SECURITY NOW!

Transcript of Episode #528

Breaches & Vigilante Worms

Description: With many massive Internet data breaches, and a prolific vigilante worm loose on the Internet, Leo and I spend a fun- and fact-filled podcast covering the past week's multitude of news.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-528.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-528-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. My goodness, it's been a busy week. This is going to be an all-news episode. We'll talk about the vigilante worm. We'll talk about security breaches at some very well-known sites. And we'll also talk a little bit about updates that make Marshmallow, Android 6.0, more secure. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 528, recorded Tuesday, October 6th, 2015: Breaches & Vigilante Worms.

It's time for Security Now!, the show in which we protect you and your family and your loved ones online. We scour the Earth for new vulnerabilities, attempt to plug the holes, and it's all thanks to this guy.

Steve Gibson: Actually, they find us, Leo. We need not scour. And if I thought I ever had to before, Twitter has solved that problem for me.

Leo: Yeah.

Steve: It's just - it's such a fabulous resource. I've got, you know, as I mentioned before - actually, I don't think I did mention that I mentioned we were nearly at - I. We. I've been watching too much politics, where it's like "our plan for this." It's like just, you know, it's...

Leo: It's all "we," yeah, the royal "we."

Steve: ...we, we, we.

Leo: Yeah.

Steve: Anyway, yeah. So I mentioned that I was nearly at 50,000, and a couple days later, like three days later, I crossed that barrier. So, and speaking of which, Snowden, last time I looked, you have to look every so often because he gained 20,000 followers from last night when I looked, when I was putting the notes together, to just now when I looked. He's at 1,383,000 something followers. And I love it that he is following exactly one: the NSA.

Leo: Yeah.

Steve: I just get a kick out of that. Of course he's just doing that to make a point. But great podcast today. We had a week full of news. We've got breaches at Patreon, Experian, and Scottrade to talk about. The return of StageFright. A vigilante worm loose on our routers. Problems with the VeraCrypt full-disk encryption solution that I talked about, that I suggested people start migrating to. I think you need to wait, if you haven't already moved.

Leo: Uh-oh.

Steve: A bunch of follow-ups and minor notes. Then I want to talk about something that came up, that arose in a security developer's mind as a consequence of the trick that Volkswagen pulled on the world. And also I've got a summary of the top 10 major security improvements in Google's recent release of Android Marshmallow. So just a pure news and information podcast.

Leo: Yikes. It's jam-packed with goodness. All right, Steve.

Steve: So our Picture of the Week on the front page or the first page of the show notes is something that caught my eye. It's an angle to the whole adblocking aspect, believe it or not, that we've never covered. And that is someone did an analysis of the cost people pay for bandwidth compared to the ad revenue generated by the bandwidth and realized that cellular carriers are making far more from mobile ads than the publishers are.

Leo: Oh, that's a hoot.

Steve: Isn't that a kick? So this is the same, you know, the same sort of analysis we've seen before where, with an adblocker, the L.A. Times, which was used as an example, loaded 20 files and 1.7MB in three seconds. Without the adblocker, it was 178 files, as opposed to 20; 6.2MB as opposed to 1.7MB; and 12 seconds, up from 3.

Leo: Yeah. This is why people use adblockers. I mean, geez Louise.

Steve: Well, yes, yes. And notice what he showed was that it is scripts. Yes, it's also more images and some additional HTML. But most of the bloat is scripts. And this was what you'll remember me grumbling about because when I did a look at this, I saw, for example, that a link on a page pulled a third-party, a little bit of third-party HTML that had some JavaScript, which then invoked a library, because the guy wanted one function, wanted to perform one function that was in the library, and pulled this entire library in just to execute that one function. So it's the inefficiency, the fact that people don't, haven't needed to care, so they haven't cared. And as a consequence you get this kind of bloat.

But anyway, all of that aside, I thought what was really interesting is that it is the carriers that charge as much as they do for the data transit. They're profiting to a much greater degree than the actual sites that host the ads. So there's a different angle that we hadn't looked at before. Patreon.

Leo: [Whimpering]

Steve: Yeah, yeah.

Leo: This is so sad. And it's kind of a "shot themselves in the foot" moment, too, which really makes it sad.

Steve: Yeah, it is. So, okay. So probably people have heard that there was a massive data breach at Patreon. Jack Conte, whom you know, the CEO and cofounder, posted to acknowledge that they had had a data breach. He said: "The unauthorized access was confirmed to have taken place on September 28th via a debug version of our website that was visible to the public. Once we identified this, we shut down the server and moved all of our non-production servers behind our firewall." Well, okay. First of all, to our audience, everyone says, wait a minute. You then moved all of your nonproduction servers behind your firewall? Where were they before? Well, unfortunately we know that they were public facing, at least one of them.

Leo: That's not unusual, by the way.

Steve: No, it's not.

Leo: Because, for instance, our sites run on Heroku because they're Node.js. And we can't run Heroku behind our firewall. I mean, we can have a copy of the website behind our firewall. But once you get to testing, you kind of need to put it in the environment.

Steve: Right.

Leo: So we have dev and staging and production servers. We trust, we hope they're well secured. But they're not on our premises.

Steve: Right. So he says: "There was no unauthorized access of our production servers. The development server included a snapshot of our production database, which included encrypted data. The development server did not have any private keys that would allow login access to any other server. We verified our authorization logs on our production servers to ensure that there was not any unauthorized access. As a precaution, we have rotated our private keys and API keys that would allow access to third-party services that we use. We protect our users' passwords with a hashing scheme called 'bcrypt'" - which of course we've been speaking of recently and talked about back when we were talking about password-based key derivation function.

Leo: And that's the good scheme; right?

Steve: Yup.

Leo: That's the one you should use.

Steve: Yup. They did everything right. And...

Leo: Well, not everything.

Steve: Well, I know. We'll get to that - "and randomly salt each individual password," which is what you have to do. Bcrypt, he says, for those who don't know, "is non-reversible, so passwords cannot be decrypted." On the other hand, you know, we know that they can technically be brute forced. And remember that, you mentioned it on Sunday, the other site that was just hacked.

Leo: Ashley Madison used bcrypt.

Steve: Oh, Ashley Madison. Ashley Madison also used it well, but they also used earlier, weaker forms. And even without that, they were still able to brute force. So while you cannot reverse it, you can go forward with a great assault and sometimes still succeed.

Anyway, so what we know is that hackers have since published nearly 15GB of password data, donation records, and the source code taken from Patreon's development server. So basically they had full access to the storage, the mass storage of this server, which had a static snapshot of their database. And it was a recent snapshot. I don't think I have here in my notes the date. But I remember in digging into this that Troy Hunt, who was the security researcher, he found, like, it was maybe a week or two old. So it was, you know, not the live data, but it was a snapshot of the live data, so it might as well have been.

Now, Troy runs the "Have I Been Pwned?" website. And so he wanted to go through it in order to extract email addresses so that he could add those to the "Have I Been Pwned?" website, where you're able to put in an email address and, in a secure fashion, Troy checks your email address against his multiple lists of prior breaches. And, now, so this has taken analysts like Troy a while because, while 15GB may seem like a treasure trove, it's also 15GB. I mean, it's a huge bunch of data. So he has gone through it now. He has found 2.3 million unique email addresses, including his own because he was involved with Patreon. So those are now part of HaveIBeenPwned.com. So anyone who's interested can put their email address in, and it should turn up.

So he said: "The amount and type of data posted by the hackers suggest the breach was more extensive and potentially damaging to users than was previously assumed." Troy wrote: "You can determine how much those using Patreon are earning, and everything private is now public." So, you know, this is nothing less than a devastating breach for Patreon to have suffered, though not technically a security breach, like in terms of being able to hack people's accounts. As long as you had a strong password, Patreon did always protect your password using state-of-the-art protection. Unfortunately, a snapshot of a recent copy of their apparently entire database was able to get loose.

Now, what was doubly damning is that Patreon was notified by a Swedish security firm, Detectify, five days before this occurred. We've discussed before this disturbing search site. And Leo, you should go there because the front page now is really pretty funny. It's Shodan.io, S-H-O-D-A-N dot I-O. And it sort of rotates through, it's got a little sort of a banner that rotates through the things that...

Leo: O-D-E-N or D-A-N? It's D-A-N.

Steve: D-A-N, Shodan, Shodan.io. And it shows you - so the search engine for the Internet of Things. The search engine for webcams. The search engine for buildings. The search engine for the web. The search engine for refrigerators. The search engine for power plants.

Leo: Oh.

Steve: It's like, oh, goodness. And so what happened was, I mean, so what this does is, this allows people to search typically for things in headers. So this, for example, will go through doing a port 80 request for every IP on the Internet. Now, so port 80 is HTTP. And one of the things that a web server says when you connect to it, it identifies itself, you know, like what type of web server it is. So that's how we know, like, how many Linux servers there are, how many Nginx servers there are, how many IIS servers there are and so forth, is that these identify themselves.

Well, Patreon's R&D server was using a well-known Python utility library called Werkzeug, W-E-R-K-Z-E-U-G, Werkzeug. And it identifies itself in the headers. It says Server: Werkzeug/8.9.6 Python/3.4.0. So here's the problem. Shodan allowed this Swedish security firm Detectify to find thousands of public servers that have Werkzeug facing the Internet. One of them was Patreon.

Now, Werkzeug is - the dangers of letting this be public are well known. Werkzeug cautions people not to do it. The default binding, that is, in terms of UNIX ports, the default port binding is 127.0.0.1, which we know is the local host IP, meaning that the Werkzeug server will only be available locally. The Patreon people changed it to 0.0.0.0, which means accept connections from any IP, bind to the public-facing interface.

Leo: Oh. That's what it means. Oh.

Steve: So then the problem is that, while you do need nonpublic and not available keys, that is, as a developer...

Leo: The API keys.

Steve: Yeah, exactly, to access the UI, essentially, in order to get into the debugging mode, you need a secret. But the other problem is, if anything ever crashes, the debugger is immediately brought forth. So someone saw that this was the case. They probably used Shodan to find Werkzeug, saw that it was Patreon. It wasn't Zeus, oh, yeah, Zach, Z-A-C-H, dot Patreon.com. That was the domain name of the machine that Shodan found. So they went there, and they managed to cause a fault. They generated an error somehow that gave them the Werkzeug interface. And from that they're able essentially to - you have full remote code execution capabilities. They probably just simply launched a remote shell, had then root access to the server, and were then able to exfiltrate, you know, the contents of the machine.

Leo: Now, I still see 15,000 Werkzeug servers when I search Shodan for that.

Steve: Yes, they're - yes. I mean, it's terrifying. I mean...

Leo: A lot of people with public-facing Werkzeug.

Steve: Yep.

Leo: Not just Zach.

Steve: No.

Leo: I'm thinking Zach probably doesn't work at Patreon anymore. I'm just guessing.

Steve: Yeah, it's not good. Not good.

Leo: But, yeah, don't leave a debugger running on your web server.

Steve: So that, yes, that is the story. No doubt Patreon is the most famous instance of this. But this is why Shodan is both useful for security firms, but unfortunately so exploitable for bad guys, is what - and we've talked about Shodan before. Remember that there was a webcam that had a default password, we learned. Well, you just drop the webcam's name into Shodan because it was part of...

Leo: Yeah, there it is, yeah.

Steve: It was part of the headers. And it was like, oh, look at, oh, here we have 3,000 webcams we can log into and play with.

Leo: Look, here's DuPont running Werkzeug on their server.

Steve: Yeah, wow.

Leo: Yeah. This is amazing. What a great - I like Shodan. What a great site. Wow.

Steve: No, Shodan is like it's the perfect, you know, chaos. It's the chaos search engine because it's - all it's doing is indexing what's publicly exposed.

Leo: Right.

Steve: And so it's like, well, hey, you know, we'll show you. And it can be very useful for both good and bad.

Leo: This is not, wouldn't be a standard spider crawling. It would have to be they're checking, they're port knocking or something; right? It's just the ports.

Steve: Yeah, well, no, they're just scanning.

Leo: Scanning ports, yeah.

Steve: Yeah, exactly. They're just scanning every single IP on the 'Net. It turns out that scanning's going on all the time. And this is how, for example, when we report that 2,300 routers have open Telnet ports, well, it's because somebody, you know, like one of the security researchers just said, I'm going to scan for port 23 over the entire IPv4 space, and says, oh, look, 2,300 Telnet servers. Wonder what's behind those?

Leo: Somebody should write some malware that just turns that Telnet port off on routers. That's what somebody should do. Did you see that?

Steve: It's on our notes here.

Leo: Oh, it is. All right. Okay.

Steve: That's the vigilante worm, my friend.

Leo: Oh, that's the vigilante worm. That's what we're talking about.

Steve: That's the vigilante worm. First we have a few more massive breaches to cover.

Leo: Oh, there's no end.

Steve: Oh, because it turns out that T-Mobile was very unhappy with their subcontractor, Experian. Experian is one of the three major credit reporting bureaus. Experian lost - they had a breach, and they lost the credit applications of 15 million T-Mobile customers. And unfortunately what - so these were credit apps. On a credit app you put basically...

Leo: Everything.

Steve: ...everything - your Social Security number, your passport number, your driver's license number, your name, address, your driver's license, the works. Basically you dump out, I mean, basically everything you need for identify theft. And it was all there. It was all being retained. Guidelines require that that information be kept for two years. So for the most recent two years, and I'm assuming they're expiring them after two years, but I don't know that for a fact, it was 15 million, however long that is.

Experian said: "Experian North America today announced that one of its business units experienced an unauthorized acquisition of information" - there's a new one, it was an unauthorized acquisition of information - "from a server that contained data on behalf of T-Mobile, USA, Inc. The data included personally identifiable information for approximately 15 million consumers in the U.S., including those who applied for T-Mobile USA postpaid services" - as opposed to prepaid - "or device financing from September 1, 2013" - okay, so there is, there's our two years - "through September 16, 2015, based on Experian's investigation to date. The data acquired included names, dates of birth, addresses, telephone numbers, and Social Security numbers and/or an alternate form of ID like a driver's license number, as well as additional information used in T-Mobile's own credit assessment."

Experian, big-hearted people that they are, "is offering affected consumers two years of free credit monitoring through a service they own, ProtectMyID.com." Having of course lost 15 million consumers' IDs. And what I loved about it, in some of the coverage of this, it was noted, I thought, quite correctly, that they're offering two years of free credit monitoring, having lost information that lasts a lifetime. That is, there is no expiration date...

Leo: Right.

Steve: ...on the stolen data.

Leo: And of course ProtectMyID is a part of Experian. So that's nice.

Steve: Really big of them.

Leo: Yeah.

Steve: So, yeah. Okay. So we talked about this before. I've got a bunch of links in the show notes. This is yet another reason to freeze your credit. ClarkHoward.com is the site that I referred to before because it has a very nice coverage, the ClarkHoward.com Credit Freeze and Thaw Guide. I imagine if you Google some of that, it'll find the link for you. But About.com has one. CreditCards.com has one. The FTC.gov site has coverage for this. Unfortunately, it's not free. You need to pay once to lock the credit, typically $10. Some states have it at $5. And then that's $10 per bureau, so 10 times three. And it's going to be inconvenient if you're in a period in your life where you're constantly needing to apply for credit. I'm past that, so I don't mind having mine locked. I want mine all locked. But you can then pay to selectively or permanently unlock.

It grinds my something that the credit bureaus are making money on this. This ought to be a free service. You ought to be able to lock yourself, lock the data they have collected on you without you asking them to. I mean, they're profiting from having this database. On the other hand, I guess once you lock it, then nobody - then they're no longer able to profit from it for you, so this is payment in lieu. But still, they're obviously, I mean, this wouldn't protect people from breaches. But what it does is it protects people from - because the way this data that has been lost is used, is it's used to impersonate you to apply for credit and then use that credit, for example, to apply for a credit card, which could be granted. Then somebody who is impersonating you would suck out the money from the credit card, and you would be held liable.

And of course we know identity theft is very difficult to recover from. Though hopefully it's getting easier because it's now not like something no one's ever heard of. People realize this is something that actually happens to people and takes years often to recover from. So really, seriously, if you aren't actively using the credit that you have established, I think it's worthwhile to freeze your credit, which you can do at each of the three different bureaus.

And not to be left out, since we have a triple-header of breaches this week, Scottrade also reported a data breach. In this case, 4.6 million customers, anyone with an existing Scottrade account prior to February of last year. And what's interesting is that I don't know how - there was no explanation of the delay. But they found that a data breach had taken place over several months from late 2013 through early 2014, and they're saying February 2014. So that explains why it's people who created Scottrade accounts for the first time after February 14th of last year wouldn't be subject to this 4.6 million customer breach.

But once again, a lot of personal financial information - names, addresses, Social Security numbers, and other personal information, probably whatever accountholders at Scottrade had to do. You know, they had to have Social Security numbers probably for tax reporting reasons because they're a stock trading service. Login information and trading platform information were not affected, which is to say probably this is the customer account data, but not things like your stock portfolio is I think what they meant when they said that trading platform information was not affected. So what they're feeling is that maybe this would open people to social engineering attacks against their customers. On the other hand, there looks like plenty of information here for, again, another identity theft exploitation.

And I know you talked about this on Sunday, Leo. StageFright is back for Round 2.

Leo: [Sighing]

Steve: Yeah. So what we know, and we discussed this during StageFright 1, if we call this one 2, is that this is a very badly written module. And just in looking through it, the people at Zimperium just found a bunch of problems, which we covered in great detail, looking at the math being done with some typecasting in C that wasn't handled correctly. And of course we also covered the fact that the first time that one of those was patched, it was patched wrong so that an "if" statement could still be caused to take the wrong branch and recreate the problem, which is why there was that later problem that was only then later picked up by the StageFright testers, and everyone got their earlier things patched, but then there was this last thing that wasn't patched.

Well, we're all back there again because there is a raft of critical remote code execution vulnerabilities which have again surfaced, which affects all versions of Android since 2010, so for the past five years. It is in - I have, I wrote October 5th Nexus OTA update. That was yesterday. So presumably the over-the-air updates for Nexus, at least Google is pushing this stuff out. We know that Google made the patched code available to their partners back in early September, so early, almost a month ago, September 10th or earlier, their partners were notified.

So again, what we saw was maybe, what, four to six weeks it took for the non-Nexus devices to finally get themselves updated. We can sort of predict something similar. This is a little different. Also worrisome. Because StageFright deals with the rendering of multimedia, Google wrote that the most severe of these issues is a critical security vulnerability that could enable remote code execution on an affected device through multiple methods such as email, web browsing, and MMS when processing media files.

Leo: Well, basically just have a malformed MP3 or MP4; right?

Steve: Correct. Correct.

Leo: So you could get it there anyway you could get it there.

Steve: Yes, exactly, if you send it through email, or if you go to a website that induces you to play it. So, yes, anything that is able to get you to run that will cause the problem.

Leo: Including, now, I wonder, the new Twitter, which plays - and Facebook, which play videos inline. I wonder, you know? Because even a preview would be sufficient.

Steve: Yeah. Google said: "We have no reports of active customer exploitation of these newly reported issues." And then they said: "Refer to the Mitigations section for details on the Android security platform protections and service protections such as SafetyNet, which improve the security of the Android platform. We encourage all customers to accept these updates to their devices." Yeah, no kidding. So I looked, and I couldn't see anything - they actually didn't have any mitigation worth speaking of.

Leo: Yeah. It's "Don't open your email."

Steve: Yes. Maybe do what Leo has done. Take a recess from Android, switch to iOS.

Leo: That's one solution.

Steve: Just for a month or two.

Leo: Not the mitigation Google was hoping, but...

Steve: Not quite what they were - and I have to say, we'll cover it at the end of the show, they in - I want to say Mushroom, but it's Marshmallow, in Marshmallow they...

Leo: Is it fixed in Marshy Mellow?

Steve: No.

Leo: No. Gosh.

Steve: No, it was after Marshmallow, unfortunately. But, you know, they'll be catching up quickly. And I did not look at the Zimperium app. I meant to, but I just ran out of time. We actually had several hours of power outage this morning, so I was sprinting. And luckily I did most of this work last night, just for some reason.

Leo: Thank you.

Steve: I was, you know, I was in the groove, so - as if I knew. But anyway, so...

Leo: You're wondering if they've updated the Zimperium app to test for this, too.

Steve: Yeah, I'm wondering if anyone in the chatroom...

Leo: Let me download it right now.

Steve: Or you, yeah, grab a new one. And what was the other one? There were two.

Leo: Oh, Lockout, or Lookout, Lookout had it. But it wasn't very good because it didn't protect...

Steve: You're right, right.

Leo: ...all of the original flaws.

Steve: You're right. I think Zimperium is the one to standardize on.

Leo: They had a Shellshock detector, and they also have a StageFright detector. Well, I'm downloading this right now.

Steve: So we can, now, as, you know, Google says no exploitation, except that we do know that we detected exploitation because the previous usage of StageFright, or misuse of StageFright, was MMS messages, which many people reported getting.

Leo: So this is the StageFright detector. It's still the old StageFright vulnerability.

Steve: Yup.

Leo: Because it says not vulnerable.

Steve: So you're seeing that...

Leo: Up-to-date Nexus 6. So it's...

Steve: So it's original 6, yeah.

Leo: Yeah, these aren't the new ones.

Steve: Okay. Okay. So I would just say pucker up.

Leo: Careful. I mean, it's got to be a malformed MP3 or MP4.

Steve: Yup.

Leo: Because StageFright is the media playing engine in this. I'm surprised they didn't replace that in Marshmallow, frankly.

Steve: Yes. Well, yes. Here's another classic example, and we'll be talking about this later, and we were talking about it last week, this notion of attack surface. What is exposed that the outside world can get to really has to be tight. And in the same way that, I mean, in retrospect, it was obvious that antivirus software had to be really written carefully because it was going to be filtering everything that came through it. So if there were any, like, if it was buffering stuff, it has to buffer. So if there were any overflows in that, bang, buffer overflow.

Similarly, here you've got a library that is by definition an attack surface because any multimedia that comes into the phone runs through that. So if the authors weren't really careful and good, and if it wasn't well audited, I mean, we can see that even auditing could have found a lot of these problems because that's the way they were found. They were not found through an active exploit or seeing somebody doing it or even fuzzing, where you just throw noise at it, and if it crashes, you go find out what you threw at it that made it crash because that could be an exploit. So, you know, but again, here we have, as you say, Leo, this is unfortunately a badly written library that is a big attack surface in Android at the moment.

Leo: The fix that Google's using for this kind of stuff is about all they can do because it doesn't - it's an open source operating system. It doesn't control who's using it. And they can't force carriers and manufacturers to update because they don't have that kind of clout. But what they are doing, which is smart, is extracting bits and pieces and putting them in the Play Store so they can push an update through the Play Store. They've done that with a lot of the Android services. They didn't do it yet with StageFright. But I would guess that's the next step is remove StageFright from the distribution and have it updatable through the Play Store. Then they could push an update out. The problem is, you know, you can't do the whole operating system that way. But you can do bits and pieces.

Steve: I saw some interesting commentary following Marshmallow's release that talked about how what we're sort of seeing is Google being forced, unfortunately, to be more than just sort of the overseer of a public community open source OS. The reality is they are having to get more involved in taking, like, frontline responsibility for the behavior. And I'm not surprised. I mean, this is what Apple has had to do and has continued to do. And we're seeing increasingly, you know, Google doing more and more of that, which I think just makes Android stronger from a security standpoint. The problem is, unlike apps, security is not sexy, but is absolutely crucial. And it's hard. It's, you know, it's hard.

Leo: Well, a conspiracy theorist might say this is exactly what Google wants because, as a result of making it open, they kind of lose control of Android, and everybody and their brother, including Amazon, can take it and do whatever they want with it. And, you know, from a business point of view, Google would prefer to have a closed source operating system that only they control. It would be more secure to be like iOS, but it would no longer be open. And that's kind of the tradeoff between...

Steve: Right.

Leo: ...openness and security, unfortunately.

Steve: Okay. Wifatch.

Leo: Love the name.

Steve: Wifatch. I don't know where the name came from. I didn't run across any derivation of the name. Maybe it's from the source code. Most of these things come from something in the source. It is the IOT, that is, the Internet of Things Vigilante Worm, Wifatch. This was actually discovered by someone like a lone security researcher, oh, I was surprised how long ago, like I want to say a year ago. Symantec is the one who brought it back to our attention.

And they wrote: "Let me introduce you to Linux.Wifatch, one of the latest pieces of code infecting Internet of Things devices. We first heard of Wifatch back in 2014, when an independent security researcher noticed something unusual happening on his own home router. The researcher identified running processes that did not seem to be part of the legitimate router software and decided to investigate further. During his analysis, he discovered a sophisticated piece of code that had turned his home router into a zombie, connected to a peer-to-peer network of similarly infected devices."

What does it do? It behaves just like a worm, scanning - and this is me talking now. I kept my Symantec voice by mistake. So what does it do? It behaves just like a worm. It's being called a "virus," but that's wrong. It's a worm because by definition a worm operates on its own, requiring no intervention from people.

So this is a worm running loose on the Internet. It scans for other opportunities, finds them, and infects them with a copy of itself. It remains hidden. It coordinates its actions through its own private peer-to-peer network. It contains, and the code has now been published and is public, so we can say, no malicious payloads. It hardens the security of its host devices. It kills any running Telnet daemon and any other problems that are known to affect routers who have public-facing vulnerabilities like, for example, Universal Plug-and-Play ports that are exposed. It keeps other viruses out by staying current on router vulnerabilities using its peer-to-peer network. It will retroactively remove any preexisting malware that it finds in the router and, as I mentioned, patches the router to cut off any other channels of entry.

And interestingly, the observation was made that, in looking, in reverse-engineering this code, there's, for example, a lot of Perl that could have easily been obfuscated by the authors, but they chose not to. You know, they weren't really trying to hide. And in fact, in one place in the code there's a copy of Richard Stallman's email signature, which reads: "To any NSA or FBI agents reading this: Please consider whether defending the U.S. Constitution against all enemies, foreign or domestic, requires you to follow Snowden's example." Seems unlikely.

So, now, Symantec estimates that somewhere, they don't have an exact count, but on the order of tens of thousands of devices are infected. In their - and I don't think I put up a - I didn't put a link here for some reason. Normally I would have. But in their analysis, in order of decreasing infection rate, China is No. 1. And remembering from memory the pie chart, it was over a third, like nearly a half of the infected routers were in China. Next up was Brazil, then Mexico, India, Vietnam, Italy, and Turkey. And Italy and Turkey, those latter ones were pretty small slices, but still enough to register.

And so what that probably means is it is tied to router brands. That's really the only reason that I can see - yeah, there's the pie chart. Leo has it up on the video. So it looks like, what, a third is China, I guess, and then maybe the balancing chunk is Brazil, and then Mexico in third place and so forth. So it must be popular brands which inherently have vulnerabilities that allow the worm to get into them. That's the only reason I can imagine. Either that or ISPs could be proactively blocking the ports at their borders that prevent access to their customers' routers behind the ISP's own border firewall. Those are the two things that could explain a geographic distribution that is so far from, like, the normal distribution based on numbers of routers.

So, okay. So then, in another twist, just yesterday, on October 5th, in an update to their posting, Symantec posted a screenshot of a dialogue that they had on their site with the author. And since then, Simon Zerafa tweeted a Forbes link that, again, I was short of time, so I didn't have a chance to put it in the show notes and even cover it because it was somewhat more refined. But I like this for - the one from Symantec, although it's barely legible, I've got a copy here zoomed way in so I can read it.

So Symantec asks, "Why did you write this and let it go?" Its author says, "First, for learning; second, for understanding; third, for fun; and, fourth, for your and our security. Apart from the learning experience, this is a truly altruistic project, and no malicious actions are planned (and it nice touch that Symantec watch over this)," the author said.

So then they said, "Why release now?" "It was never intended to be secret," writes its author. "And to be truly ethical (Stallman said), it needs to have a free license," which this person says "agree" in parens, "and ask before acting (also agree, so only halfway there)." Oh, interesting. So this person apparently presumes to make a future version which may notify its users somehow that it's in the router, and it's helping them.

So Symantec asks, "Why not release earlier?" Response: "To avoid unwanted attention, especially by other malware authors who want to avoid detection. Plan failed, unwanted attention has been attracted, so release is fine."

Symantec: "Who are you?" Response: "We are nobody important. Really."

Symantec: "Do you feel bad about abusing resources by others?" Answer: "Yes, although the amount of saved bandwidth by taking down other scanning malware, the amount energy saved by killing illegal bitcoin miners, the number of reboots and service interruptions prevented by not overheating these devices, the number of credentials and money not stolen, should all outweigh this. We co-opted your devices to help the general public (in a small way)."

Symantec asks: "Can I trust you to not do evil things with my devices?" Response: "Yes, but that is of no help. Somebody could steal the key, no matter how well I protect it. More likely there is a bug in the code that allows access to anybody." So this person is modest and realistic.

"Should I trust you?" asks Symantec. "Of course not," is the response. "You should secure your device." Which is wonderful.

Symantec: "Why is this not a problem?" Answer: "Linux.Wifatch doesn't use elaborate backdoors or zero-day exploits to hack devices. It basically just uses Telnet and a few other protocols and tries a few really dumb or default passwords. Our favorite is 'password.' These passwords are already well known. Anybody else can do that without having to steal any secret key. Basically, it only infects devices that are not protected at all in the first place!" exclamation point. So we have the do-gooder vigilante worm.

Leo: He's quite a Robin Hood.

Steve: Yeah. And, you know, back in the, boy, I think it was maybe Code Red or Nimda, those were both early notorious worms. I was involved in some dialogue with the DoJ at the time. That's back when I had been subject to a lot of denial-of-service attacks, and I was giving sort of like helpful, this is how the Internet works presentations to the FBI and law enforcement groups and so forth. There was some active discussion back then about whether we could write an immunizing worm. And of course the answer was absolutely not. It is absolutely illegal to modify somebody else's equipment without their express permission. And of course that was not available. So we didn't do it.

But, I mean, this guy is performing a public service. These are vulnerable routers, and he's, I mean, all of his logic is exactly right. He's not doing anything fancy. He's not using unpublished, unknown exploits. He's just gone in and taken over, closed the door behind him, removed the stuff he's found, and is then maintaining basically a public-facing security network among all of these routers that would otherwise be really prone to being victimized. So I say, sort of in the spirit of Edward Snowden, who this guy obviously follows, it may be technically breaking the law, but he's doing a good, you know, the outcome seems to be worthy of the method.

In our Q&A last week, someone asked about the HOLA distributed VPN network. One of our listeners apparently did some digging. And this is where I said we would get back to the issue of attack surface again. Turns out HOLA, in addition to all the other problems that I articulated, basically I worried that bad people could use this distributed VPN and, for example, your IP address would be associated with downloading illegal content or serving illegal content of any kind; and that that really, you know, it's one thing for a big VPN provider to have a massive endpoint node where if the MPAA comes knocking, someone like proXPN can say, look, we're a VPN provider. We're not doing this, and we're not logging, so we can't help you. An individual is in a far weaker position to defend themselves against that, in fact probably they have no defense.

But what's worse, as it turns out, HOLA has remote code execution vulnerabilities. So you're being a peer on this VPN network in order to use this facility. And unfortunately, that means that HOLA itself is creating an attack surface where none existed before, and it's vulnerable. There are apparently a bunch of remote code execution vulnerabilities. And it is the case that it has been and presumably is being used maliciously. That has been confirmed. So again - in fact, there's a site, adios-hola.org, that has chronicled a bunch of this, A-D-I-O-S hyphen H-O-L-A dot org, if anyone has any further interest, which I think would be ill-advised.

Problems with VeraCrypt: I picked this up in the Security Now! newsgroup at GRC. Someone posted, actually Dave DeBruce, last Thursday, October 1st, he wrote in a posting in GRC's newsgroup, "I've been a long time TrueCrypt user, but because of SN-527 [last week] I decided to give VeraCrypt a try." And just to remind people, what was found was a weakness in all of the TrueCrypt and TrueCrypt-derived full-disk encryption systems, which in the case of TrueCrypt will never be patched, and in the case of VeraCrypt had been patched with v1.15. And so my advice last week was it's not house burning down, but we know now that there is a problem in TrueCrypt, took us 16 months to find it, but we have, so time to move away from TrueCrypt.

So he says: "I'm on Win 7 Pro 64-bit. It installed fine," meaning VeraCrypt because he decided to give VeraCrypt a try. "It installed fine alongside TrueCrypt as they do not bump into each other at all. It was able to mount my TrueCrypt volume fine. I mounted both a new .hc volume and my old .tc volume." Well, now, "tc" is clearly TrueCrypt. I don't know why VeraCrypt would be "hc," but that's what he wrote. "Copied all over to VeraCrypt and thought all was fine, but there are issues. With 1.15, you can delete files from the volume, but not directories. You get an error saying the drive letter cannot be found. I'm sure this will be fixed, but it looks like I stay with TrueCrypt for now. This is a known and reported issue to the VeraCrypt folks, but it is a big enough issue where I cannot use it like this. Anyone else try this and have any issues?"

And a lot of dialogue ensued on the thread with people both being able to reproduce it and not. But there were links to forums that VeraCrypt hosts where other people are having problems and other problems in addition. So the sense was this was rushed out, perhaps, and did not receive the testing it needs. So I just wanted to let people know that maybe, depending upon how you feel about the vulnerability that is now found, the privilege escalation vulnerability in TrueCrypt, that you may want to at least watch for VeraCrypt to get these things resolved and for it to get stable.

And it's interesting that it said that the drive letter could not be found because that was the - we were never told exactly what the problem was, but it was about the - it was the drive letter handling of the TrueCrypt kernel driver where the glitch was found. So it makes sense that VeraCrypt changed something that broke something else when they were fixing the known problem that existed.

Just wanted to note that F-Secure, we've spoken of them often, they're a frontline security research firm, F-Secure jumped into the iOS adblocking game. They've got something called F-Secure Adblocker. And it's in iTunes. Some of our listeners have tried it. They reported, unfortunately, that there is no whitelisting option. So I think that pretty much rules it out. Either they'll have to add that, or it'll fall by the wayside.

But I just wanted to note that, you know, the problem with this, and this was expected, is that it is trivial to create an adblocker because now there's an API. There are well-known, publicly available blocking lists of domains. So it's just an afternoon's work for somebody who knows iOS app development to whip one out. Unfortunately, I don't think many of them are going to end up getting much traction or surviving, and in the meantime we'll have a lot of them. So it's sort of a nonevent, but this highlights that.

Leo: Incidentally, we're going to interview the chief research officer for F-Secure. I know you know Mikko Hypponen. He's going to be on Triangulation...

Steve: Oh, yeah, yeah.

Leo: ...a week from Monday. I'm really looking forward to it.

Steve: Cool. Very neat.

Leo: And Paul Syverson, the inventor of Tor, is coming up. I think he's going to be on The New Screen Savers this week and a Triangulation in weeks to come.

Steve: Nice.

Leo: So, yeah, we're trying to get a lot of these security guys on. In fact, you're welcome to join us. If you want to be part of the interview, we'd love to have you.

Steve: Yeah. I'll [crosstalk] watching, yeah.

Leo: Monday morning. Okay. All right.

Steve: Okay. So I don't get this one. VeriSign has launched a free public DNS service. And they - it's like, what? But sure enough, they now offer a pair of DNS servers. For anyone who's interested, I know our listeners like to mess with this kind of stuff, the IPs are 64.6.64.6 and 64.6.65.6. I thought I was mispronouncing it, but not. So 64.6.64.6 and 64.6.65.6. This is often done because best practices suggest that redundant servers be on different subnets. And even though these may be on the same subnet, they still, you know, that's why it's not .6 and .7, for example. Instead, it's 64.6 and 65.6, to sort of look like they are further away in IP space from being adjacent. And who knows? Maybe they are in different buildings or something. That would be nice.

Anyway, I don't get it because they say in their own FAQ, "Why choose VeriSign public DNS?" And so they said, "Stability: Confidence in a highly reliable public DNS platform. Security: Robust protection from security flaws. And Privacy: Assurance that your public DNS data will not be sold to third parties." So, I mean, those are not bad things. But, you know, DNS is not private. DNS doesn't have the equivalent of HTTPS. So it's all in the clear. Even if all of your connections, for example, in a Starbucks, to pick on an often picked-on target, even if all of your browsing is encrypted over HTTPS, your DNS isn't.

DNS queries are UDP packets that have no encryption. And so, even if you were using a third-party DNS and not, as is often used, your own ISP's DNS servers, if your ISP had any interest at all in logging your DNS, they could certainly do that because all of your DNS traffic transits through them anyway. Everyone knows that I have moved to Cox, and I am absolutely delighted with the service that I have been getting from them. However, the default DNS servers had that annoying habit of routing me to a Cox page in the event of a typo or a DNS lookup failure. They bounce me to their page rather than returning a "domain not found" error.

The good news, because that is a problem for - actually, it causes a serious problem for some automated systems that don't understand, like, why they got an IP address, they start trying to talk to it, thinking it's what they asked for, but it's not. Cox offers a different pair that don't do that. And so I manually configured DNS just to override. But I did run my DNS benchmark, looking at Google's DNS, OpenDNS, however many hundreds of DNS servers that GRC's DNS Benchmark checks. And the Cox DNS was, not surprisingly, faster than all. Why? Because it's right here. It's more local to me. It's the closest DNS server, by definition, that I could have.

So I haven't run 64.6.64.6 and 64.6.65.6 through the benchmark. This only came up a day ago, so I don't know how VeriSign compares in performance. But I knew that this would be of interest to some of our listeners who, for whatever reason, might want to use a third party who is representing no logging, privacy, they're going to keep their security, their DNS servers up to date and keep them online. So, I mean, it's nice that VeriSign's done this, but it's like, okay, I just don't get it, really.

We already talked about how carriers, mobile carriers are generating more revenue from just the bulk of mobile ads, although actually it's not quite accurate. This was a posting over on Medium.com, where this guy analyzed it, and he found that consumers pay 16.6 times more in data costs than the top 50 news sites are generating in ad revenue. So those are the numbers; and if anyone wants the deep dig, I've got the link here in the show notes. And he does, I mean, a really thorough analysis that was impressive.

[medium.com/@robleathern/carriers-are-making-more-from-mobile-ads-than-publishers-are-d5d3c0827b39]

Oh, and we knew that the absolute requirement for adblockers was an easy whitelisting facility, which is why Purify was our choice for the easiest, least feature-rich, but it's from the guy who maintained the uBlock, not uBlock Origin, the original uBlock filter. We knew that this was going to happen. We're starting to see sites which are popping up a notice. And I got a tweet from a Christy Ramsey, who sent me a picture of one that he got at fossBytes.com. Leo, you might want to just go, F-O-S-S-B-Y-T-E-S dot com, with your adblocker on, and you can see the notice. The notice reads, in bold...

Leo: Ooh.

Steve: Yes. "Please consider reading this notice. We've found out that you are using Adblock Plus or some other adblocking software which is preventing the page from fully loading. We don't have any disturbing banner, Flash, animation, obnoxious sound, or popup ad. We do not implement these annoying types of ads. We need money to operate the site, and almost all of it comes from our online advertising. And currently we are running low on budget. Please add fossBytes.com to your adblocking whitelist, or disable your adblocking software."

Leo: Hmm. I'll do that, sure.

Steve: So, well, exactly. So this is what we expected. And...

Leo: So how can I - does this turn, on uBlock Origin, does this turn off globally, or just for that site?

Steve: No, it is sticky for that site.

Leo: Okay. So that's all you have to do.

Steve: So you do that. And then, if you go down and click that refresh arrow at the bottom...

Leo: Yeah, I see.

Steve: ...the site will come up, and it's all happy.

Leo: Oh, yeah, it loads up just fine. You know, I was noticing Pastebin is doing this, too.

Steve: Yeah. And we're going to see it. What's interesting is that exactly the same notice is appearing on different sites. And, I mean, not necessarily all. So they've got their own. But there is a third-party site...

Leo: Oh.

Steve: ...which is offering this adblock, what they're calling "adblocker busting service." And they are representing that they're going to make sure that the ads that are being hosted, if you then lower your blocker, are not going to be annoying. But they are allowing, they give sites the ability to block, to notify and allow, or to completely block, so that the user must whitelist and then refresh in order to proceed.

So, I mean, this has been - it's an ex-Googler whose business, he had like a web marketing business of some sort he sold for $400 million to Google, stayed there for a few years, and he saw the writing on the walls. And so he left and founded this company to bust the ad busters. And so the fact that we're seeing the same notice on multiple sites tipped me off to the fact that, okay, this is a third-party that is presenting this. Although obviously sites could detect it themselves. But once again, they're subbing that out in order to get this job done.

And I did, I brought up Firefox's network analysis. I flushed my cache. With blocking on, the site required 1.7MB of traffic. With blocking off, so that ads were displayed, it only went up to 2.2. So only about a 20 percent load increase, and I thought, well, okay, that's acceptable.

I already talked about Snowden. So I did want to take a minute. Someone tweeted me a picture of SpinRite's DynaStat running. This was Anthony Gladden, who sent me a note on Saturday, October 3rd, so a couple days ago. He tweeted: "SpinRite saving my ass..ets once again. Thank you."

And something's made me a little uncomfortable, that I did want to mention. And that is that I'm sensing that people are using SpinRite to recover from tragedy or to pull back from the brink. And it really is the case that, if you run it before your drives get to that state, it will fix them and prevent them from getting there. So my point is that, yes, and Leo, your description on The Tech Guy over the weekend was absolutely perfect, when you had a caller who found some other, it was some commander, an older version of like a...

Leo: It was total commander, it was Norton Commander, sort of, yeah.

Steve: Right. And he was able to get some recovery because it didn't just abort at the first error, but tried. Or he may have just gotten lucky. It may have been that the drive, you know, just gave him the sector. But you described, you know, that oftentimes just being determined - you know, now, as we know, SpinRite does do more than just ask a whole lot. If asking a whole lot doesn't work, then it's able to do partial sector recovery, which is completely unique, that is, it's able to read the data even in a sector that never is fully readable, which allows you oftentimes to get what you need.

But anyway, I just wanted to say that I'm a little uncomfortable. I feel like I've empowered people with something that they may rely on, like may over rely on. And I would hate it for people to not use SpinRite in a maintenance mode, and for it then to be relied on and fail to be able to help them. And I recognize that it's difficult with SpinRite 6 because drives have gotten so large that nobody scans a whole drive. You can't format one of these multi-terabyte drives. If you ever try to do a long format, it takes a week. So no one does. Everyone does the quick format, which just assumes that the whole drive is fine. And maybe someday we'll get out there and take a look around, but probably not. In the meantime, we're just going to stay out here on the edge, and put a directory and an OS and things there, and just assume the rest of the drive is fine.

So, and that also suggests that you don't have to run SpinRite over the entire drive. You could just run it for a few hours, for example. And, you know, the first 10% or so, and get all of the benefit on that portion of the drive. But anyway, so please, I'll be making it faster with 6.1, as everyone knows, which will be a free update for everyone. But consider that you've got this tool, and maybe run it when, you know, maybe when you're not going to be using your computer overnight or for the weekend or something. And don't rely on it too much. But I thank everybody for their support, who has purchased it.

Leo: It's not just a disaster recovery solution.

Steve: It's really...

Leo: And I say this all the time. It's a hard drive maintenance solution.

Steve: Right. And many people have purchased it preemptively, and it's so cool. You know, we hear from people who it's like, I got it three years ago, and I finally was able to use it to save my data.

Leo: Right, right.

Steve: So that's neat, too. But had he used it a year ago, it would have prevented there from being a problem a year later, rather than waiting for multiple years.

So Marshmallow, which is the most recent release, what is it, version...

Leo: Six.

Steve: Six? Six, v6 of Android has substantially tightened its security. There are a bunch of features that security-aware people will appreciate. For example, we finally, and I've seen some people say, like, belatedly, got boot verification, which for a mobile platform is crucial. So in adding this, they catch up with Chrome, I'm sorry, with the Chromebook OS because it's had it for a while. So Marshmallow implements boot verification to warn users that the core software may have been interfered with or corrupted during the boot process. So not necessarily malicious, but probably.

So there's three levels of warning. There's the yellow level of warning, where the bootloader finds that an unknown OS has been loaded. You get a yellow triangle, and it says, "Your device has loaded a different operating system." So it's like, okay. I guess you probably know that, but maybe not. So if you see that, and you're not expecting it, there's a problem.

Or orange level is the bootloader is not locked. Now, of course, that's a problem because if it's not locked, then it could be modified. So the message there, the orange triangle, says "Your device software cannot be checked for corruption. Please lock the bootloader."

And then finally, stepping again a next level in toward the boot process, the red triangle is the boot image is corrupted. And the message says, "Your device is corrupt. It cannot be trusted and may not work properly." So new features of Marshmallow, and definitely beginning to work on locking down Android more than we've ever had before. And so that's just all good news.

Leo: Presumably you could turn that off because unlocking the bootloader of course is the first step towards putting your own ROMs on there. Which is not always a malicious activity. It's something a lot of Android users like to do. And I'm sure that there's a way to disable that in the default or options or whatever.

Steve: Yes.

Leo: I'm actually going to install it and let you know.

Steve: Yes. And I'm sure that their intention is for the bulk of users who just, you know, casual users.

Leo: Right.

Steve: Also they've added more control and visibility into app permissions. Users can now agree to app permissions as they are needed, rather than as just a monolithic list when the software is installed. And they now provide the ability to examine all the apps which have been granted a given permission. So, for example, you're able to look at the permission and see which apps have access to that on a permission-by-permission basis. It is necessary, however, that apps be modified to support this in new APIs which have been added at v6.

So don't expect to see it soon. But, you know, this is always the way these things happen is, you know, the apps can't do it until the OS offers it. And so the OS offers it, and then it's not supported until the apps catch up. So but this is great that, instead of just having to say yes to a whole block of things when you install something - you know, I remember one of the StageFright apps did that. And they modified it afterwards. I think it might have been Zimperium's because one of the features later was we're not asking for all these permissions. Because people were saying, wait a minute, why is this thing that wants to check for StageFright needing this block of permissions that sort of seemed unrelated to that. So this is all good.

They're also beginning to move towards a LastPass-style password store. Google calls it Smart Lock. It allows the passwords of apps and websites to be saved on the user's Google account. When disabled, no passwords will be saved or returned from the user's account. And as is the case with the granularity of permissions, again, it requires that a new API now available be supported. So it'll take a while for apps to get updated. But this is Google's first step into creating an Android Google-native password store, you know, safe.

And Marshmallow returns to full drive, or full disk, as they call it, FDE, Full Disk Encryption, enabled by default. We talked about it on this podcast back when it was causing problems for earlier Nexus users who found the performance hit unacceptable. And what was surprising is that the hardware built into the ARM core was not being used. They were doing the encryption in software, thus the expected overhead of this encryption being present. And as a consequence, many people were turning it off because this thing just slowed down their devices too much.

Well, with Marshmallow full drive encryption, the necessity - remember that it has to happen on the fly. Everything you write needs to run through a cipher on the way to store and through it again on the way back. That's now down in the hardware, where it should have always been. And so it's turned back on by default. So it's there. And I don't know what happens when you upgrade, Leo. You probably need to turn it on manually so it can run through and do an encryption pass over your device. But the underlying hardware has been supporting it for quite a while. Now we've got it in software. And it'll just be on for new Nexus 6 devices.

Leo: Yeah, I mean, the Nexus 6 I have here, in fact, took some hits in the press because its hard drive or its storage is very slow because it's encrypted by default. And in fact you can't disable it.

Steve: Oh, no kidding. I thought you were able to back out of it.

Leo: No, no.

Steve: Ohhh.

Leo: You can, but you have to root it. You have to start doing some strange things to it.

Steve: Okay. So you should be able - so the point is, when you update that to Marshmallow, you ought to see a performance jump.

Leo: Because I can turn off encryption.

Steve: What?

Leo: We're not talking about encryption?

Steve: Oh, yeah, we are. But, no, it's because it's now in hardware.

Leo: Oh, I see what you're saying. Yeah, yeah, yeah.

Steve: Marshmallow does the encryption in hardware. So presumably...

Leo: So it'll stay encrypted, but it should be a lot faster to access the drive.

Steve: It ought to have...

Leo: If this has the hardware in it. Now, the 6P I'm sure will. I don't know, this is the old 6. I don't know if it will or not.

Steve: I think it does. What I remember...

Leo: Well, that would be funny, that they didn't turn it on. That's annoying.

Steve: That was what was so mystifying, and we talked about it on this podcast months ago, is that the ARM core has the hardware, but the Android wasn't using it.

Leo: Okay. Got it, got it. Okay.

Steve: And so now they are using it. And so I think you're going to see a jump in performance when you move to Marshmallow, which would be great.

Leo: We'll see, yeah.

Steve: I wasn't familiar with the way VPN configuration was happening before. But they've also surfaced that at the UI. Under Settings > More > VPN, you now have the ability to configure VPN settings and usage, which is handy for the bring-your-own-device approach, where a corporation wants you to use your device, but their VPN. So this allows you flexibility in jumping around among VPN services. You know, you could do that in the app settings per app, but not globally. And again, I don't know whether this requires API changes. Sounds like it would. But there should be a Settings > More > VPN page to give you control.

And with what they're calling in the Nexus phones, at least, they're branding as Nexus Imprint, we've got integrated fingerprint authentication. So before this, individual phone makers had to integrate fingerprint support themselves. And of course we heard reports about this not being secure and not being done the way we wished it had been done. Google responded with, in Marshmallow, offering a new fingerprint authentication API. So it will be up to apps to support that, but it exists now, and they'll be able to rely on it in anything using v6 of Android. So that's great. It will allow users to lock and unlock their devices with a finger scan and will hopefully encourage more vendors to support them. I know you've become as big a fan of fingerprint scanning, Leo, as I have always been.

Leo: Oh, absolutely.

Steve: I mean, it's just - it's the way for a personal device like this to easily and repetitively reassert your identity. So fabulous that it's now in the OS. That's where it needs to be.

They've also integrated app-level backup, so that individual apps will be able to use OS features to have Google store up to 25MB of individual app settings data without having to implement that themselves. So again, a nice useful feature, the idea being, for example, if you uninstall and reinstall an app, the app, upon reinstallation, would be able to check and find that, oh, look, there's a bunch of settings data, and then be able to suck it back down in order to essentially reinstantiate its previous settings data.

Also, though not technically a security feature, there is now voice control available without unlocking the phone. So you can do things from the locked screen without essentially dropping your security in order to unlock the phone. So that could be some benefit. And also I like the fact that they are very clearly showing the Android security patch level. There's a page, I should have taken a snapshot of it because I liked just how simple it was, where it simply says the date of the last security update. And I think in the picture I saw it was, like, October 1st, 2015 it was showing. And so the regular patching applies to all Google-controlled Nexus devices.

But what I wrote or what I read about this, it said so far the Android M devices are the only ones that show this information. In the future, with Marshmallow, all devices should make it very clear how up-to-date the security patches are so that anyone can quickly check, when they hear like something has happened, oh, here's the date of the latest release. They can see whether they have it or not. And lastly, they've made, along with integrating encryption in the hardware, they extend that into SD cards. And they call it "flexible secure storage."

So sort of following on from the encryption, which is now on by default, Android M devices will automatically extend App Store onto SD cards without it having to be done manually. And that encryption, that extended encryption, will also be encrypted. So while not technically again a security feature, it is increasing the security of using add-on storage for additional storage space on devices. So a very impressive roundup of security improvements in v6. And I'm really pleased by all of them. They're like, they're what we want to see.

Leo: Well, as I said, I'm installing it on my Nexus 6. And of course it will come on my Nexus 6P, so I look forward to that.

Steve: Right, right.

Leo: I have a Nexus 7, too.

Steve: Okay. Finally, my last little bit I wanted to talk about is - and I teased this last week. Nadim Kobeissi, whom we've spoken of a number of times, he is the neat guy that did Cryptocat, which was the web-based implementation of the OTP protocol to do very strong, well-vetted online chat encryption on a browser-based solution. He posted something on October 25th on his own blog. He has a site, Nadim, N-A-D-I-M, dot computer, nice domain, that caught my attention. I thought it was really interesting because essentially my own retitling is "What the VW discovery means for the encryption software industry."

And what Nadim observed is that what we had with the behavior that we discovered with VW was that unexamined, invisible, legally unexaminable software had essentially conditional behavior. It was able to change its behavior based on circumstances, or I'll use the word "context" because what I've come away from with sort of extending this is what I would call "context-aware security," CAS, Context Aware Security, the idea being that security could become a function of its context of use.

And here's an example. We know the way TLS connections are established to create an HTTPS tunnel between a client and server. The first thing the client does is generate a "client hello" containing a client random blob, which is part of the key exchange. And we've often talked about how the quality of the random number is excruciatingly important. The numbers generated have to be high quality because, failing that, it makes attacks far more practical.

So imagine a security stack running, that was made widely available, where no one vetted it, no one looked at it, it wasn't open source; and it knew, just for example, the IP range for the networks in China. And when this particular client is initiating a TLS connection with a server in China, because it uses context-aware security, it doesn't choose a very good random number. Nobody would ever know that, in the same way that nobody knew there were all these VWs buzzing around, spitting out 40 times the NO gas that they should, because whenever anyone looked at them they seemed fine.

Similarly, if security researchers in, for example, the U.S., checked the behavior of this security stack by, for example, initiating lots of connections, you know, they wouldn't have any idea that this was context-aware security. So they could initiate connections, verify that the "client hello" packet contains a very good random number, and it would never occur to them that, well, yes, because their IP is in the good area, it's not a Chinese IP. But this context-aware security changes its behavior subtly when it's in a different context.

And anyway, this essentially was the point that Nadim noted, and which I have expanded on, that I thought was very interesting because, I mean, this is a problem that we now have to assume will get exploited, or a capability, anytime there is no visibility. The reason this happened, even with Bosch warning VW that this could not be taken out, you know, you can't drive cars that have this modified code that you asked for and we provided you. And VW said yes, yes, yes, we know.

The problem was that the manufacturers were all saying, oh, no, this is all proprietary. And it's of course DMCA protected, blah blah blah. Just trust us. Well, closed source security products could behave differently under any circumstances they chose. And just like the VWs, we would never know. And so I thought that was a really, really salient point that Nadim had brought up, and I wanted to share it with our listeners.

Leo: Nice. And there you go. There you have it. We are complete.

Steve: Yes. It wasn't three hours long, but it was an hour and 45 minutes long.

Leo: We are up to date.

Steve: And I'm exhausted, so...

Leo: Everything you need to know, right there. Steve Gibson is at GRC.com. That's his place for everything you hear about, including SpinRite, the world's best hard drive maintenance and recovery utility. So go there and get it. That's Steve's bread and butter. You support him by doing that. He also has the show there, 16Kb versions, as well as 64Kb audio, MP3 audio, as well as written transcriptions. It's all at GRC.com. Questions can be left there, GRC.com/feedback, or leave them for Steve on Twitter. He's @SGgrc.

Steve: You know, I was thinking I need a slogan. You've got that - I can't get it out of my head - "Takes a bending and keeps on sending" for your fiber optics [crosstalk]...

Leo: You need something for SpinRite, huh?

Steve: How about "Takes a spinning and keeps on singing"? That's not quite the same. But, you know, yeah, we need a slogan, a SpinRite slogan.

Leo: "Get your bits right with SpinRite." I don't know. We'll come up with something.

Steve: But I tell you, "It takes a bending and keeps on sending." Oh.

Leo: We also have the show on our website, TWiT.tv/sn, and wherever podcasts are aggregated. The best thing to do is be subscribed. That way you get every episode ever released. Well, from now on, anyway. You can go back in time on the website, as well. And I keep waiting for somebody to use the API to write a Steve Gibson downloader that'll grab all the shows. You could easily do that through the API. Well, maybe somebody will.

Steve: Boy, I sure get a lot of requests for, like, an RSS feed or some way. "Do I have to download each one of these, you know, manually?" Like, well...

Leo: Be a very big RSS feed.

Steve: Ooh, baby.

Leo: I think the best way to do this is a one - I think you could do it in a one-liner with a loop calling the API and just successively going through episode after episode.

MARK RICHEY: One line of cURL.

Leo: One - it's a cURL. A line of - actually, you wouldn't even need the API since we have a consistent naming protocol.

Steve: Yes. Please download the full bandwidth versions because otherwise my bandwidth gets slammed.

Leo: Don't suck Steve's bandwidth, suck ours. And I do think it's a consistent naming all the way through. So starting with TWiT.tv/sn, is it 001?

Steve: Yeah, you've got four digits.

Leo: So it's 0001.

Steve: In your URL.

Leo: That means we thought we'd have 10,000 episodes.

Steve: Yeah. I'm using three digits, so this game is over at 999.

Leo: Yeah. So it's, oh, no, that doesn't find it. It says it's 404. Let's try 0002. So maybe it's just episode - no, 0002 doesn't work either. So maybe, oh, remember that - I know what happened. Yeah. We'll have to, well, there's got to be a way to cURL it.

MARK: Yeah. So that's four lines of cURL.

Leo: Yeah. Yeah, it's an easy thing. Easy enough. Mark Richey is here. He's going to write that for us. Now you're on the spot, Mark. You're right, it should be easy.

MARK: cURL has the option [indiscernible], as well.

Leo: Yeah, yeah.

MARK: So [indiscernible] options to fill in all the numbers and download the podcasts.

Leo: Yup, loop, yup. Be a snap.

Steve: And lord knows how many hours of this one will then receive.

Leo: Oooh.

Steve: But there are a lot of people who want the whole archive, so I, you know...

Leo: Reverend Dan says we went to four digits late - I don't know. I'll have to, you know what, it really isn't the website that you have to care about, it's CacheFly URLs. So...

Steve: Yes. And in fact my server is intelligent. And because I used a uniform representation and the redirection code, it bounces through Podtrac and then CacheFly. It knows how to convert them so they're all there. So whatever its logic, I don't remember now what logic I had designed.

Leo: We've got quite a few shows.

Steve: Yes.

Leo: 528.

Steve: Takes a bending and keeps on sending.

Leo: Oh, lord. Oh, lord.

Steve: Okay, my friend.

Leo: Thank you, my friend. It's great to have you here, as always. And next week maybe a Q&A episode, if there's no other big news.

Steve: Oh, after this week of breaches and vigilante worms and things, yeah, hopefully it'll quiet down, we'll do a Q&A, and have another great podcast for everyone.

Leo: Yeah. Watch out for the vigilante worms. We'll see you next time on Security Now!.

Steve: Bye.