More News and Feedback

Description: This week we discuss the long-awaited end of StartCom & StartSSL, inside last week's macOS passwordless root account access and problems with Apple's patches, the question of Apple allowing 3D facial data access to apps, Facebook's new and controversial use of camera images, in-the-wild exploitation of one of last month's patched Windows vulnerabilities, an annoying evolution in browser-based cryptocurrency mining, exploitation of Unicode in email headers, Google's advancing protection for Android users, a terrific list of authentication dongle-supporting sites and services, Mirai finds another 100,000 exposed ZyXEL routers, Google moves to reduce system crashes, a bit of miscellany including another security-related Humble Bundle offering, and some closing-the-loop feedback from our terrific listeners.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-640.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-640-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Lots to talk about including the demise, long-awaited demise of StartCom; another 100,000 exposed routers to the Mirai botnet; and, already, an in-the-wild exploitation of one of the Windows vulnerabilities patched last month. Plus the latest on cryptocurrency mining. Ooh, boy. It's all next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 640, recorded Tuesday, December 5th, 2017: More News and Feedback.

It's time for Security Now!, the show where we cover the latest security news, help you protect yourself and your loved ones online, and sometimes laugh, laugh ha ha, at the futility.

Steve Gibson: Of even thinking there's hope.

Leo: Mr. Steve Gibson is here from GRC.com. First, thanks to Robert Ballecer. Father Robert filled in last week for me. I ran out of the studio so we could go to New York for a couple of days; and he did, as always, a great job. But I'm back in the saddle.

Steve: Glad to have you, yup.

Leo: Yeah. And if I had a saddle, I'd be back in it.

Steve: Well, and we don't have any big topic, as we didn't also last week. Nothing really stood out. There were a couple themes. I am going to talk about, and I had gotten ready to talk about, the ElcomSoft, what I would have to call sort of a takedown of what they feel are some tradeoffs that Apple has made in the interest of convenience that has really hurt iOS 11 security.

Leo: Oh, I'm so glad to hear you say that because I was waiting for your analysis of this. When we talked about it on Sunday on TWiT, the panel seemed to think, well, ElcomSoft makes money protecting people, and so it's in their interest to kind of spin it negatively. But Rene Ritchie today on MacBreak Weekly said, no, they're right. So I want to hear what you have to say.

Steve: Yes. And we don't do that today, unfortunately, because you know me, I want to absolutely have all my ducks in a row.

Leo: You bet.

Steve: And a few of them wandered off. So I will get them back lined up, and I'm sure we'll talk about it next week.

Leo: Well, that's one of the reasons we value you, Steve, is you always make sure that you spend the energy to get it right the first time out.

Steve: Well, and it's not as if we don't have enough to talk about. We're going to talk about the long-awaited end of StartCom and their StartSSL certs.

Leo: Oh, boy.

Steve: And also, as I'm sure you did in the previous hour, take a look inside last week's macOS passwordless root account access bug.

Leo: Another one I wanted to ask you about, yeah.

Steve: And the subsequent, yes, and the subsequent problems with Apple's patches of that, which, I mean, it's like, oh, boy, did they sort of stumble over that, yeah. Also an interesting question. I'm not as alarmed, and I know you won't be, as I've seen some of the coverage of the announcement of Apple's intention to allow apps to access the 3D facial data in the iPhone X, which is interesting. Also we have another example of what I heard you talking about already on the TWiT network, I'm not sure which podcast it was, of Facebook's new and controversial use of camera images. You guys were talking about Facebook asking people to upload nudes of themselves.

Leo: Yeah, yeah.

Steve: And there's even more. So we'll talk about that. Also, in-the-wild exploitation of last month's Patch Tuesday, one of the 60-some vulnerabilities - or, no, I think it was 53. It was 60-some over on the Adobe side that were patched. An annoying evolution in browser-based cryptocurrency mining. Exploitation of Unicode in email headers, who could have not seen that coming? Google's advancing protection, which is welcome, of Android users. A terrific list of authentication dongle-supporting sites and services. Mirai finds another 100,000 newly attackable routers. Google moving to reduce system crashes that are a consequence of what's happening with Chrome. Some miscellany, including another security-related Humble Bundle offering that will be certainly of interest to our listeners, and some feedback from those same listeners. So as you can see, I don't think we're going to run out of things to talk about.

Leo: We're full of good stuff today. And of course don't forget our Picture of the Week.

Steve: Oh, yes.

Leo: It's coming up. All right.

Steve: So our Picture of the Week follows on from one that we had three weeks ago that you'll remember, where the zebra was defragged, and so all of the zebra stripes were pushed to one end, and the zebra was now 50/50 black and white. So this following on from that, we have your standard-looking zebra who's busily chewing some grass, who is being confronted by a zebra with a QR code instead of stripes. And the zebra with the QR code is saying to the one with the stripes, "Upgrade, man." Meaning of course that the normal zebra is a barcode, which is old school, and new zebra has QR code. So thanks to one of our listeners for sending this to me.

Leo: Love it.

Steve: Just kind of fun. Geeky fun.

Leo: Upgrade, man. Yeah.

Steve: So StartCom is this Chinese-based certificate authority, a CA based in Beijing. And we've sort of been following their rocky history for a while. To remind people, they're the people who were behind the StartSSL certificates. They opened branches over time in China, Hong Kong, the U.K., and Spain. Then they began having problems because, as we've often said, there is some serious responsibility that comes with a certificate authority's inherent ability to basically print money, which is what happens when you hand out bits and people pay you for them because they're special bits.

But with those special bits comes some serious obligation. And if you mess that up,

because our entire public key ecosystem is based on trust, if that trust is broken, then your ability to charge people for your bits goes away. So we've seen multiple failures on the company's end over time. All StartCom certificates were removed from Firefox in October of 2016. Then Google's Chrome browser followed suit March of this year, March of 2017. It was then discovered, due to some website shenanigans which we discussed at the time, that StartCom had been secretly acquired by WoSign in Shenzhen, China, through multiple companies. I mean, and it seemed really suspicious. And remember that someone discovered that the WoSign, I guess it was the StartCom site was secretly using WoSign on the backend, and then it was figured out that WoSign had acquired StartCom. So there was no disclosure of this. Not disclosing that is a violation of the CAB guidelines, the CA Browser guidelines. So this was just really making everybody feel uncomfortable. There were, of course, misissuance problems.

Finally Mozilla and Apple sanctioned the company. StartCom announced it would be restructured during 2016 by the WoSign parent, which was the Qihoo 360 Group. They detached StartCom from WoSign that was having its own problems to make it its own subsidiary. Anyway, the last time we checked in, I remember talking about how weird StartCom's website was. And I remember, Leo, you sort of agreeing with me on this because it looked like a fire sale. And it's even up there now. If you go to StartCom.org, it's "UNLIMITED number," UNLIMITED all caps, "UNLIMITED number of 2-year EV SSL Certificates for FREE - up to 99 domains," which pretty much ought to cover anybody's needs.

Leo: But don't do it; right? I mean, well, you'll see why in a second.

Steve: Oh, yeah, they're worthless.

Leo: Yeah.

Steve: And also an "UNLIMITED number of 2-year OV SSL Certificates for Free, with multidomain and wildcard." And an "UNLIMITED number of 3-year OV Client Certificates for FREE," or a "3-year Code Signing Certificate, including kernel signing." Sounds great. And actually those offers were there last time we looked, quite some time ago. So finally, last week, or two weeks ago, on November 16th, StartCom announced its termination of business, which comes as no surprise to anybody except, I guess, somebody - I don't know if you've even been able to pay them for anything recently because they seem to be giving away everything for free.

So they said: "StartCom has played a critical role" - it in fact gives "critical" a new meaning - "as a Certification Authority in data security and electronic commerce by providing an independent 'trusted third party' guarantee all these years. Around a year ago," they write - and a whole bunch of our listeners forwarded me the email that they sent out, which is what I'm reading. "Around a year ago, the majority of the browser makers decided to distrust StartCom..."

Leo: Out of nowhere. We couldn't understand why. What happened?

Steve: Yeah, we don't know what happened - "...remove the StartCom root certificates from their root stores and not accept newly end entity certificates" - which is their term

for server certificates, the end of the certificate chain - "end entity certificates issued by StartCom. Despite the efforts made during this time by StartCom, up to now there has not been any clear indication" - quite the opposite, in fact - "from the browsers that StartCom would be able to regain the trust. Therefore" - yes, the trust has been lost - "the owners of StartCom have decided to terminate StartCom as a Certification Authority (CA)."

Leo: Although I notice on this very same page they are still offering certs.

Steve: Yeah, well, you know, what have you got now, a week?

Leo: We'll take your money.

Steve: Two weeks, maybe?

Leo: Yeah, till January 1st, yeah.

Steve: "From January 1st, 2018, StartCom will not issue any new end entity certificate" - because really, now, what's the point? - "and will only provide validation services through its OCSP and CRL" - that's the two types of certificate revocation. So it's nice that they're maintaining those for two years starting January 1st, 2018. And then starting 2020, all remaining valid certificates will be revoked. Of course, they won't be trusted anyway, so it doesn't really matter whether they're revoked or not because no one is going to care. StartCom, however, they conclude, "wants to thank all our customers and their partners during these years for their support." So it ends. And we've been covering...

Leo: Steve, do you think they were up to no good, or just inept?

Steve: I don't think this was malicious.

Leo: They were just inept.

Steve: Yeah. And, I mean, even VeriSign, I mean Symantec, who is much more prominent and a major CA, as we know, they're selling their assets to DigiCert because they, too, arguably, are responsible for the behavior of their subsidiaries, and they allowed a subsidiary to make mistakes. And so it doesn't matter whether you're some obscure CA in Beijing or you're Symantec, who purchased VeriSign's business. You really do have to behave yourself.

And, I mean, the good news is that, as we know, there's been an explosion of certificate authorities in all of our root stores, meaning that everybody trusts everybody. So it's only the threat of this happening. I mean, and this is bad. I mean, Symantec's not happy to be selling off their CA business, but they don't have a choice because they're not going to be trusted any longer. And similarly, StartCom, they were printing money, but they screwed up.

Leo: Who is their root certificate holder? Was it the Hong Kong Post Office? Do we know?

Steve: I don't know who...

Leo: Because don't they have to have...

Steve: No, they don't. They were a root.

Leo: Oh, they were a root. Oh.

Steve: Yes, yeah. So they did not have anybody signing for them. The only reason you need to have someone signing for you is, for example, when Google was bootstrapping itself, I think they had someone like Global Cert or one of the old...

Leo: Somebody that's already in the browser certificate list, yeah.

Steve: Right, right. And Let's Encrypt did the same thing. Their certificates were cross-signed. They signed them, and they had somebody else sign them. And then once their own roots propagate out into all of the roots, then they no longer need to be cross-signed.

Leo: That makes sense because their root could have yanked it. But since they don't have a root, the browsers have to yank their certification. Okay.

Steve: Right, right, right. So anyway, so it goes. And I think so it should go because this, I mean, all the other CAs are watching this and thinking, oh, there but by the grace of our IT staff go us. So they don't want to go there either because - and this is what browsers have to do. Otherwise it's nothing. The system falls apart.

Leo: It's proof that there's integrity in the system. "Strengths" in our chatroom says: "Any sufficiently advanced incompetence is indistinguishable from malice."

Steve: Yes.

Leo: Nice paraphrase there. I like it.

Steve: Yes. Okay. So last week, right as we were starting to record this podcast, the news of this seemingly horrific - well, and more than seemingly, but we wanted to be careful - this horrific flaw in macOS High Sierra came out. And it was like, what? Our listeners who remember last week will remember that the news was you could simply

declare yourself root as your username and leave the password field empty; try it not once, but twice; and be able to log into any High Sierra system with full privileges. And, yes, it even worked remotely. So it was like, holy crap. Apple jumped on this, of course, and attempted to fix it. I say "attempted" because, as we'll discuss here in a second, that was a little bit of a fumble, as well.

But Patrick Wardle, who is the Chief Security Researcher of R&D at Synack, who maintains his own site, Objective-See, S-E-E, is just interested in security. He's a macOS guy. He describes himself in his site. He says: "As Macs become more prevalent, so does macOS malware. Unfortunately, current Mac security and antivirus software is fairly trivial to generically bypass. Objective-See [his site] was created to provide simple, yet effective OS X [and now macOS] security tools. Always free of charge, no strings attached. I created Objective-See to publish the personal tools that I've created to secure my own Mac computer."

So he plowed into this and figured out, I mean, by reverse-engineering the OS, what exactly it was that happened. It turns out that there was a function, "od_verify_crypt_password." So od_verify_crypt_password, which is the function you call to have the system stored authenticated password hash compared with a newly created hash of the password the user has just provided. So it compares the hashes.

Well, it turns out that that function returns its success on its ability to compare the hashes and, separately, whether they actually compared. So the original code in 10.13 and 10.13.1, it reported, yes, I was able to determine whether the hashes compare. And the code was using its declaration of its ability to successfully compare them, rather than whether or not they compared.

Leo: Well, that makes sense. That would be an easy mistake to make, I think, to say, oh, I got success, and not understand that the success wasn't what you thought it was.

Steve: Correct. So in a different...

Leo: That kind of makes sense now. I get it, yeah.

Steve: Yes. In a different register, it was returning the validity of the comparison. So its own return, the function's own return said the other register's statement of comparison is valid. And so the original code forgot to check the register which contained the validity. Instead, it was checking whether that register value was valid. Which it always was, whether or not the password hashes matched or not.

So what was happening was, if the account was disabled - that is, the root account is not normally enabled at all. It's offline. It's disabled. But as a convenience, if you attempt to use it, and it's disabled, it will enable it with the provided password. That's why it took two times. The first time woke it up. It woke up the root account and gave it a null password. So then it was ready, but it didn't work the first time. So then you just did it again. And now the root account has been wakened up, and it's been given the hash of a null password. So when you give it another null password, or anything else, it doesn't even matter if the passwords match because this function says, yes, I have successfully determined whether they match.

Leo: Right. But it would normally do...

Steve: But not that they matched.

Leo: It had done in previous versions, before it would then allow you to log in, it would say, all right, well, but now you have to provide a password. And it just skipped that step.

Steve: Correct, correct. So then the little fumble here is that, as you no doubt know, Leo, patches were issued for both of the two current OSes 10.13, many of which were still out in the wild, which had not been yet updated to 10.13.1. So that fixed everything. The problem was that what was then discovered, first of all, was that the patch for 10.13.1 broke remote filesharing for some users. So that turned out to need a little bit of a fix. And there was a support article, and it didn't require too much. You had to use a terminal command and issue some magic incantations, but then that was okay. What was worse was that, if a system, if a 10.13 system was then upgraded to 10.13.1, the patch to that had not been applied to 10.13.1, so it regressed you back to the original vulnerability so that your post-patched updated 10.13.1 was now again vulnerable. Unless you reapplied the patch, this null root login patch for 10.13.1. And even if...

Leo: And rebooted.

Steve: Yes. Yes, exactly.

Leo: Yeah, the one little thing, yeah.

Steve: And even if you did, it didn't tell you that you needed to restart the system, and it didn't make you do that.

Leo: Whoops, yeah.

Steve: So you were still vulnerable. Yikes. So, boy. They got it. They figured that out, and the word got out, and they stumbled a bit more than we wanted. But that's what was going on is that there was a function which said "I successfully determined whether you entered the proper password." But separately, there was whether or not you entered the proper password that the code wasn't checking.

Leo: Success can mean different things to different people.

Steve: Indeed it can.

Leo: That's the moral of the story.

Steve: So the Washington Post got all freaked out over the fact that Apple has announced, and apparently it's already available because I saw a sample app that's doing this. In fact, Leo, there's a cool-looking app called MeasureKit, which is available for iOS 11. If you look at MeasureKit.com, you can see, I think on his site he shows you a bunch of samples of this thing. It's a cool augmented reality app that allows you to point at the wall at pictures and determine whether they're level or not. You can use it to walk around and do really cool augmented reality measuring stuff. It's free with, I guess, some things not enabled, or three bucks if you want all the features enabled. But measure door frames just by moving this little thing around. It's really cool-looking.

One of the things that the author of this has said he will be supporting is this availability of the Face ID wireframe, the whole 3D modeling of the face. So this is what's got the Washington Post guy - and the author of the Washington Post article managed to find a whole bunch of other people who are running around in circles screaming that this was the end of life as we know it on Earth. It's like, eh, okay. All I'm wanting to do with our listeners is to note that this is a thing. That is, that an app can say they want access to your camera, and that's all apparently they have to do. They then get access to the camera.

What this now means in an iPhone X world is that all of the 3D facial mapping, this 30,000 points of 3D grid will then be available to apps. So they can watch you raise your eyebrows, make a frowny face if you don't like something that the app just showed you, I mean, who knows. Maybe it's going to create some cool features. It's worth knowing that, if someone is concerned about privacy, maybe this is an issue. I think we sort of have to see how that goes. If it becomes a problem, I imagine Apple will make it more clear, maybe make apps more explicitly say that this is what they want to do, or explain why they want it, or maybe make that expire or time out or who knows.

But anyway, I read through this huffing and puffing thinking, you know, okay, maybe. But I guess if we were to compare it to the - it's different than Touch ID was because Apple never exported any aspect of Touch ID. Apps could get a go-no-go, and apps can certainly get that with Face ID. But now this is certainly much more than that. So this is Apple exposing the 3D mesh in real-time to the apps of the people who are looking at the phone as they're doing so. So I wouldn't be surprised if it ends up being used for a bunch of interesting things. And so we'll have to see if this evolves as a privacy issue.

Leo: It doesn't release the Face ID data points that are stored in the Secure Enclave. It only gives a developer access to the routines in the camera. So it's not that much different, it seems to me, than just giving a developer access to the camera.

Steve: Well, for example, someone's already stated that they will turn this into 3D printing data so you can print a mini-me of yourself.

Leo: But we were doing that before. You just had to move the camera around. So now they have that built-in face-mapping camera.

Steve: Right. Yeah, and the argument was that, because this is a big, like a unique

distinguishing feature that Apple has, maybe Apple is promoting it and saying, hey, we have this grid, this mesh, this 3D mesh data that we're going to let you have as an app running on the phone. So, yeah. It's not clear...

Leo: You'd want to make that available to developers. Although it's interesting because Apple didn't make NFC available to developers for the longest time, and that's a minor, compared to the face-mapping, that's a minor...

Steve: Well, and even Touch ID. Remember that was initially - we had to wait for that to get out beyond Apple's own use.

Leo: But, yeah, you could make that same point. I mean, my bank app allows me to identify, as does LastPass, with my fingerprint, and now my Face ID. But that...

Steve: Well, but that's...

Leo: But all they're getting from Apple is a yes; right?

Steve: Yes, correct, right. A go-no-go.

Leo: Right.

Steve: So anyway, we'll see if it ends up being useful. It'd be fun to see what developers do with it. Okay, so...

Leo: [Crosstalk] security flaw.

Steve: And on the topic of faces, Facebook is reportedly testing a new CAPTCHA system that requires users to upload a fresh clear photo of their face to prove the account is not fake and belongs to a real person. So as we understand it at this point, Facebook is being a little quiet about this because they don't want to provide too much detail to allow this to be bypassed. But at various points in Facebook's authentication flow, Facebook may require and is experimenting with requiring its users to aim their smartphone's camera at their face to provide a unique and never before seen image of the user.

These new tests were first spotted last week by someone in the U.S. who said that - oh, and our friends at Bleeping Computer picked this up, saying that "Facebook is now locking users out of account features" - this person was a little grumpy - "then requiring that those users verify their account to get back in by scanning an image of their face." So the text that is presented says: "Please upload a photo of yourself that clearly shows your face. We'll check it and then permanently delete it from our servers." So the new system is designed to catch and block the creation of fraudulent and automated accounts.

And when asked about this, Facebook confirmed that the new CAPTCHA system, well,

confirmed the existence of the new CAPTCHA system, saying that its role is to catch suspicious activity at various points of interaction on the site. So it's automated. It uses facial recognition and requires the use of unique photos to prevent bots from taking existing photos of random people off the Internet or from other Facebook accounts to authenticate. So for this to work, Facebook must feel that it has acquired a database literally from scraping the Internet and Facebook of all current photos. And so you show it something that you're saying isn't online, and Facebook's system says, yup, never seen that, never encountered that picture anywhere, so we think you're you. It's like, okay. So be interesting to see how this works.

And then you reported, Leo, on I don't remember which podcast, it might have been the podcast after - it might have been with Jeff and Steph that Facebook had been asking potential victims of revenge porn - I mean Stacey, yeah - of revenge porn to upload nude photos of themselves in anticipation of the possibility that those photos might get loose so that Facebook could proactively block them from being posted by anybody else. And the company said it would create a hash of the nude photo and use that hash to ban other people from uploading a bit-identical, because of course it has to be bit-identical for the hash to collide and have any value. And then Facebook initially said that the system was fully automatic, then later admitted that a Facebook employee would be manually reviewing all nude photo uploads.

Leo: Oh, lord. And by the way, I want to give credit to Jeff Jarvis because on the fly he said: "Why don't they just give people the app that generates the hash and let them upload the hash? Then no one has to see the nude photo." And I thought Jeff deserves an A+ in computer science for coming up with a better system than Facebook came up with. I hate - I'm starting to really hate Facebook. These guys are a problem, I'm starting to feel.

Steve: Yeah.

Leo: This is just terrible.

Steve: Yikes. So we'll see how that goes. We'll see how many people say, hey, here's me.

Leo: Here's my nude photos.

Steve: That I'm planning to release into the wild.

Leo: I'm worried, yeah.

Steve: So please proactively block it. How does this make any sense at all?

Leo: Well, especially since there's a better way that's completely private.

Steve: Yes. And Jeff, I listened to that, and I was...

Leo: Wasn't it impressive that he thought of that?

Steve: It was, yes. Jeff, you know, you are turning him into a techie, Leo, kicking and screaming all the way.

Leo: Yeah. Isn't that awesome? I was so impressed with that.

Steve: So we did have news that hackers have recently exploited a bug that we found out about officially on Patch Tuesday. And of course this is the problem that we're seeing now is that, because of the fact that many systems aren't being patched, the patches can be reverse-engineered in order to determine what it is they're patching, and then those vulnerabilities can be exploited until they're fixed. In this case, this was a juicy one. This was a problem in Microsoft Office which, going back to, well, going back 17 years - 17 years, like before there was the Internet almost - there was DDE, Dynamic Data Exchange, which was an early Windows system which allows applications to share data back and forth. It was sort of behind the scenes of drag and drop.

When you drag and drop something, you're actually sending a message to another app. And there's this DDE, Dynamic Data Exchange, which is a messaging system that allows one app to package up some memory and hand it off to another app. So because it was ever used, it is still present in Windows 10 Creators Update. I mean, like all Windows ever have had this, and all Office has had this in order to support drag and drop. That's one of the means for doing that. So there was always, has always been a bug, which came to light relatively recently and was just patched two weeks ago in the Patch Tuesday, November's Patch Tuesday, last month, where 53 vulnerabilities were patched. What's happened is that it was immediately jumped on by miscreants who figured out how to create a malicious RTF document having a .dot extension.

Leo: Damn those miscreants. I hate them.

Steve: Don't you hate them? They're just...

Leo: I hate miscreants.

Steve: Those pesky miscreants who used DDE, this still-present-everywhere Dynamic Data Exchange, to launch a PowerShell script which in turn runs a component of a pen-testing library known as Cobalt Strike to install malware into a user's machine after they open the malicious document. They don't have to click on anything or do anything. Opening the document is enough. And we've talked about those problems before. Here is another one. So it's not wormable, meaning that some user interaction is required. It can't just be received in email when we've run across those that can. This one can't be. But merely opening and observing a document attachment is enough.

So our takeaway is, so first of all, this has been patched. So you definitely want to make sure that you applied the November Patch Tuesday patches across systems, especially

that would have access to incoming email, phishing schemes and so forth. But also this forms yet another example of why everyone needs to always be cautious and suspicious when opening documents sent via email. That is, don't, unless you absolutely have no choice but to do so. So try to avoid doing that, if there's any way you can. This is fixed.

But the problem is, as we know, there's this window of opportunity between - especially when people, you know, right now we're seeing more people being annoyed at Windows trying to update them right in the middle of giving a presentation. When I was in Utah attending the DigiCert conference, one of the people opened their laptop, and I could see the screen, which was sitting - it was a Windows 10. It took, like, and I'm not exaggerating, half the day for him to have access to his machine. It was just sitting there spinning with the little rollercoaster dots running around.

Leo: He's very patient.

Steve: Oh, my lord. And I remember saying, oh, look, it's finally back. Yikes. Anyway, the point is people are tending to defer updating because it's so annoying, and it can take so much time, and they have other things to do like use their computer. In this case, fine, but don't read any email or open any docs until you know that your system is current. And in general the big takeaway is this problem has been there for 17 years. It allows bad stuff to get into your system and run if you just open a document that you receive in email. So now this problem is fixed. We know probably there are others. So the takeaway is don't allow this to have the opportunity of happening.

Leo: Wow. On we go with the newly caffeinated Steven "Tiberius" Gibson.

Steve: Yes, recaffeinated.

Leo: Recaffeinated.

Steve: So...

Leo: Did you lose your place? What are you thinking?

Steve: No, I just can't believe this.

Leo: Oh, okay.

Steve: I'm just thinking, oh, you're kidding me. Believe it or not, cryptocurrency mining, browser-based, which we've been talking about...

Leo: Coinhive and all that.

Steve: ...has figured out to keep running after you stop your browser.

Leo: That's really bad. Now they're using your CPU cycles all the time.

Steve: Oh, lord. Okay. So then, as we know, the current most popular browser on the Internet is Chrome. So it's the biggest target. We also have talked about how Chrome has designed their architecture so that they have a separate process per tab, or in some cases per window, which gives them some advantages. The downside is it tends to be significantly more resource intensive. It burns up memory because you're creating whole processes per tab instance. But the flipside is you get better isolation and, in some cases, superior performance.

So the recent rise in bitcoin price, and as we know it's dancing around $12,000 now - it briefly touched it a couple days ago, it's 11.7 at the moment - it's created increased pressure to steal mining cycles from, well, pretty much anybody they can. So of course people visit websites. And as we discussed, Coinhive has been until very recently - we're seeing some now almost predictable expansion of that. But Coinhive was the early JavaScript-based crypto mining source. But an analysis that we discussed a couple weeks ago of Coinhive use demonstrated that it didn't look like it was individual sites, with a few exceptions, that were deliberately hosting Coinhive instances and using visitors' processing power to mine bitcoins or Monero.

Leo: Actually, it's Monero, yeah, they weren't doing - yeah.

Steve: Yes, exactly. Instead it looked like all of the instances across the 'Net, thousands of sites were all accruing to the benefit of one or two entities. So it looked like these were JavaScript injection attacks of various sorts. For example, if a library could be commandeered and mining injected into the library, then all the sites everywhere that were causing their visitors' browsers to download that library would inadvertently be also inducing mining on those visitors' machines.

So anyway, so naturally, as bitcoin or Monero, whatever - I think this may be bitcoin that is being mined in this new instance because it's being hosted now on an AWS, on CloudFront, from Amazon. And what's happened is the guys at Malwarebytes found websites spawning where visitors went with Chrome because this requires Chrome because of its independent process model. They're spawning a tiny pop-under Chrome window which hides the mining code under the date of the taskbar on Windows. So it's literally just barely not offscreen. And if your Windows UI has transparency set on the taskbar, you can see this little mining window.

And so what happens is you go to one of these sites. Without you being any the wiser, another instance of Chrome is started under your Windows taskbar. And so you don't know it. And you do whatever you're doing. Maybe you think, gee, my system seems a little sluggish. Wonder what's going on? But whatever. You then close Chrome. Well, you didn't close the pop-under, which is running in its own process. And it's deliberately not using 100% of your system because it doesn't want to give itself away. It keeps itself to between 50 and 60% of your CPU.

So you're not getting, I mean, you are having your processing cycles stolen. This thing establishes a connection to the cloud, and somebody is gaining some benefit from tying up half and more than half of your system persistently in the background. There will be

the Google Chrome icon on the taskbar for users who know to look for that and understand that it means here's a running instance, not just a click here if I want to launch Chrome. And of course savvy users who know how to run Task Manager could bring Task Manager up, and you will see chrome.exe with 50+% of the system of your CPU constantly being sucked up. So the user visits a website, silently loads the crypto mining code. The CPU rises, but is not maxed out. User leaves the site, closes Chrome completely, yet your CPU continues running at a little more than half of its use as it continues crypto mining in the background.

So I just, as I'm looking at this, I'm thinking - I sort of stand back, and I just think, what a world that we're living in now. We have virtual currency, which is a thing, and we're on the Internet. We're using powerful computers. It's possible now to visit a site which, through no fault of our own, no insecurities, no vulnerabilities, everything patched up to the hilt, and a chunk of the system that we're using gets commandeered by either a malicious site that we make the mistake of going to, or a good site that's own security may not be up to snuff, or in fact it is an ad network which is spreading this.

So a site is being paid by an advertiser to give it a presence on its page, and that presence allows it to run script to cause it to launch another instance of the browser, which hides itself where it won't be seen and then statically burns half of our processing power in the background until we realize, wait, what's that thing running on my tray; or we reboot, because the system does seem a little slower than usual, in order to cleanse ourselves of this. And meanwhile, somebody somewhere that we have no relationship with is making some small, I mean, not even a lot because our systems are not optimized for crypto currency mining in this way. But it's more than nothing. And that's the world we're in as we enter 2018. Incredible.

Also, we keep seeing problems when we retrofit things that were not designed from the start into newer technology that we now want. And of course Unicode keeps giving us problems. We've had lots of coverage of Unicode being used in URLs to spoof websites, where it's possible for a domain name to contain a cleverly designed set of Unicode so that it looks like PayPal.com, but it's not PayPal.com. It's another domain with Unicode in its name which is intended to be deliberate because we want to allow foreign language characters, not just the original seven-bit ASCII which is just the normal printable characters on our keyboard, upper and lowercase and numeric and so forth. We want to allow accented characters and so forth. So in abusing that capability, miscreants - they're back Leo, the miscreants are back - they're able to register domains with Unicode that look like PayPal.

So, okay. Turns out that a researcher, Sabri Haddouche is a developer and pen tester - penetration tester - bug hunter and privacy advocate. In his day job he's part of the security team at Wire. In his free time he dabbles with his own security projects such as this one. He figured out how it was possible to similarly abuse Unicode in email headers such that all of our antispam/antispoofing measures would still function, yet a user could receive email that looked authentic - and we might as well keep picking on PayPal because probably - from PayPal. And no matter how closely you inspect it, and even if you then, like we've talked about add-ons that would dynamically show that the DMARC, which is the state of the art in anti-email spoofing, which uses DKIM and SPF, all of that cryptographic signature stuff, you could have a big green checkmark saying, yes, this is authentic from PayPal.com email, and it not be.

So it turns out way back in 1992 RFC 1342 was titled "Representation of Non-ASCII Text in Internet Message Headers." The subtitle: "What could possibly go wrong?" And in fact, yes, 33 different email products and offerings were vulnerable when Sabri responsibly informed everybody. And I've got a link here, Leo, in the show notes to a Google Docs

spreadsheet which he's maintaining, showing all of the products which are vulnerable.

And unfortunately, iOS, actually all of Apple products, both iOS and macOS, are vulnerable to this display of spoofed Unicode. Not everybody is. But it's rather sweeping in terms of the exposure. If you keep scrolling down, you'll find a link to Google Docs in his coverage there. So he responsibly announced; and, as of today, because this happened this morning, everybody got at least three months' prior notice. Eight products have fixed this, eight of the 33, so about a quarter of them.

Leo: It's pretty much everything I've ever used.

Steve: I know. I mean, it is widespread.

Leo: Oh, Claws is safe. I really love Claws Mail. Okay, good.

Steve: Whew. Especially for this time of year because we wouldn't want Santa to have a problem with his email.

Leo: No. But this would be - Unicode would even affect this if you didn't use HTML mail; right? Just text mail would work.

Steve: Yes. For example, iOS's own email client has a problem because you're able to embed a null into this Unicode string, and Apple stops showing text after the null.

Leo: Gmail on the web is safe. Good news.

Steve: Yay, good.

Leo: I would bet that's the most used email.

Steve: I think you're right, yes. And do you know if they were, or did they fix it?

Leo: I just see no, no, no, all green, across the board.

Steve: Oh, wait, wait. It's listed there because it was vulnerable.

Leo: Oh, okay. So they fixed it.

Steve: So yay to them. Yes, they did fix it. Good.

Leo: Okay. Doesn't have a date of fixing, so I don't know.

Steve: No. And...

Leo: That's good news. And FastMail, which I use on the web, is also safe.

Steve: And Proton Mail also. They jumped on it and got that fixed quickly.

Leo: But, I mean, on the web says it is still vulnerable.

Steve: Oh, oh, oh. I thought I...

Leo: Yeah, see, this is unclear. So he says "Is affected by Mailsploit? Yes. Spoofing, yes." But then it says "Fixed as of 1 September." It's unclear. I guess "no" means never vulnerable. "Yes" means once vulnerable. And then you've got to check the date. This is not very clear.

Steve: No. He's good at finding exploits, but not so good at explaining what it is that is going on afterwards.

Leo: Not so much. Yeah, okay.

Steve: But interestingly, two vendors, Mozilla and Opera, both said they won't fix the bug.

Leo: What?

Steve: They consider it to be a server-side problem.

Leo: Well, they might be right. Like Gmail doesn't have the problem; right? On any browser.

Steve: Right, right. And another one, which is Mailbird, closed the ticket on this guy without responding. So anyway, I'm glad to have the research. And it's interesting, I mean, I guess not that surprising that here again Unicode has bitten us in a way that wasn't - oh, I forgot to also say there are some sites which he also discovered a cross-site scripting hack that can be leveraged in the same way. So that can be even worse. And on that page that I have a link to, that he links to from his page and that I have a link to in the show notes, he does also have a column for whether they are vulnerable to various cross-site scripting attacks, which a number of them can be vulnerable to.

Okay. Google has announced from their blog posting just this month, I think it was on the first, from their safe browsing team, that they're going to start expanding their enforcement of something we covered about a month ago which they called the "unwanted software policy." They initially sort of announced it, but they didn't explain exactly what was going to happen, the idea being - and we'll remember that they were saying that they were going to start getting proactive about websites whose behavior they didn't like, and that developers could register ahead of time to be informed if Google was unhappy with them in order to resolve this before the hammer fell. We now are getting more information about what this means.

In the show notes here I have a big picture of what they intend to display, which would pretty much drain the blood out of anybody. It's this big red screen, big X in a stop sign, Deceptive Site Ahead, saying that attackers on whatever site you're about to visit may trick you into doing something dangerous like installing software or revealing your personal information, for example, passwords, phone numbers, or credit cards. And the default is to go back to safety, meaning don't proceed. So you're able to click something if you want more details. But this is pretty much going to stop anybody who doesn't really know - I'm not even sure if you can bypass it. It doesn't look like there's an, "Okay, fine, I want to proceed."

So their intention is that, as this big scary banner says: "Apps handling personal user data or device data will be required to prompt users and to provide their own privacy policy in the app. Additionally, if an app collects and transmits personal data unrelated to the functionality of the app, then, prior to collection and transmission, the app must prominently highlight how the user data will be used and have the user provide affirmative consent for such use."

So, I mean, these are reasonable things. Basically, as it has been, apps have been able to just, as we all know, when you install it, it says, well, we need all of the following permissions. And it's like, okay. So lots of apps get overly broad permissions from their users at install time and are never accountable afterwards. Google has decided, okay, enough of that. We're going to, if an app is doing something beyond its clearly intended scope and function, it's going to have to provide a pop-up and say we want to do these things for these reasons, and the user provide affirmative consent. So yay to Google for this. I mean, this is all for the better.

So what's going to happen is that, starting in 60 days, two months from now, this expanded enforcement of Google's unwanted software policy will likely be resulting in warnings shown on user devices via Google Play Protect, or on web pages that lead to these apps. Webmasters whose sites show warnings due to distribution of these apps should refer to the Search Console, that is, Google's Search Console, for guidance on how to deal with remediation and resolution of the warnings. Developers whose apps show warnings should refer to guidance in the Unwanted Software Help Center. And then developers can also request an app review by Google using the article that Google posted on app verification and appeals.

So basically Google's providing a bunch of tools to keep this from being a big deal, but to begin to enforce apps being more responsible. And then apps published in Google Play will have specific criteria to meet Google Play's enforcement of this unwanted software policy, which they announced, and we covered at the time, back in August of 2017. So this is great. This is Google being proactive, holding apps accountable and just tightening up security for everybody who are using Android and Google Play Store stuff. And I think this will, if Google is able to do this, probably serve to essentially fence off a lot of the sketchy things that have been going on without any control.

So Leo, you'll remember, because I remember you pulled this up and you poked around at it, when we talked about twofactorauth.org, T-W-O-F-A-C-T-O-R-A-U-T-H dot org, where they had this really very nice, by website type, breakdown of which sites and services offer two-factor authentication. So that was the first step. Now we have USB-Dongle Authentication, a similar site showing which sites and services offer the use of hardware dongles for...

Leo: Oh, I love this.

Steve: Yes, yes, yes, yes. Very cool.

Leo: I'm all in in the YubiKey. I just use it everywhere now. Facebook does, which is great.

Steve: Yup. And so it's www...

Leo: I can be secure when they're stealing my social graph. It's great.

Steve: Yes, exactly, www.dongleauth.info is the site, D-O-N-G-L-E-A-U-T-H dot I-N-F-O, dongleauth.info, which is a list of websites and whether or not they support one-time passwords or universal second factor, U2F. And they do the same thing. In order to help people quickly go somewhere and answer a question, they break it down by category: backup and sync, cloud computing, communication services, cryptocurrencies, developer sites, domains - I guess that means like domain registrars - education, email, entertainment, finance, gaming, health, hosting and virtual private servers, identity management, investing, payments, remote access, retail, security, and social. So a complete breakdown there. And then also they have a separate index by devices. So you can click on "dongles," and it will show you all of the current one-time password and U2F dongles, and which ones support which of those or both protocols. So nice piece of reference information for people who are interested in hardware-enforced second-factor support.

Leo: That's great.

Steve: Very neat, yeah.

Leo: Yeah, that's really great. Do you do that? Do you use a YubiKey?

Steve: I haven't. I was...

Leo: You're going to be doing SQRL soon enough.

Steve: Exactly. And so we'll see if that - hopefully that will get some traction. I know

that Yubico themselves have said that they will be supporting SQRL in their hardware as soon as they're able to. So we'll see what goes on. But I do, I'm a big user, as we know, of the time-based tokens. I think that's very strong security.

Leo: Authenticator, using Authenticator, yeah.

Steve: Yes, exactly.

Leo: Do you think that's as good as a hardware key?

Steve: The argument you could make is that, well, if somebody got into your phone, and they got into the authenticator data, then they could get all of your private keys. So the authenticator works by having private keys in your phone, which are technically vulnerable. The advantage of the hardware dongle is the hardware keys exist in it. And so the idea is it can't be hacked, we hope. So, yes. And so I guess the point would be, if you absolutely really desperately needed security, then it would make sense to do that. But my sense is you're potentially inconveniencing yourself a lot for maybe only a little bit greater security.

Leo: Yeah. I mean, your phone becomes your dongle, basically.

Steve: Right, right.

Leo: So that's fine. I don't have a problem with that, yeah. And it is inconvenient. I mean, I keep my dongle, my YubiKey on my keys. But I'm always reaching for my keys. And maybe that's a problem because, if you found my keys, you'd have my YubiKey, although I don't know if you'd know what to do with it. You'd have to figure out who I am, figure out what my logins are, get my passwords, and then you'd have the second factor.

Steve: Right. And now every listener to the podcast knows...

Leo: Yeah, shoot. Oh, dang it.

Steve: Oh. Yeah. Now fake them out, Leo. Put a dummy YubiKey on your keychain and keep it in your shoe or something.

Leo: I should do that. I keep two on there. I keep two of them on there. You get to figure out which one.

Steve: So we have - and this is going to be, unfortunately, an ongoing topic. We have the persistent danger presented by insecure Internet-facing routers. And this is back in the news because last week researchers from Netlab 360 spotted a new publicly available

Mirai variant. You know about the Mirai worm, which was causing huge havoc earlier this year. So an update in the Mirai malware has allowed malware to spread to another hundred thousand networking devices made by ZyXEL. Is that how you pronounce it, ZyXEL?

Leo: I say ZyXEL [zy-cel].

Steve: ZyXEL, that's probably it, ZyXEL.

Leo: Z-Y-X-E-L, yeah. They've been around forever.

Steve: Thanks for making a pronounceable name. Oh, yeah, they have, and I have liked their stuff. I think I have a couple of their dumb switches.

Leo: Remember they used to have modems, really the best, the best.

Steve: Early, yes.

Leo: In the USRobotics phase.

Steve: So on the end of October, actually on Halloween, October 31st, a new exploit was posted that allowed remote access, unauthenticated remote access to a class of these modems. Over the course of just 60 hours, starting on November 22nd, when this was deployed into Mirai, nearly 100,000 new devices, which almost without exception had IP addresses in Argentina, suggesting that they were all provided by an ISP, like a single ISP would say, here, take this fabulous router in order to use our services, were commandeered by Mirai. It's a well-known CVE. It's 2016-10401, which explains that affected devices all share the same fixed superuser password, allowing, not surprisingly, remote attackers to obtain root access when a non-root account password is known. And that's also defaulted.

So this appeared on October 31st. A couple weeks later Mirai gets it. And now 100,000 new devices are under the control of Mirai. Now, as it happens, the two domains that the attackers were using to control these newly infected devices were seized and quickly sink-holed, which had the short-term effect of stopping the infection from spreading further and preventing the attackers from using the hijacked devices to cause Internet outages. Remember that's what Mirai did was it was a massive DDoS bandwidth attack that was aimed at various targets.

But those 100,000 Internet-connected devices remain insecure. So they are still susceptible to takeover by any other Internet worm that may wish to take up residence in them. And unfortunately, this is where we are today. We have a large inventory, and probably a growing inventory, of insecure and persistently insecure, well-connected routers which are going to be hosts for worms and service attack platforms. I don't know how we get out of that situation.

So various classes of software have been injecting their own code into Chrome in order to

provide features like accessibility features, but also AV software has been doing this prevalently. The problem is that, in literally sinking their hooks into web browsers through a process known as code injection, they tend to destabilize the browser.

Google's telemetry has observed that over 15% of Chrome users running code-injecting third-party applications on their Windows machines are experiencing crashes of Chrome, which is causing them a great deal of trouble. They don't know that it's because they've got some flaky AV that hooked itself into Chrome in order to protect them from stuff that they're downloading. They just know that their system crashed when they were using Chrome, and they're not happy. And of course they blame Chrome, where in fact it's actually a third party that is running in-process stuff. So it induces significant instability, and Google has pretty much had enough of it.

So last Thursday in the Chromium blog they announced their plan to roll back and eventually block third-party software from injecting its own code into Chrome. And they're going to, of course, as Google always does, they're going to do this in a staged - announce it and try to keep anybody from being inconvenienced, but yet the hammer's going to fall. So in April of next year, 2018, with the release of Chrome 66, Google will begin informing users if code injection causes their browsers to crash. In other words, Google's going to first create some accountability here so it just doesn't, like, oh, Chrome crashed, darn it, and the responsible party is not identified. So they're going to alert them with the name of the responsible application and a guide to update or remove it, which they can do because they've been collecting telemetry on these apps. They know now who the culprits are.

So then, a few months later, in July of 2018, with Chrome 68, they're going to start blocking third-party software from injecting code into Chrome processes. But if this blocking prevents Chrome from starting, the browser will restart and allow the injection. So again, turns out that there are some situations where the third-party add-on software can have hooked things in a way that Chrome becomes dependent upon it so that, if Chrome then says we're not allowing you to do an injection here, Chrome will have a problem. So Google will watch that. If it can't start, then it will allow the injection but display a warning guiding users to remove that particular software. So Google's really going to push back.

And then finally, one year from now essentially, January of 2019, with no exception, starting with Chrome 72, Google will completely block code injection from any third-party software. So basically any AV, any accessibility solutions or anything else which is using code process injection in order to change, enhance, protect, whatever the reason, in Chrome, that's going to be impossible. Chrome's simply going to not allow that a year from now.

However, in their announcement, Google notes that there are safe means for achieving the same things using the native messaging API that Chrome offers, or just having the application create a standard Chrome extension to add functionality to the browser. The user would then need to install the extension or the installation of the feature, or the app, the AV, whatever, would have to take them through doing that. So it makes it apparent. But what this does is it then creates a sanctioned means for apps to add functionality to the browser, rather than the third party just autonomously reverse-engineering hooks which they then insert into Chrome and create instability in the process.

So again, I think this is the right way to do it. It's unfortunate that this wasn't blocked from the beginning because then applications would have always had to do it the right way. Chrome is saying, okay, we're going to force this now because this is causing too

many problems; 15% of Chrome users are having this happen. That's big. That's a large number. So Google is just saying, nope, we know who you are. We're going to warn people in April. We're going to get stronger in July. And we're going to shut it down completely a year from now in January of 2019, which I think seems like the right thing to do.

I haven't talked a lot about sharing a real fun success story about SpinRite for a while, but I ran across one in the mailbag when I was pulling the show together. Rana Omar, who is in Canberra, hope I pronounced that right, Canberra, Australia?

Leo: Yeah, Canberra.

Steve: Canberra, with the subject "SpinRite yet again saves the day." He said: "Hi, Steve. Hope you and Leo are well. Big fan of the Security Now! show. Love the work both of you do." He said: "I thought I'd finally share my story of how SpinRite saved my day." He said: "You've probably had enough of these stories by now." No. No, no, no, no, no. No, no, no. Love, love, love them. Keep them coming, please. They always make me feel good, and I love - and then they're all different in various ways. And this one is, too. We haven't had one like this.

He said: "After listening to you mentioning other SpinRite users' feedback all these times, I thought I knew the ins and outs of the whole process; but, boy, I was wrong. I've had a single 3TB NAS drive which I had been using for years with all sorts of data on it." So that's Network Attached Storage, NAS. "One day the drive suddenly died, and I thought to myself that me being the SpinRite expert," he says, "after listening to this podcast, this will be walk in the park for me to restore.

"In Windows OS, the drive would show as MBR/RAW" - which is not good. That means Windows doesn't recognize it. And he says: "And SpinRite wasn't completely happy with it, either." So that sounds like there was some spooky stuff on the drive. He said: "I can't remember the exact message. I decided to pick up and scan the drive anyway. SpinRite encountered trouble that caused it to stop before it was completed. So I came to terms," he said, "with the failure to complete the scan message, and thought the drive is never going to be brought back. I let it go for few months and bought a new NAS [Network Attached Storage] with multiple drive bays.

"I realized after a few weeks ago that you had mentioned a similar scenario. Suddenly it dawned on me. I sprinted towards my PC to plug in the drive and, to my surprise, found out the drive was working and allowed me to back up almost all the data off it." He said, parens: "(There were few folders which I couldn't back up, but they were not important anyway.) All this time, I didn't think to plug in the drive on my PC to check. I'm so glad that I didn't throw it away.

"The lesson here is that, if SpinRite shows you that it has failed to complete the scan, in reality it has done enough magic to bring it back to life. As always, thank you very much for the awesome product. Can't wait for the new version. Keep up the good work. Thank you." Well, and Rana, thank you very much for sharing the story. And that is the case. We know that the problem is that drives are so autonomous now. SpinRite works with them to fix things.

But first of all, there's a limit to how much software can do. As we've been saying, ultimately in this battle for the drive to die, it will succeed. So don't push it too far. We know that, had he been running this on SpinRite periodically, this probably wouldn't have

happened. But the message I wanted to share and what I'm so glad he shared was that, if you run SpinRite on it, even if it doesn't seem to have worked perfectly, it very often can have done enough to be very useful to you. So definitely give it a try afterwards and see if you got enough back.

So we've talked about Quad 9, this DNS provider. I think it was two weeks ago, Leo, that we first talked about it. Last week with Father Robert there were many people super excited, some people unhappy that it was too slow for them. They liked the idea of all the security benefits from using Quad 9, but they just, like, okay, it wasn't as fast as my own testing showed it to be. I wanted to tell people that there is now a points-of-presence map for the service. It's at pch.net/about/pops, P-O-P-S, which is short for Points of Presences. So www.pch.net/about/pops, which people can look at and also follow over time because these guys will be more than doubling their points of presences over 2018, which means for people who are not currently near a Quad 9 DNS server, there's a good hope that'll change in the future. Also...

Leo: Yeah, they're all over the place. That's great, yeah.

Steve: Yeah, yeah. They've got 70-something at release a couple weeks ago, and they're intending to go to 180, so many, many more.

There is a Network Security and Certification books bundle at Humble Bundle that I wanted to tell our listeners about. It's got a lot of time left, 13 days. And the books are, as the title, it's network-security-certification-books, all hyphenated. I've got the links in the show notes. They're MCSA Windows Server 2012 R2, various study guides for different exams, for the MCSA exams, and also for System Center and Hyper-V. A dollar gets you five of those. Eight dollars gets you the CompTIA Network+ Study Guide Exam, Third Edition; Cisco Networking Essentials, Second Edition; Microsoft Windows Networking Essentials; and Data Storage Networking: Real World Skills for the CompTIA Storage+ Certification and Beyond. That's all of the first five plus those four for eight bucks.

And then for a total of $15 you get another four: The CompTIA Security+ Study Guide, Sixth Edition; Network Attacks and Exploitation, a Framework; Network Security Bible, Second Edition; and Phishing Dark Waters: The Offensive and Defensive Sides of Malicious Emails. Ooh. So anyway, I knew that would be of interest. And those of course eBooks that you're able to purchase. These are all published by Wiley. This is a Wiley bundle, whereas the previous was - there was a sci-fi set, and then there was also another, oh, it was the Java. And those were all O'Reilly offerings.

And, finally, some great feedback from our listeners. @theNickNiti, Nick is his name, said: "iOS 11.2 is almost usable on my iPhone 7. Looks like most glitches were fixed. Maybe Apple actually QC'd this version." And of course 11.2 just came out, what, a day or two ago, I guess. No, I think it was Friday, wasn't it. And so I've updated all my stuff. And I caught your talking on MacBreak about the subtle little new things.

Leo: Like a second bug, yeah.

Steve: I hadn't noticed that little gray underline.

Leo: Oh, that bar, yeah, that was weird.

Steve: Yeah. And the weird spinning...

Leo: Yeah, the fidget spinner underneath the icons, yeah.

Steve: Yeah, crazy.

Leo: I don't know what that is all about.

Steve: Yeah, yeah. Also the black, @theblack - I don't - @theblack - I don't know what this - @theblacktandog, I guess is his name. He said: "Wow, Quad 9 and Firefox 57." Which of course everyone's liking the speed improvements of Firefox 57; and so he's saying put those two together, and you've really got something. Also, big news, thanks to a number of our listeners who sent me this. I noted here the first person who I encountered, and that was Dan Kutka, who said: "Quad 9 does have a secondary secure DNS address."

Remember that 9.9.9.9 - and that's the only one I knew about. 9.9.9.10 exists, but it's nonfiltered. So there is a secondary secure, that is to say filtered, DNS. It is 149.112.112.112. So again, 149.112.112.112, that's what you want to use as your secondary DNS behind 9.9.9.9. And I saw somebody somewhere say that there were some DNS resolvers which always queried all registered DNS. I had said last week or the week before, my own observation of Windows, and I have studied what Windows does with DNS because of course I wrote the Benchmark, it always issues the primary. And then, if that fails, it issues all. And then it remembers who responded first.

I did see some comment somewhere that there were some systems that always issued all and then used whoever came back first. So it suggests you do want to, if you can, if you want to use Quad 9, use the filtered IPs. And so for the secondary you want the 149.112.112.112. And so thanks to everyone for making sure that I knew about that.

Tracy Lipp tweeted: "I don't think iOS is to blame with instability issues in your phone." Okay, well, first of all, they're not instability, but okay. He says: "I have the same phone. In Finland it works perfect," he tweeted. "Just got to L.A., am using T-Mobile, and it's terrible. Phone is occasionally unresponsive and unstable. It didn't do this at all in Finland." So data point, I guess. That's interesting. For me it seems to be entirely performance. Well, okay. There were cosmetic bugs. And I heard you refer to them exactly that way in the last podcast also, Leo.

Leo: So that's what you were experiencing? Because I wasn't sure based on what you...

Steve: Yeah.

Leo: So it's not showstoppers, it's just ugly.

Steve: Yeah. Although...

Leo: Weird stuff, though, like [crosstalk].

Steve: Yes, like I've had iMessage balloons like stick on the screen when other stuff scrolls behind them. And like, okay, whoops.

Leo: Yeah, yeah, yeah. Updating issues, yeah.

Steve: Yeah. But definitely seems to be performance. It's like, for example, and I just went back and checked it after iOS 11.2, if I swipe sideways to scroll through the apps, there's always a pause. Whereas the phone used to just follow my finger right from the get-go, now if I do a swipe it thinks about it for a while, then it goes, oh, he's swiping, and then it does it. It's like, okay.

Leo: Have you tried turning off - going into Accessibility and turning off animations and...

Steve: I always turn it off.

Leo: Okay.

Steve: I don't like it at all.

Leo: There's a lot of cosmetic stuff that would absolutely slow things down.

Steve: Yeah. All that zooming and swooping and all that, yeah.

Leo: Yeah, yeah, I don't like it. So you turn that off, and you're still getting these hesitations.

Steve: Yeah. I just think it's old, although I don't, I mean, I'm loving my iPhone X.

Leo: Me, too. Isn't it nice?

Steve: It really is a nice experience. And I like that it's smaller, yet has the same physical presence. And I didn't think I was going to like the rabbit ears, but I don't care

at all. They don't - I don't even see them.

Leo: No. I don't see them anymore. Well, because a lot of times they're not visible at all.

Steve: Yeah, yeah.

Leo: Screens work around them.

Steve: And a last question from Yann Saint-Laurent. He said: "On the last episode of Security Now!, you talked about an HP printer that the drive was encrypted" - and I'm just going to quote from him directly - "it gets swapped to a non-encryptable, then back to a FIPS encrypted drive. If the drive is providing the encryption, how could it now be readable? Is the private key still in the printer?"

Okay. So to clarify, this was a very clever hack by the guys that took a deep poke at HP last week. And Leo, you and I talked about how disconcerting it was years ago when it was discovered that enterprise-class printers that had been taken out of service had had hard drives kind of secretly in them, had all of their print jobs sitting on the hard drives, which is representing a huge privacy concern for the companies that had decommissioned these printers and didn't know there was a hard drive that had financial statements and who knows, corporate future planning stuff, potentially, on those drives.

Well, it turns out that HP uses encrypting drives where the drive itself supports its own native encryption. And so when the printer boots, the board that the printer is docked onto, the printer's motherboard provides the key to the drive to unlock it. So these clever researchers realized, huh. If we pull the drive out in order to suck the firmware out, it's encrypted. So that gets in our way. So what they did was they just stuck into the printer a drive that did not support native encryption. Well, of course it was empty. But then the printer thought, oh, I must have had a failure of my hard drive. So they reinstalled the firmware through the USB thumb drive stuck into the printer, and said, oh, here, please repair your hard drive.

So the HP printer sucks in the firmware, loads it on the hard drive, tries to give it the key for encrypting itself, but it doesn't support encryption. So the drive ignores the key, stores all of the firmware unencrypted. Then they pull the drive out and mount it on their research machine and have access to all of the firmware. And as we mentioned last week, unfortunately it's using Windows CE, so they have file systems, and they're able to reverse engineer, and they found an incredible number of vulnerabilities in HP enterprise-class printers.

But so to answer Yann's question, the idea was they did not put back in a FIPS standard encrypted drive. They put back in a drive that did not support encryption, reinitialized the drive using the printer firmware from a thumb drive, and then pulled that out and had access to it. So very clever hack by those guys, and hats off to them. And that's our podcast.

Leo: Wow. I hardly can believe it. We are here at the end already? Wow. All right. And next week we'll talk a little more about ElcomSoft's contention...

Steve: I'm going to do a - that's going to be a, yes, a deep dive.

Leo: ...that Apple has overbalanced for convenience over security on iOS.

Steve: Yes. That will be our topic. I'll be fully tuned up for that next week.

Leo: Yeah, good. Because you have been singing the praises, of course, for iOS and its security policies.

Steve: Why I really want to understand, yes, what they've done.

Leo: All right, good. We'll look forward to that. That's next week. You can find this podcast and SpinRite, the world's finest hard drive recovery and maintenance utility, and many other freebies at Steve's site, GRC.com. That's Gibson Research Corporation, GRC.com. He also has transcripts, and Elaine should have an easy day today.

Steve: Yay.

Leo: It'll get up there, it takes, what, a few days; right? Four or five days to get the transcript up. He'll also have audio, MP3 audio of the show. We have audio plus video, if you want to see Steve's - he looks just like he does on the T-shirt and the mug, actually. The moustache really grew back nicely, I must say.

Steve: A little whiter. I think it's a little whiter than it was. But it's punishing me for ever removing it.

Leo: How dare you. You'll find that at our site, TWiT.tv/security - or I guess it's "sn" to save typing, TWiT.tv/sn. If you want to subscribe, I encourage that because you really do want to download these automatically, even if you don't get a chance to listen every week. Having each episode is a very nice thing to do. So find your favorite podcast app and type in Security Now!. You'll find it. It's one of the longest running shows on Broadway.

Steve: Number two. It's the second show on the TWiT podcast network.

Leo: That's absolutely right, yeah. Let's see. What else can I tell people? We do have the Steve mugs and T-shirts in stock at TWiT.tv/store. They make a lovely holiday gift for the security-minded geek in your life. I'm just saying. I'm just saying. And we will be back here as we are, typically, every Tuesday, 1:30 Pacific, 4:30 Eastern time, 21:30 UTC if you want to watch live at TWiT.tv/live. You could also join us in the studio, Sharif did, from Vallejo. Hi, Sharif. All you have to do is email tickets@twit.tv, and we'll put a - well, actually we don't have to put a chair out.

We've got a lovely Barcalounger for you.

Steve: Hi, Sharif.

Leo: He says hi. You know there's a handle on the side there. You can put your feet up, if you just want to get comfy. He's got like a 25-inch laptop. That's big. Is it 17? Yeah, looks like it. Big old laptop in his lap. So he's clearly involved, taking notes as we go. We love having the studio audience. But do email us so we know you're going to be coming, tickets@twit.tv. That'll save you the cavity search that Moe, our front door guard, normally would have to do. We could prepare ahead for you.

If you can't watch live, again, download it. If you want to watch live, be in the chatroom, irc.twit.tv, some very nice people in there who can talk you through any questions you have. They're listening intently. And we, Steve, at what point are we going to do the SQRL show?

Steve: We're close.

Leo: I hope so.

Steve: And we're going to do one.

Leo: Good.

Steve: Believe me, yes, we're loading up on nuts right now.

Leo: We have [groaning].

Steve: Ah, careful.

Leo: We have decided, I think, on our holiday episode because you know this show airs two days after Christmas, and we don't want to make anybody work on December 27th, so we're going to do a special.

Steve: I think I'm having my teeth cleaned that day, actually, so...

Leo: Well, good. It's about time, yes. I think I am, too. I save it for Christmas every year. It's easy to remember that way. Well, I won't - we'll just surprise you with the episode. It should be a lot of fun.

Steve: Okay, yes. We do know what we're going to do. People will - it's one of our "blast

from the past" episodes. Not one we have...

Leo: Not the Dog Killer, no.

Steve: No, we're not going that.

Leo: A very timely one.

Steve: It's not one we've ever repeated before. But it's the right one.

Leo: It's timely, as they say.

Steve: Yes.

Leo: All right, Steve. Have a great week.

Steve: Okay, my friend.

Leo: I will be here next week. I'm sorry to have surprised you last week. Thanks to Father Robert for filling in. But I will be back next Tuesday, December 12th, for another thrilling, gripping edition of Security Now!.

Steve: I can't wait. Oh, wait, I'll be here, too.

Leo: You can wait, and you must wait.

Steve: I will. I must. Okay. Bye.