported, and a bunch of cosmetic things. And I was never using the term "rekey," which I'm working to be conscious of. We used to talk about creating a new identity. But you're really rekeying your identity, which I think is much clearer. So I have to change some of the UI text to catch up. I've just been working on the core technology. But very close.

**Leo:** Yeah. Good. Can't wait. We also have the show at TWiT.tv/sn, and you can subscribe on every podcatcher known to man and get the apps on the Apple TV or the Roku or the, I mean, they're just everywhere. Security Now!. You ought to listen. If you want to be in the studio, we do have limited seating during the show, and you can always come join us. We have a nice couple who's here visiting us from Wisconsin. They're falling asleep, but that's okay. It's the afternoon. You can always email tickets@twit.tv. We'd be glad to have you. Thanks, Steve. We're always glad to have you, and we'll see you next time.

**Steve:** Always fun, Leo. And I understand why they're sleeping. Two raised to the 128 factorial doesn't wind everyone's clock. But I know that we have some listeners who get a big kick out of that, so...

**Leo:** That's hard.

**Steve:** Yeah. Thank you, my friend. See you next week.

**Leo:** Thank you, Steve.

**Steve:** Bye-bye.

Copyright (c) 2014 by Steve Gibson and Leo Laporte. SOME RIGHTS RESERVED

This work is licensed for the good of the Internet Community under the Creative Commons License v2.5. See the following Web page for details: http://creativecommons.org/licenses/by-nc-sa/2.5/