Listener Feedback #208

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

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-499.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-499-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Lots to talk about, of course. We'll get through all the security news and then answer some questions. How secure is the new Type-C connector? Is it susceptible to BadUSB? Steve has a good idea, something that could make him a million dollars, but he's going to give it to you for free next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 499, recorded Tuesday, March 17, 2015: Your questions, Steve's answers, #208.

It's time for Security Now!, the show that protects you and your privacy and all that stuff online. And there's never been a better time for Security Now!. Steve Gibson is here. He is our Protector in Chief. Hello, Steve.

Steve Gibson: Hey, Leo. Great to be back with you on our regularly scheduled Tuesday.

Leo: You sound better. Are you all better? You feeling good?

Steve: Ah, we're just sort of in the mopping-up phases of last week's disaster. Yeah, as I mentioned then, I hadn't been sick in, well, I have never been sick on a podcast, and I don't think I've been sick in 10 years. So it was - I guess it was something you have to go through every so often as these viruses - it's funny, too, because I had various friends wanting to just get together and hang out. And I said, no, no, no, no, not until I am absolutely sure that this thing is dead and gone. It exists to try to reproduce and live. That's what a virus is, you know, it's a life form trying to perpetuate itself. And this one stops here.

Leo: Yeah, there was a science fiction novel I read, oh, I wish I could remember who it was. That's exactly what he - he's talking about his parents, who got replicated out of existence by another life form, a virus. Oh, I wish I could remember. Anyway, yeah, that's kind of what it is. If you think of it that way, it is, it's another life form.

Steve: You might be thinking of, what was his name, Phssthpok? I think it was a Larry Niven novel ["The Protector"].

Leo: Might have been a Larry Niven. I wouldn't be surprised, yeah.

Steve: Yeah. And we don't understand the lifecycle of the species until the very end. One of the things I loved about Niven's books is that they really surprise you. You feel like you're engaged, and you're watching the plot, and you know what's happening. But there's a whole big meteor about to drop on you.

Leo: I love that kind of - I love that.

Steve: Where you just don't - you're completely blindsided. You do not see it coming. Now, that kind of book doesn't read a second time as well because it really does depend upon you being fully informed, yet completely unaware of something crucial. But anyway, yeah. And I don't think that Jerry does that. He has that when he co-authors with Niven, but...

Leo: Oh, maybe it's a Niven thing; huh?

Steve: I think it's a Niven thing because in some of Larry Niven's own books, he does that. And I see it when they do it together, but not in Jerry's solo work.

Leo: I love plot twists. I just love them.

Steve: Yeah, yeah. So we have sort of an interesting week of stuff. Not huge, catastrophic, sky is falling news, but good stuff to talk about. There's a new nasty crypto thing now making itself known called TeslaCrypt. I want to talk about the Yahoo! announcement at SXSW on Sunday, their move to eliminate passwords. Then there was a big scurry because, if you didn't really understand the problem, you could easily say "Comodo does it again." But that's why I said, uh, kinda. And then the InstantCryptor guys, who I introduced a couple weeks ago as sort of a teachable moment, where they were doing a web-based encryption, but it really missed something important. They found out, tried again, but still haven't got it right yet.

Leo: Oh, no.

Steve: So we have more teachability. And then a great - we have a Q&A this week, so

10 questions, comments, and answers from our listeners.

Leo: Good show ahead. The news. That'll kick things off.

Steve: So can you bring up our picture of the week?

Leo: Yes, I can.

Steve: This is a capture of the screen which someone is presented when they've been unlucky enough to get TeslaCrypt installed.

Leo: It's like a CryptoLocker, kind of, but for a specific...

Steve: Well, okay. So it is, it's like, it's very much like CryptoLocker. In fact, it even borrows a little bit on the CryptoLocker brand by using a CryptoLocker.lnk link which it leaves on your desktop, even though...

Steve: Oh, that's a...

Leo: ...the guys, yeah, it's weird, sort of like they want to trade on the brand. The guys at Sophos have noted that the code is completely different. So in no way is this CryptoLocker. But what strikes me when I'm looking at this is how professional-looking this has become. The first CryptoLocker was that annoying neon red, amateurish sort of, it looked like it was just sort of thrown together at the last minute. This thing's got TeslaCrypt in barcode across the top, and that's actually a legitimate barcode.

Leo: Wow. Wow.

Steve: So they took the time to actually do - if you notice, the barcode above the T at the front and the back are the same. No other letters in there repeat, so you can't verify that further. But, and, I mean, look at it. What I love is just how cheeky this thing is. It says, oh, by the way, all your important files are encrypted. It's like, what? And then very clean-looking text, "Your Bitcoin address for payment" is like this. And at the moment the cost for private key for decrypting your files will be 1.5 BTC. And they handily convert it into U.S. dollars. It's about $415 U.S. at this point in March.

And so they can accept payment to give you the benefit of their decryption of the files that they just stole from you, essentially, either by Bitcoin or by something called PaySafeCard or Ukash. Although the cost for using either of those two, the non-bitcoin choice, is higher, it's 400, which is around $600 U.S. at this point. And as a final teaser, sort of as a proof that they really can do this, they will decrypt one file for you to prove they can. So you could, like, oh, my god. And then when you sort of settle down, and you think, okay, what one file must I really have?

And they don't decrypt it on your system because that would subject their private key to capture. Instead you upload the file to them, they use your private key which only they have to decrypt the file, and then allow you to download it again in order to verify that they're able to do that. So this is what we predicted on this podcast the first instant that CryptoLocker came out, the first one, because it was making too much money. Oh, and I should also say that it says here, "Payment verification may take up to 12 hours." And similarly there is a time limit on your ability to do this.

So basically they're moving forward with this model of encrypt your files because it makes, well, basically it makes money. They've taken something away from you in a way that you can pay to get it back. And unfortunately, many people do. What's different about this latest version, this TeslaCrypt, although brand new code, so not apparently related to the others, is that, as like the others, it will seek out photos, financial spreadsheets, office documents and so forth. But in addition, it seeks out files related to dozens of games, including saved games, configurations, maps, and replays.

Leo: It knows how valuable those are to some people.

Steve: Exactly. You have a major first-person investment in that stuff. So it targets Call of Duty, World of Warcraft, DayZ, Minecraft, Fallout, and Diablo, as well as configuration files for Steam, which is of course the online gaming platform. And also, maybe because of the time of year, looks for files related to tax returns and personal finance such as Intuit's Quicken software and iTunes stuff. And it can now extend its reach into devices and drives connected by any means. And it will encrypt the files out on those, too. USB drives, network file shares, cloud storage folders, and other connected storage devices.

So, you know, this is, I mean, this was foreseeable when we first saw the first one. And what was so troubling was that these guys had done the crypto right so that it was pay them or, hopefully, fall back on a backup and recover from the point of backup, essentially. But so this is, you know, the predictable evolution of this concept of encrypting your files. I think, unfortunately...

Leo: Not surprising, yeah.

Steve: It is, exactly, it's the way of the future.

Leo: Did you see Triangulation yesterday? Because we had a guy named Marc Goodman on it. It was great.

Steve: I missed it.

Leo: Yeah, it's about - the book is called "Future Crimes." But he's a former LAPD street cop who - it's a great story. In '95 his sergeant said, "Hey, Goodman, do you know how to start spell check in WordPerfect?" He said, "Yeah, it's Control F2." He said, "Good, you're hired. You're on our forensics team." And he was like, well, you know something about computers, anyway. But he became a forensics guy, worked for the Secret Service, the FBI, Interpol. And it's a really compelling book because

he's talked about how these guys - this isn't some teenage hacker in Bulgaria anymore.

Steve: Right, right.

Leo: He said they're like startups. They have health plans. They have HR. They have, you know, they recruit. You get a job. And that's why it's starting to look like this. They probably have actually designers working there because there's so much money in it. So these things are run like startups.

Steve: You can imagine them on some foreign soil saying, oh, let's get the money of those rich Americans. I mean, for them it's just sport. It's like, why not?

Leo: Well, it does raise this issue, and he says this, too, he says, you know, I feel like if you're young and you're smart and you're a math major or a computer science guy or gal, and you're thinking about what to do, I would hope you would have, you know, some altruism, maybe want to do public service, maybe want to help law enforcement fight these guys. But unfortunately there's so much money in being a bad guy. And I worry that maybe we've raised a generation that's not immoral, but just kind of amoral, and that they're going, well, you know, it's a good job. It's a good gig. I'll do it.

Steve: Yeah. Hey, I had it in my rsum, I floated it around there, nobody would hire me, except these guys would. So I'm doing it.

Leo: Well, and he says we need about, what was it, a million new cyber experts in law enforcement next year. And there's just nobody - they can't hire enough people anymore.

Steve: No.

Leo: You would like it because he recommended, he's a listener, and he recommended all the things we talk about. He has a protocol, in fact you can go to TrueCrimes.com and read it, you'll completely recognize this, called Update, which is what everybody should do. He says the best model for this is not to fight fire with fire, but in fact kind of treat it like a public health issue with kind of an epidemiological foundation, have people protect themselves, do the right thing. Update their software. Don't run as administrator. All that stuff we've been talking about for years. The only thing I disagree with is one thing. He says, "Turn off your router when you're not using it, or unplug the Ethernet cable because that reduces your attack surface by that many hours." I don't think that's...

Steve: Technically. But, yeah, it's not...

Leo: But everything else...

Steve: I wouldn't have mine on right now if that was true, yeah.

Leo: Well, I mean, like, when you go to bed, unplug. But you would never do that because you've got services running all the time. I don't know if that's going to help you much, anyway, but...

Steve: Yeah, I sort of agree. And it's funny, too, because immediately I jumped to - and this idea of the epidemiological model, that people, eh, you know, really they don't get involved because they feel like they don't really understand their computer.

Leo: And they feel like, hey, it's just me. And we have to emphasize that, if you don't get vaccines, it's everybody who suffers from it. It's a societal problem. If you don't protect yourself, your computer becomes part of the problem.

Steve: Right. And what I was going to say was that, from a health standpoint, I'm increasingly in touch with how actually the model is the same on health. People don't understand their own body, don't really know what they should do. They get conflicting advice and suggestions from everyone. So they just sort of tune out. It's like, eh, well, I'll just have some, you know, chips, and that doesn't seem to hurt me immediately.

Leo: We should get you on the radio show more, maybe, and do kind of this "for real people" stuff, you know, for non-geek.

Steve: I mean, as you noted, it's becoming increasingly important.

Leo: Yeah, no, let's talk because I would love to get you on maybe the radio show every few weeks and talk about it.

Steve: Good.

Leo: Yeah.

Steve: So Yahoo! announced at SXSW on Sunday that they had decided to make the option of no passwords.

Leo: This cracked me up.

Steve: Yeah.

Leo: Oh, I'm really curious what you think of this.

Steve: Well...

Leo: They are one-time-only passwords; right?

Steve: What I like about this is that listeners to the podcast already have all of the information that we need in order to, like, parse this, to understand it. We were - it's funny because I'm seeing people in the press talk about something you know and something you have. And I'm thinking, you know, did they get that from us? Because, I mean, we've been talking about this stuff for years, you know, not quite a decade, I think. But those were the terms that we introduced for this notion of multifactor authentication.

And what Yahoo! has done is, I mean, they do have multifactor authentication, and they have traditional single-factor username and password, where it's something you know. What they've now done is created a different single-factor authentication which is not something you know but something you have. In other words, what they're promoting is that, if you elect to flip your Yahoo! account in this direction, then when you want to log in, you do not give it a username and password. I guess you have to give it a username. It has to know where to send the one-time password. So you just must give it a username. And to a device which you have preregistered with your account, it then sends a - and they call it an "on-demand password" is their new term for that. It's like, okay. So they own a little chunk of rather useless intellectual property.

But the idea is that it then sends your phone, via SMS, exactly the kind of thing we've always been talking about, some sort of never-to-be-repeated string. You enter that into this login screen, and it says, oh, okay, fine, you're authenticated. And so Yahoo! is of course billing this as that wasn't a password that you had to remember. You now no longer have to remember anything. Isn't that wonderful? And the reaction has been, well, largely from the security community, everybody's in favor of innovating on authentication. Of course I've been spending the last 18 months innovating on authentication.

And it's funny because I see, I've got so many people tweeting, I was cc'd with mentions in people tweeting to Yahoo! or anybody else in the press who dared mention this, about SQRL. It's like, yeah, yeah, yeah, we'll - don't. It'll happen if it's going to happen, and it's not yet ready to be shown. But so in general people are happy with the notion of innovation. But a great chunk of the security community is made uncomfortable by the fact that now you don't need something you know. It's only something you have. Or rather, it's only something you are sent. And that's a crucial difference. It's not actually something you have. They're billing it, not as something you know, but something you have. It's not. It's something you are sent.

So that distinction is the Achilles heel of this, as many people have noted, because it's an SMS message. And one of the number one things that malware does when it can in a mobile platform is capture SMS messages that are received by the phone and send them back out. So it just - it makes people uncomfortable that Yahoo! has, like, almost out of frustration with the problems they're having, done something different, but with a different set of problems. And mark my words, we will see this thing defeated. I mean, it's not like they've come up with something super secure.

Leo: No, it's less secure; right?

Steve: Yeah, exactly. The reason SQRL is special is that it is an online transaction. The server challenges your device to prove a secret that only it has, and it responds to the challenge. This has nothing like that. So that, if somebody intercepted this token going to your phone and entered it themselves, they're logged in as you.

Leo: And that's easy to do, I think. I mean, SMS is not very secure.

Steve: Exactly. It's a very weak, low-security messaging system, really never designed for authentication. It was designed for, you know, "Hey, I'll be five minutes late" sort of messages.

Leo: Well, it's fine for second factor. But this is one - this isn't second factor. And I think that's kind of...

Steve: Yes.

Leo: To me it feels like Yahoo's trying to trick people into thinking, oh, this is like second factor, it's more safe, when it fact it's much less safe.

Steve: Right, because you are not needing to provide anything you know. And in fact this would be something which, in a situation where you might be coerced to login, you can't say "I'm not going to give you my password" to the authorities. They put you in front of a screen, type your name into Yahoo!, and your phone sends them the key they need to log in as you.

Leo: They don't need you at all.

Steve: That's just one example of a problem that this system has.

Leo: They don't need you at all. They need your phone. They can type in your name. They don't need you.

Steve: Correct. Correct. Correct. And so the challenge in designing something that is easier to use is we do want a device that can stand in for the user. And that's why, and I don't mean to keep coming back to SQRL, but until this thing is out of my brain, that's where I'm living. But so that's why SQRL prompts you every single time, only once for your full password, but then it reencrypts the first "n" characters, and you can decide how large "n" is. And it keeps that in RAM. It never writes it anywhere. So every single time, every single time that you want to log in, you just get a little dialogue to confirm the name of the server where you're logging in, and you type click click click click, just like the first four characters of your password, and you're done. It's not like people are

logging in two times a minute. I mean, when you think about it, you're not really authenticating that often.

But, and the key is, you miss that once, and it wipes it out of RAM. So now you have to provide your full password again. So it's, you know, I've worked to come up with a secure compromise where it wouldn't get in your way, yet it really, it truly would be a way for you to authenticate the device to then stand in for you. Which is the power we're giving devices in this sort of a loop, like Yahoo! has implemented. But even in this day and age it's just like, eh, we'll just send you an SMS. Good luck.

Leo: We should point out you don't have to use this; right? It's opt-in.

Steve: Correct, correct. And they do have two-factor authentication. So the problem is, as you said, Leo, people aren't going to understand the danger. They're going to go, oh, this is new. This has just been announced. This must, you know, Yahoo! must have figured out something to make this safe. It's like, yeah, right, you know? The authorities get your phone, and they're now able to log you into Yahoo!, if you choose this option.

Leo: I don't think they even need your phone because can't they do a pen register and just get your text messages?

Steve: Exactly. You are able to pull past messages received.

Leo: This really is a pretty straightforward process.

Steve: Although I'm sure that it has a short lifetime and a single-use function so that you're only able - I hope it has those features so that it would expire and immediately expires when you use it.

Leo: Somebody makes an excellent point, though. If the system you use has some form of SMS password recovery, that would be no more secure than this, if that worked to recover the password with just a text message.

Steve: Yeah, that is a good point. If you say, oh, shoot, I don't remember my password...

Leo: Just send me a text.

Steve: Yeah.

Leo: We've got to get rid of passwords. We really do.

Steve: We do. We do. We do.

Leo: It's just a bad system.

Steve: So this article, this whole issue of the Live.fi certificate that got loose. Microsoft's all in a froth now, and they're sending out special notices, and everyone's running around. I just - I have some technology, I'm pointing to it on the screen here, which monitors Chrome's CRLsets, which is, of course, the very controversial, very weak revocation mechanism that Chrome now has. We went over this in depth, and I got into it with Google, of course, when I realized how worthless this was. But the one thing they do is at the very top of that file are hard-coded revoked certificates. And I'm somewhat dubiously proud to be number five in the list...

Leo: Oh, wow.

Steve: ...for my, well, for my revoked, remember revoked.grc.com. So they added me manually when people said, oh, look, Chrome has a bug. And so they took me out. And it's like, okay, fine, and then I changed my certificate, and now I'm back again, and they never changed their header. So that revoked certificate that I'm no longer using is still revoked, and maybe it's always going to be there, I don't know. But I'm no longer revoked because I just changed my cert. But anyway, for the longest time it had eight entries. This morning, about an hour ago, ninth popped up. And I got a notification from the monitor, hey, the blocked, they call it "blocked SPKIs" changed, and there's a number nine. And this must be the certificate that everyone's running around like crazy about.

So here's the story, though. This is really not Comodo's fault. Any certificate service could have made this mistake. And it's really not the certificates - so my point is it's not the CA's fault in any way. And I'll also note, just as an aside, I was a little surprised when I was doing some background research. Comodo is running around flapping their wings because, as of February 17th, I think the blog post of the founder said, they are now the number one certificate provider on the Internet.

Leo: What. Wow.

Steve: For the first time. Historically, for 20 years, it had been VeriSign.

Leo: Right.

Steve: And of course VeriSign, they were the certificate provider. Symantec purchased them some years ago, and but you still get to them at VeriSign.com. They had always been number one. They had been tied, and now Comodo has passed them by as the number one certificate provider on the Internet. Which, you know, I don't think says anything particularly useful about them. They're the last provider I would use for a certificate because, I mean, for example, they're the Superfish people also, and all of the Komodia...

Leo: No, no, no, no. They don't do Komodia. They do something similar. They used Komodia in one of their things. Not Superfish, though.

Steve: Correct. Oh, okay, right.

Leo: Comodo does, what is it, I forgot what it's called, but they have an equally horrible program, yeah.

Steve: That's, you're right, good. Okay. So here's the problem. When you go to any certificate provider, certificate authority, and say I want a cert for my site, they say, okay, you need to prove you're in control of that domain. So what does that mean? How could you do that? For example, they can give you some text which you put on a certain page. Like I've seen this, they'll - or they'll give you a crazy-looking page name, some grid-looking thing, just gibberish.html, and say, put this on your root, and let us know when you have.

And so then their automation will go check for that page on that domain. And if it's there, it proves you're somebody who has control of the web server of that domain. And so they go, okay, you proved it, so we'll give you a certificate in order to add TLS, HTTPS encryption to that site. But if you'd rather, you could just use email. They say, just make sure that you're able to receive email from admin, administrator, postmaster, hostmaster, or webmaster at domain, whatever your domain is, dot whatever. And so these are the traditional webmaster or domain master sort of admin accounts: admin, administrator, postmaster, hostmaster, and webmaster. Those are the options that Comodo offers, although all of the CAs offer variations on that.

So just because this guy in Finland was curious, he thought - he had an account with Microsoft's Live service. In this case it was Live.fi. And the email service allowed you to set up aliases. And he said, huh. What if I try to set up an alias for hostmaster? And it let him. So he then became able to receive email sent to hostmaster. That's all this was.

Leo: Wow.

Steve: And then he goes to Comodo because, you know, they're now number one.

Leo: I'm the hostmaster.

Steve: And he says, "I'd like to have a certificate, an SSL domain certificate for Live.fi." And they go, okay, yeah, we do that. Prove that you're in control. And they say, what email would you like us to send your authentication challenge to? And he said, "Uh, let's use hostmaster@live.fi."

Leo: Whoa.

Steve: Email came in, he clicked the link in the email, they said, "Yay, you qualify. Give

us your 25 bucks, and we'll send you a certificate." That's what this was. And any other CA would have done it.

Leo: Oh, my god.

Steve: So the mistake...

Leo: Is Microsoft at fault for letting you create a hostmaster address?

Steve: Well, yeah, I think so. I mean...

Leo: Probably shouldn't do that.

Steve: So the real problem is that - and I love - Dan Goodin wrote, has written a number of pieces about this in Ars Technica. And one line that I liked in his most recent note, he said, "The ease in obtaining such certificates and the difficulty in killing them off once they're issued are potent reminders of the continued insecurity of one of the Internet's most important security mechanisms. And so the real fault is that...

Leo: You're right, revocation.

Steve: ...it is that easy, is that authentication of domain control is as weak as it is, as weak as receiving an email to one of a number of preset accounts. Now, what this means is, of course, to protect themselves, everybody who has an email system where users can create aliases or their own accounts absolutely must block all possible common names that any CA might use. That's the problem. There's no standard. Comodo, I have never seen admin before, or really even administrator. They just made those up. But postmaster, certainly. Hostmaster, okay. Webmaster, yes. But administrator or admin?

So the problem is this particular CA added a couple. So to be secure against how trivially easy it is to authenticate via email, you'd have to make sure that you the domain controller went around and looked at every policy of every CA, and, I mean, that includes the Hong Kong Post Office, and makes sure that you have blocked anyone acquiring any of those email addresses because, if you don't, someone can, and they will prove to the system that they own it, and off they go. Unbelievable.

Now, Windows 8 and on has added an automatic updater for revoked certificates. It is available, but not installed by default, for Vista and 7 and also the server-side versions of that, Server 2008 and Server 2008 R2. So I have a note here, I have a link in the show notes to the automatic updater. If you're using 8 or later, you're already covered. If you're not, that is, if you're on 7 or Vista, it's probably worth adding this. It's a little lightweight gizmo, a little service that runs in the background, which will then, from now on, automatically allow Microsoft to revoke any certs of this sort that happen.

You know, this doesn't happen often. There's a grand total of nine, and I'm one of them, on Chrome's bad guy list. So it's not like there are a lot of these. But when they happen, they tend to panic a lot of people. And of course everyone ran around, even though this

one guy had the cert, he could have said, hey, look, I didn't give it to anybody. Here, you know, I'll shred it. And oh, my god, no. And so everybody, you know, all the browsers now have to scramble around and make sure that they don't, you know, that this didn't actually somehow get out and go somewhere else, when it was this easy to do all along, which is, again, the wakeup call for this whole system.

Note that none of this works for EV. That is, it only works for domain. On the other hand, the problem is that having a redundant domain certificate is fine. That is, Live.fi might have been protected, probably is, by a legitimate Microsoft Extended Validation cert. But that doesn't mean that the domain cert isn't just as good because that level of verification is all you're asserting. And as we've said on this podcast before, it ain't much. And this story just goes to prove it.

Leo: Wow.

Steve: Okay, now, InstantCryptor. These are the guys, it's www.instantcryptor.com. And you'll find a new little blurb on their page acknowledging indirectly our coverage of this a couple weeks ago. Remember that screenshot, I had a picture of Notepad, Leo, a couple weeks ago where it was just all full of crazy Chinese characters because I had decrypted with a different password, and InstantCryptor wasn't verifying the password in any way, which meant that, if you mis-entered it, you got gibberish back.

And I said, eh, you know, what that means is all they're doing is hashing the password to create a key and using the key with AES and Cipher Block Chaining to encrypt the file. Which is fine and all, but it's really not the way it ought to be. For one thing, you're not able to detect any modification to the file, which you'd really like to have. What is necessary, I explained, was known as "authenticated encryption," where you're both authenticating and encrypting. And there are a number of ways to do that. So yesterday I saw a tweet from CloudRail, who are the guys behind this, and they said, "@SGgrc Thanks for your feedback about InstantCryptor. We've updated it according to your advice to make it even more secure. Any feedback?" And unfortunately, here comes the feedback.

Leo: Oh, boy. You asked.

Steve: Yeah. And I also found in my mailbag when I was pulling the Q&A questions together, an email that the subject was "Feedback on InstantCryptor.com," from Mannheim, Germany. And the email read: "Hello, Steve. One of your listeners, who calls himself 'Advait from India,' has made us aware that you've discussed InstantCryptor.com in your podcast. I'm the main developer behind this product and was eager to learn of your opinion. In fact, we've modified the encryption procedure such that now the plaintext plus a hash value of that very plaintext is input for AES-CBC encryption, which allows us to check upon decryption if the password used was correct and if the message has been modified." Except it doesn't. "I'd be happy," Florian goes on, "to get feedback from your side and could provide you with more details. Feel free to contact me under the mentioned email address."

Okay. So what they did was they - it was another good effort. But this is another, you know, the underlying lesson here is stop trying to invent cryptography. That is, we're done with that phase, as I have said often. There's nothing more. There may be more bits, and every decade or two we get some major breakthrough the way we got elliptic

curve after RSA. You know, but that was a long time ago, and now it's only now beginning to roll out. And the ways we put these pieces, the way the blocks are assembled, that's all done now, too. But these guys sort of did another ad hoc solution, which it turns out is wrong. So what they do is you upload, or, I'm sorry, not upload, because this all happens in your browser. You point your browser at the file you wish to encrypt. What they were doing was simply encrypting it and then sending it up to the cloud.

Now, they're doing an SHA-256, that is, they're doing a digital signature of the file, sticking that on the end of the file, then encrypting the blob. And that's what they send up. And their logic was, and their logic was correct, but unfortunately not thorough, is that when they brought the blob back down into your browser, and you gave it a password, it would use that password to decrypt the blob, which should return the original plaintext and the SHA-256, which is 256 bits or 32 bytes tacked onto the end. Then if they run SHA-256 again on all but the last 32 bytes, they should wind up with the same result as those last 32 bytes. That is to say, that has of the preceding file, the digest, should match. And it's certainly the case that, if you entered the password wrong, that is, if you used a different password, you would get gibberish, and there's absolutely no chance that a hash of that gibberish would hash to the gibberish, the different gibberish at the end.

So this does detect, I mean, be careful with this because there's so much it doesn't detect. It does detect a different key used to decrypt. It turns out, though, that solid as we would think this is, there are well-known ways to fool it. If it were possible to get the user to upload something malicious, for example, it's possible to insert that malicious different stuff into this one and then have it go undetected. That is, there are well-understood means in the crypto world for circumventing this solution. And I knew that was the case, but I thought, okay, I need somehow, I need something I can explain clearly.

So I shot a note off to a friend of the podcast, Taylor Hornby, who's a.k.a. FireXware, who has his site Defuse.ca, D-E-F-U-S-E dot ca. And I said, "Hey, Taylor, here's the situation." I brought him quickly up to speed. He was a few episodes behind, so he hadn't run across this yet. And I told him, I said, I'm looking for the succinctest way of explaining how this is wrong. And so he said the quick answer is that it's well-known how to forge messages under this scheme, that is, where you hash, using an unkeyed hash, and then you encrypt. And that's part of the problem here is that SHA-256 is not keyed. So bad guys - and you can intuitively get it that anyone who wanted the hash to balance can make the hash balance by changing the cryptographic content and then rebalancing the hash. And the math behind this makes your eyes cross. But this is nothing for the cipher guru people. They're able to pull this off.

So he said: "The attack is pretty neat. It's explained in the second part of this answer." Then he provided me a link to StackExchange.com where some of this sort of math is shown [crypto.stackexchange.com/a/16431]. And he said the more complete answer is that today the burden of being called "secure" is to have been proven secure in what's known formally, because the other thing that's neat about cryptography now is this is all - all this stuff has moved into formal academic land. So it's called an "indistinguishability/non-malleability model." And there's even some, like, symbology for this. It's IND-CCA2.

And then he provides a Wikipedia link to ciphertext indistinguishability. And he says the dominant attitude today - and here's the real takeaway. "The dominant attitude today is that, instead of finding and fixing individual problems, we just outright refuse to accept anything that isn't proven IND-CCA2 secure." And that's this INDistinguishability under

adaptive Chosen Ciphertext Attack. That's what this approach is still victim to. He says: "That's the important lesson. You can't iterate on crypto design by fixing problems that come up. You have to start with something proven secure. And we have things that are proven secure." And he also, he ended by saying the Telegram messenger is another example of ignoring this lesson because, again, they just sort of made something up.

And so the way to do this right, and I will forward this text to Florian since he was kind enough to write to me and tweet me and provide this content for the show, is the only way to do this, and you've also heard me say this, is encrypt, then hash. That is, encrypt, then authenticate. This is the problem that SSL made back in the early days. So for these guys to do this right, they would encrypt the message under a key derived from the user's password. Then they HMAC, and that's a keyed - HMAC is a Hashed Message Authentication Code where you give it a key.

And the point is there's no way for the bad guys to know what that key is. The key you derive also from the password, from the user's password. Maybe just run, they're going to run the password through SHA-256. Just run it through SHA-256 again in order to create a key which you use for the HMAC. So then you key this HMAC, and that produces an SHA-256-esque hash, which is the authentication code that you tack on the end. So the result is no bigger than it is right now in what these guys are doing. Yet because you've encrypted first, and you've authenticated that encrypted result with a keyed HMAC, using a different key than the encryption key, that is utterly tamperproof.

And then, when you receive the blob, the first thing you do, because the last thing you did when you were generating it was to authenticate, the first thing you do when you're going to decrypt it is authenticate that there was no change to what you've received, and then you use the key from the password to decrypt the file. So, and if everyone would just do that, this is what I meant when I said this problem has been solved. If you do that, if you encrypt, then authenticate with a keyed hash, then you get what Taylor explains is this crazy indistinguishability under adaptive chosen ciphertext attack-proof solution.

And I will mention - this is me speaking, not Taylor - that he consults. He loves auditing people's crypto. That's, I mean, he really enjoys it. So Florian, if you're hearing this, I will mention this in my note to you that follows, but it's Defuse.ca, D-E-F-U-S-E dot ca. And Taylor would, I'm sure, love to help. He did dig into your code a little bit and confirm the things that I had said, so he's already up to speed.

Leo: He's a good man.

Steve: Now, because we've talked so much about the Large Hadron Collider and that fabulous...

Leo: "Particle Fever." "Particle Fever."

Steve: ...fabulous movie "Particle Fever," I just thought I would note...

Leo: Which is free on Netflix now, by the way.

Steve: Is it.

Leo: Yeah.

Steve: Is it really. Oh, fantastic. And everybody, if you are a Netflix subscriber and you have not yet seen "Particle Fever," please, please, please. You will, you know, this is the one I talked about a year ago where I came out of the theater, and the elderly friend of mine who I went with and I simultaneously said, "That's the best five bucks I ever spent." I mean, because it's a little small art theater, and they weren't asking for a ton of money, and it was a matinee, and it was five bucks. And, wow, I mean, it was really interesting. Anyway, with that lead-in, they are in the middle, after having been down for two years for refitting and beefing up, essentially, they are in the careful process now of bringing the LHC back online at the highest power level it has ever operated. It was running at 8 TeV. That's Trillion Electron Volts.

Leo: Oh, that sounds like a lot.

Steve: I know. Whew. And they're adding five to that. So they will be at 13 TeV. Now, 8 TeV, essentially the level at which this operates, that is, the power at which this runs determines their ability, essentially how much they can see. And they were able to prove the existence of the famous Higgs boson two years ago. And in 2013 Peter Higgs, who is still around, shared the Nobel Prize for its discovery, although the Nobel waited until it actually had been discovered. But he hypothesized, he said there has to be this thing here, even though no one had ever seen one. He says, "It's got to be there."

Well, it took them smashing stuff together, photons, at 8 TeV in order to finally see the Higgs boson. Now they're going to 13 because what they want to find is something that surprises them. So far the standard model is holding up. But there's this, you know, there still remains questions. What's up with this dark matter that most of the universe seems to be made of, but we can't find it anywhere. And, you know, which is to say the model says there must be dark matter, in the way that the model said there must be the Higgs boson. Well, they found that. But they haven't found the dark matter yet. But maybe that means there's a glitch in the model. Who knows? The model is the most successful thing that physics has ever created. It's done so much, the so-called "standard model."

Anyway, what they're hoping is that, essentially, it's like they're turning up the resolution on their scope. They're going from 8 TeV to 13, and they don't know what they're going to find, but they hope some stuff they cannot explain because it will then be up to them to explain it. And in doing so, they're going to learn something. So far they've managed to prove something and demonstrate that this crazy experiment is bearing fruit, which I just think is so very cool.

Leo: Well, not fruit, but bosons, anyway.

Steve: Bosons. Ah, wait a minute, let's see. Well, I'm thinking. I was wondering if there are any fruit names in there because we've got charm, and we've got, you know, all the quarks have, like, these weird designations...

Leo: Yeah, I know, it's so magical.

Steve: ...spin and charm and so forth. But I don't think we have bananas and pears and...

Leo: I don't think so.

Steve: I don't think so.

Leo: So it is, like, accepted that they confirmed that the boson exists. That's not a question.

Steve: Yes.

Leo: All right.

Steve: Yes, yeah, they did. And Peter got his Nobel and...

Leo: Yeah, I guess you get a Nobel Prize, that's confirmation enough.

Steve: Yeah. Yeah. Okay, so...

Leo: Do you - before you get...

Steve: ...I was going to say something about Barack and the Peace Prize, but I thought, no, no, no.

Leo: Yeah, no, no, no, please. Before you get to SpinRite, I know you want to do that, but a couple of questions I had for you, and maybe something we can talk about down the road, or maybe it's in the questions, actually, I haven't looked in the questions, Google's new OAuth solution, some people in the chatroom are asking about that.

Steve: Didn't know there was one.

Leo: All right. So I don't either, so we'll have to look at that. And then we've talked about BadUSB before. And a number of people raised this issue.

Steve: Oh, it is, it is in the questions.

Leo: Okay. We'll get to it, then, good.

Steve: Yes, yes. I was raving about the MacBook last week and how, you know, about this one connector. And it really - what we're going to need, I'll just - I'll give people a tease. We're going to need a condom.

Leo: [Laughing] Okay. I have a Snuglet, will that do?

Steve: I don't know.

Leo: All right.

Steve: We're going to need a USB charging condom, which we'll discuss here in a minute.

Leo: Oh, that's good idea.

Steve: In the meantime, in the meantime SpinRite. This is a little quickie in keeping with the spirit of the Q&A. Brandon asked, from Wisconsin, he said: "SpinRite GPT/EFI/UEFI?" He said, "I'm a longtime listener, and user of your products. It all started for me with ShieldsUP!. And I've been wondering about your future ideas for SpinRite. I know that you're re-writing it for much faster speeds and all sorts of wonderful features we've heard about, but I don't believe anyone's brought up the compatibility side of things. Are you planning on updating SpinRite to work with GPT formatted drives" - that's the GUID partition table as opposed to the so-called MBR, the Master Boot Record format - "drives that SpinRite 6 doesn't agree with because they are 'MBR Followed by EFI'? And are you planning on adding UEFI boot support?"

So I just thought I'd be explicit, since I guess I hadn't mentioned it before. Yes, SpinRite's full support for the Mac, among other things, requires that. And even later model PCs with Windows, what, I think at least 7, certainly 8 and beyond, everything's moved to UEFI booting rather than the old-style BIOS, and to drives using this GPT because they've got much larger fields. The problem with the old Master Boot Record format is that it had a partition size limit of 32 bits' worth of sectors. And 32 bits' worth of sectors, that's, of course, it was at one time seemed like we're never going to get there, 4.3 billion. But 4.3 billion times 512 bytes per is about 2.2TB. And we now know that there are partitions bigger than that.

So drives and OSes and everybody have been forced into a format that allows greater than 32-bit descriptors, which is what the GPT does. And so SpinRite will be 100% aware of all of that stuff and be able to run on drives of probably, it's safe to say now, any size. Although I have said that before and been wrong. So it's a little bit like Gates saying no one will ever need more than 640K of RAM because, you know, that was 10 times what the Apple II had, and surely that's enough.

Leo: It's got to be plenty.

Steve: Yeah.

Leo: Got to be enough.

Steve: Two things. While you were reading that, I just checked my Twitter feed, and Simon Zerafa sent a note saying that the axion, A-X-I-O-N, apparently is a candidate for the dark matter particle. There's a Wikipedia entry says: "The axion is a hypothetical elementary particle postulated by the Peccei-Quinn theory in '77 to resolve the strong CP problem" - we all know about that - "in quantum chromodynamics."

Leo: Oooh.

Steve: Oh. I hate those quantum chromodynamics problems.

Leo: Hate that when it happens.

Steve: Especially when there's strong CP.

Leo: Oh, yeah.

Steve: "If axions exist" - at least they gave it a good name - "and have low mass within a specific range, they are of interest as a possible component of cold dark matter." And this thing goes on and on and on. So I have, I mean, this is, whoa. But, you know, good. Whatever that is.

Leo: Sounds important. That's all I can say.

Steve: Yeah.

Leo: Sounds like 18 TeV might not be enough. We need more power.

Steve: If I add seven to today's date the 17th, I get March 24th. Which strikes me as the last day that voting is open for the Podcast Awards.

Leo: Well, get in there. It's not too late.

Steve: So that means that this is the second to the last podcast in which I'll ever be able

to annoy and remind and bother people that I would love to win for best technical podcast.

Leo: Come on, everybody, let's get him in there.

Steve: And it is necessary, by the way, to check your verification email, if you haven't, because they are sending out verification emails in order to make sure that you're not, like, making up a new email address every time. So you have to click a link in the verification email in order to prove that you're legitimate. I found one in my own.

Leo: Yeah. Have that sent to hostmaster@live.fi, and we'll make sure to count your vote.

Steve: Yup, and somebody at Microsoft will happily click on that for you.

Leo: But you can - so does that eliminate the vote often thing? You have to - or do you...

Steve: No, they still want you to come back and vote more. And they say you can do it every day. And so some people have been really neat. They've been tweeting reminders out to the people who are following @SGgrc. So really appreciate that.

Leo: If you don't win this, we've failed. We have failed in every way possible.

Steve: One way or the other, I'll never mention it again, either in humiliation or in glory.

Leo: Okay, good. Ready? Question 1 has arrived. Thump. Message for you, sir. Winfield, a listener in the U.S., wonders about freezing his credit, the cold dark heart of his axion: I just listened to your piece on freezing credit reports. I'm curious what your opinion is on services such as LifeLock, LifeLock.com. Also - and I have an opinion which I will share with you about that.

Steve: Good.

Leo: Also, wouldn't freezing one credit reporting bureau be enough, instead of all three? If you freeze one and have an issue with others, doesn't the one that's frozen prove to the others they're incorrect? Now, if only.

Steve: Yeah.

Leo: I've not had an issue, but certainly do worry about the potential. Thanks for the

terrific show and information.

Steve: So we'll get to LifeLock in a second. I wanted to say that, unfortunately, the way the bureaus operate is...

Leo: Independently.

Steve: Independently. And a credit-granting organization normally has a relationship with only one. So, for example, a bank or a car dealer or loan shark or whoever, I don't know. They'll have an account with Experian, or with TransUnion. And so they always go to TransUnion in order to check someone's credit. That's what they do. So unfortunately it sounds like Winfield is hoping to sort of, like, cheap out his solution by only freezing one and spending $10, or at least in California, rather than three and spending $30 total. But, unfortunately, you really do need to lock them all. So tell us about LifeLock, Leo.

Leo: So I've actually been a LifeLock customer for years. In fact, I bought it for my kids, too, because a kind of worst-case scenario on identity theft is that your kids' socials get compromised because that's a lifetime; right?

Steve: Yes. We're so old, it doesn't really matter that much; you're right.

Leo: Yeah, at this point I don't need good credit.

Steve: Well, we're not buying that much.

Leo: I don't need good credit as much as, you know, they have a lifetime ahead of them of credit that they're going to need. LifeLock is sometimes controversial. I don't think it's controversial. I explain where the controversy comes from.

Steve: In the old days; right? Because I think they overstated what they were doing, and they got slapped for that.

Leo: I don't think they did.

Steve: Oh.

Leo: So here's what happened. What they chiefly did was they would put fraud alerts on your accounts, was one of the things they would do immediately. And Experian and Equifax and TransUnion didn't like that so much because, if there's a fraud alert on your account, they can't issue credit cards. You won't get those credit card offers in the mail, and that's a big revenue source for the credit unions.

Steve: Wait, I can stop those offers by putting a fraud alert on my account?

Leo: Absolutely.

Steve: Ooh. That might be a good thing.

Leo: So they used to do it. So the credit unions, I think, in my opinion, there's some evidence for this, but it's, you know, it's murky because nobody's saying what happened. But in my opinion the credit unions, and we've seen this happen in other cases, ganged up on and got states' attorneys general to sue LifeLock. For mostly - and the FTC, for exactly what you said, overstating their capabilities. I don't think they did. However, they got enough judgments against them that they have and I know this because they've been an advertiser on the radio show in the past, and I think, no, we never had them on the podcast. But I happily endorse them because I know they work.

But they have gotten so gun shy, they can't say anything anymore. So their ads are horrible. I keep telling them, guys, nobody's going to buy this product because they cannot tell you what they do because their wrists are tied by these various judgments. So what they've done, which I think is fantastic, they bought one of the biggest, if not the biggest backend providers for credit cards that does the analytics that credit card companies use to detect fraud. Credit card companies use this company that LifeLock owns to do fraud detection. So as a result, LifeLock has jacked into the flow of credit card transactions from almost everybody, basically. And so they notify you in the same way that they notify Visa and MasterCard.

Steve: Ah, that somebody's applying.

Leo: Saying we know what's going on. Because they can no longer put the fraud alerts on. They were stopped. And so but they do other things. They do analytics that I think are really very, very valuable. It's controversial, it's not cheap, and I think one complaint is, oh, I don't want to spend all that much money, and I would understand that.

Steve: And it is an annual fee; right?

Leo: It's an annual fee. I do it because I'm a public figure, and so presumptively may be more of a target than others.

Steve: Be a target, yup.

Leo: So, by the way, I've never had a problem. And I've found them to be very, very good. They will do other things. You're also buying insurance to help you if you do get identity fraud and things like that. But I believe that they in fact do exactly what they say they do. In fact, they do more because they've been hogtied by these

judgments against them, so they can't really tell you what they do. I keep saying, you guys, you've got to tell people about this. "We can't." It's like, but this is great. "We can't." So really they became an enemy. In my opinion, what happened is they became an enemy of the credit unions, the credit reporting - not credit unions, the credit reporting agencies.

Steve: Right, right.

Leo: Because they were effective, in fact. And so they, you know, I think - let's put it this way. I use it. I can't prove that I need it, but I've never had a problem. They've been very quick. I check all the time.

Steve: And is it, like, 49 bucks a year, that kind of thing?

Leo: Yeah, it may be more than that because I have the kids involved. It's not a huge amount. Maybe it's 150 with the kids. I can't remember exactly. And you can log in. You can see a lot of information that you wouldn't see otherwise. I feel like they do the right thing. It's 10 bucks a month, somebody's saying, Packrat. And I'm not sure I would recommend that for everybody.

Steve: Right.

Leo: If you're willing to take the time to pay attention and get your credit - you can get an annual credit report for free. If you're ever turned down for credit, you could get that. If you don't want to get LifeLock, you need to do all that. You need to get your credit report every year. You need to check. You need to be proactive. You need to have the great time calling the credit reporting agencies. They've automated that now. There's no human. And it's not a very pleasant process. So if you could do a credit freeze, do it. But they really don't want to do it because it costs them money, in effect.

Steve: Yeah.

Leo: If you've had problems, though, you can get it. So LifeLock Ultimate is $30 a month; the base package is $10 a month. I think I'm doing it all. I can't remember, though, exactly what I'm doing.

Steve: Why are we not surprised? Well, at least you're not doing the gold Watch, so, you know...

Leo: No, no. But I feel like I am, I could be a target. And so...

Steve: No, I agree, yeah.

Leo: There are certain things that I do that I don't necessarily recommend everybody do. And that's...

Steve: And your nature is to be very open and sort of forthright and, oh, here's where I am.

Leo: It was on this show that I accidentally flashed my Amex card on the screen. That worked out well.

Steve: I never tell anyone I'm going to be out of town until I get back, and then I say I was. It's just, you know, standard security protocol.

Leo: Yeah. I think if you do the freezes it ends up costing you so much money that you might as well just do LifeLock, frankly.

Steve: Well, I did, I froze all three for $10 each, and I'm really happy I did that.

Leo: For how long, though? Like every three months you have to do that?

Steve: No, it's ever, period. You pay it once. You pay it once, and you're locked until you need to unlock it.

Leo: That's good.

Steve: Yeah, for me that's the perfect solution. At my point in life I'm not buying anything that I need credit for. I don't want anyone messing with my credit. So I just locked all three of them. It's a one-time payment of $10 in California. It varies by state.

Leo: It's the kind of thing that we can do because we're not applying for credit cards, buying cars, buying houses. If you're doing a lot of stuff like that, it's a pain, kind of a pain.

Steve: Right, because then you need to pay, you need to then do $10 for an exception to that, and that starts to add up.

Leo: Right. Freeze lifts vary by state, and if you have to pay when you're shopping for credit, it's worth it. Okay, even if you have to pay. Just a planning issue, says Mark in our chatroom.

Question 2. That's a great question, by the way. We've never really addressed that, so thank you. Jesse in Easton, Maryland. Some input on HTTP/2 mandatory TLS. And

if you listen to this show that all makes sense: The Packet Pushers podcast recently covered HTTP/2, and their host mentioned that mandatory TLS in HTTP/2 was removed from the spec because of concerns from ISPs it would impact their ability to provide caching as well as supercookies if all web pages were encrypted. And the IETF working group had initially proposed mandatory encryption. Just wanted to know your thoughts on that. Love the podcast. Been fans of you and Leo since the TechTV days, and SpinRite user for 10 years. Remember, I interviewed Vint Cerf, and I asked him if there's anything he would change. He was the father of the Internet. And he said the one thing we would have changed, we would have built in encryption into all these protocols.

Steve: Yes. Yes. It would have been difficult way back then because it was, I mean, well, first of all, we're still trying to get it right today. I mean, people are still, like, oh, no, you can't do compression because there's a BEAST attack. And if you change, if you twiddle something that we see how it compresses differently, then we can reverse engineer it. I mean, it's just staggering how difficult some of these things are to get right. But I just thought it was interesting.

I liked this question because it was, you know, I made the point of noting when we were talking about the evolution of SPDY into HTTP/2, that whereas SPDY was exclusively encrypted, we lost that when we went to HTTP/2. And assuming that Jesse knows of what he speaks, and I assume he does, it certainly does make sense that there are still people who have not given up on unencrypted web connections. ISPs would like the ability to cache in order to reduce their bandwidth.

But, I mean, anybody who has that as part of their business model today, or supercookies, for example, needs to absolutely give that up. They need to, if nothing else, plan from this moment on that connections will be opaque to them, and do whatever you have to - add routers, add bandwidth, but plan for the idea that your customers will be connecting directly to the servers, and you're not going to be able to see into it. Because that's the way it's going to go, whether it's in the spec or not, whether HTTP/2 can optionally be unencrypted. I mean, it makes sense that it should be because it'll be really fast. I mean, we saw that adding encryption and this compression and the binary protocol that is added in HTTP/2 is so - makes the system work so well that it completely removes the overhead of encryption. So if you had no encryption overhead and had HTTP/2 unencrypted, that would be even faster. And browsers are still going to support it. Web servers are still going to support it. So there'll be a place for it.

But more and more websites are just going to be encrypted because users are going to want that more. So, you know, the web is going encrypted, like it or not. And I think everybody pretty much likes it except those people who are, in one way or another, counting on it in some model not being encrypted. And that's of course the NSA and ISPs. And Verizon, who wants to put a supercookie on your stuff as it heads out on to the Internet.

Leo: Kyle Day in a Dallas, Texas - I guess a typo. [Silly accent] In a Dallas a Texas. He wonders, Steve, how could SQRL be used for multiple account logins to the same domain?

Steve: Sorry, Kyle.

Leo: All the usual praise for you and Leo. Seriously, your podcasts are worth more than free. Well, you know, they're free because we've got great advertisers, so you don't have to pay for them, but they do. I have a question about your SQRL solution. I've listened to all your podcast episodes. I never quite picked up on one thing you may have already covered: How does SQRL handle logging into multiple accounts on a single domain? For example, with my current password manager, I have both my and my wife's logins saved to iCloud.com. Could be any domain, of course. Using my current password manager, I would navigate to iCloud.com, select which login, mine or my wife's, that I want it to populate and log me in with. I do the same thing with Google. I have several Gmail accounts.

Steve: Right.

Leo: I think eight. So I get to choose. But how would SQRL handle multiple logins to the same domain? As I understand it, the single SQRL identity is what identifies me to the website's server. Is it possible to identify as someone else, or maybe a second account that I own? I have a PayPal business and a PayPal personal account, but they're both accessed at the same domain. They're both my own personal accounts. How am I going to manage multiple SQRL identities to accomplish this? Thanks. Kyle.

Steve: This question caught me because just in the last week we finalized the way of doing that. There are sort of two different concepts in SQRL. What SQRL promotes, as everyone knows, is the idea that one identity can be used, potentially for your entire life. That is, you could very likely create a SQRL identity with GRC's client and export that to your iOS or your Android or your Mac or wherever, or create an identity on any of those, and cross-export them, the point being they would all have the same identity so that you would be known as the same pseudonymous entity at any site, no matter which one of your devices you were using. But this is per domain.

So this is what Kyle is asking, is what if I want two accounts, if I, I myself, my identity, want two accounts at the same domain? And what we have is a - we're still not sure what we're going to call it, a sub ID or an account ID or an alternate ID. The idea is that remember I was just saying earlier that every single time you authenticate, the first time you give it your full password, and then subsequently you can give it a shortcut. We call it the password hint. And if you did that, you'd just get your normal sort of primary account at that domain.

If you wish to appear for any reason you have, we're not telling you how or why, you want to just, for some reason, one time you want to appear to be someone else, or you do want to maintain sort of formally additional accounts at the same domain, there's a button on the dialogue that just says "Account." And so you type in your shortcut, do do do do, hit the Account button, and then anything you want. You could number them, one, two, three. You could do A, B, C. You could use your own little acronym for the second account and third account. And there's no limit.

What we do is we take whatever you type in after you press the Account button, and that gets added to the hash. So it creates an infinite number of alternative identities for your primary identity at that domain. So it just expands it infinitely. And I'll just mention that this is different than multiple users sharing a computer because Kyle's question was a little mixed up. For example, he talks about he and his wife. In the two different people

mode, SQRL clients will support multiple, like, people identities.

So in a household who had a shared computer, you could have Mom, Dad, Bobby, and Susie, and their client would show those four identities. And so they would choose who they are and then just normally log in under that identity. And then all of those identities, if they wish to, could have multiple accounts under them. So, I mean, we got that. That's in there, too. And it's going to end up being very simple and easy to use and give people complete control and freedom.

Leo: Cool.

Steve: And we'll just hope. We'll get it done, we'll show it to the world, and we'll see what happens.

Leo: More and more it's starting to feel like it's something that needs to happen.

Steve: I know. We needed this a year ago, but it's better late than never.

Leo: The threat to you is that somebody like Microsoft or Google or some consortium comes up with some solution which would not be as good, but it would be some solution.

Steve: There is FIDO.

Leo: There's FIDO, yeah.

Steve: FIDO is, you know, there's that. And the problem with FIDO is they didn't know, or, yeah, well, actually they didn't know, I know that they didn't know because when I ran into Brad Hill at the DigiCert conference, he was stunned by the crypto I was using, which allows me to not require the server to keep my credential. That's what FIDO does which is so crazy is you give your credentials to the server to hold for you. Then, when you go there, you ask for them back. And then you authenticate against them. It's like, okay, I guess, okay. Anyway, SQRL doesn't need that. So, which is why Brad was really impressed and said it was the most well-thought-out authentication protocol he'd ever seen.

So it's certainly the case that FIDO could happen. But FIDO is device-tied. And even though it's technically an open spec, only one company has ever managed to write the FIDO protocol because it is so complicated, this Nok Nok Labs. They're the only people who have it, and they'd like to make money licensing it. And so when you hear that, like, Microsoft is going to support it under Windows 10, yeah, they licensed the code. Instead, with SQRL, it is really, I want to say "dead simple," except even simple turns out to be complicated. But we've got, like, 12 or 13 people, I mean, there are test servers up and running, and people are in the process of following my implementation and verifying theirs against mine. So, I mean, we'll see what happens.

Leo: Somebody said all you have to do is get Tim Cook to announce it and say it's magical, and you'd be done.

Steve: That would make my day.

Leo: Kostas Kritsilas in Calgary, Alberta, Canada wonders about RowHammer: In the last episode you discussed RowHammer, a vulnerability that researchers at Carnegie-Mellon discovered, where they could flip a bit or bits on an adjacent row by repeatedly accessing it between refresh cycles 64ms apart. Additionally, you said the Code Zero group within Google had weaponized it. My question is this: In order to do this, would the code need to have - would the code not heed to have - would the code not heed, but I think he means "need," to have direct control over the read/write operations, and perhaps some level of knowledge exactly as to when the refresh cycles are to occur at the hardware/chip level? That being the case, would it not be fairly difficult to take control of this, given there are multiple board chipsets, and only two manufacturers, Intel and AMD, but many versions of the chipsets? Or do all chipsets basically operate pretty much the same way? Kostas.

Steve: So I was glad to see this because we were up against a hard out last week on Thursday. We had Tech News Tonight, or Tech News - what do you call the morning one?

Leo: TNT, Tech News Today.

Steve: Today. And so I couldn't spend as much time on RowHammer as I wanted to. There are a couple points that, as a consequence, I left out. First is, well, since then there's already been some news. Linus has submitted a change to Linux which will remove that visibility into non-privileged applications port mapping. I mentioned that last week as being crucial to this. The application has to have access to its own page table in order to know where to hammer in order to change a row. And so the quickest remediation for this flaw is to take that away from non-privileged processes, and that's been submitted into the official Linux tree.

The other really tricky thing is, exactly as Kostas says, is that in order to hammer the row, you have to bypass everything the system has done to keep this from happening. That is, I mentioned at the beginning of my discussion this last week how slow dynamic RAM has remained. It just - it cannot get sped up in the same way that all of our static silicon processing has. As a consequence, processing speed has just shot past it. The only thing engineers have been able to do is caching. And of course everyone is familiar with L1, L2, L3 cache and so forth. There's even cache in the RAM chip itself. That is, when you read a row, the results of that read are stored so that, if you read other things from the row, it won't read them again. And rows are typically about 64 bits long in today's dynamic RAM. And so the idea is that every single bit of caching has to be bypassed in order to hammer on the chip at full speed.

The way they do this is there is an instruction that the Intel has that I'm sure AMD must. I'm trying to think whether the ARM architecture does. I think that the ARM doesn't, but Intel does. It's a non-privileged instruction. So one question people had was why was, I think it's cfflush, or maybe just cflush, cache flush. But so at the instruction level, the code running basically is in a tight loop. All it's wanting to do is to pound on the two rows

back and forth on either side of the target row. And so it does a write to one row, then it - or does it do a read? I can't remember. Then it flushes the cache in order so that - because what happens is normally when you are - when you're reading from dynamic RAM, you bring a block, a so-called cache line, which is like 64 bytes now, that bring a whole chunk in at once. And so if you were - I think that they are reading. And so if you're bouncing back and forth between doing reads, you won't be going back to the DRAM every time. You will instead be pulling it out of any of this pipeline of caches, the L1, the L2, or the L3 cache. And so that'll be done very, very fastly - very fastly. Very, very quickly, excuse me.

The reason, though, that you may sometimes need to explicitly flush the caches, that is, tell the caches to forget what they have, is if you had two processors sharing memory, and these processors might have processes running on them that are sharing a region of memory for, like, sharing data. So one processor might put something in the shared region. The other one would read that out. Well, before it does that, it needs to flush its local cashes so that it's actually reading the physical memory. Otherwise it'll just be reading out of cache, and so you can't trust that shared region of memory as a communications mailbox unless each processor that is about to look removes any memory it has of what it saw there last, so it actually pulls it from the physical RAM. And anyway, so pulling it from the physical RAM is what it turns out disturbs the adjacent line or the adjacent row in the cache.

And the last part is none of this works if you have ECC or even parity check memory, which is the other point I didn't have time to talk about last week, which is, for example, in server-class machines, you will often have error correcting memory specifically to absorb these kinds of problems. If a bit flips, the ECC works on a different fashion, but similarly to the way it does on a hard disk, where it's able to correct a single-bit error or reliably detect a two-bit error within that little chunk of, you know, it's like eight bytes of memory that it's guarding. And then there's another, an additional byte to protect that and provide the ECC data. So if you are a server-class machine with ECC, you never have to worry about this. And even if you just went for parity memory, it'll catch single-bit flips, as well, though it would not detect two-bit flips because that brings the parity back again.

So, and the last point I made was that, in Google's testing of, was it 24 laptops, they found that they were able to induce these errors in a usable fashion in half of them. And that's why I made the comment that you could imagine this being a part of a toolkit that an attacker would have, where they would have identified which laptops were potential victims, and if they were like, if their target happened to be one of those known vulnerable laptops, then they could try the RAMhammer attack on it and see if they were able to get control.

Leo: Right. Half is not a very - or maybe that's better than nothing, I guess. It's not...

Steve: Well, and this is a new problem. We didn't have this more than two years ago. So this is - we're introducing this new because we keep making the DRAM more dense and the cell sizes smaller, and therefore the signal is less large compared to the noise. The so-called "signal-to-noise ratio" is unfortunately dropping.

Leo: Sam Abuelsamid - and by the way, I should know your name, Sam, because I

see you all the time, Sam Abuelsamid in Ypsilanti, Michigan. He says: Terraced batteries, nothing new. As nice a piece of engineering as the new MacBook is, Apple did not invent the stepped battery. Motorola and LG both have used similar batteries in their phones for years. You can see it at AndroidCentral.com.

Steve: So, okay. Thank you. I'm...

Leo: There's a lot of those. Oh, you know, they didn't invent that or this. Apple doesn't, you know, Apple doesn't have to invent new stuff. It just is an interesting use of it.

Steve: Well, and, for example, we know that they acquired the fingerprint scanner people.

Leo: Right, AuthenTec.

Steve: But, boy, they sure turned that into something far more useful than the fingerprint scanner people ever had.

Leo: You still going to get a MacBook?

Steve: Oh, yeah. I'm going to - yeah, yeah. It's just - it's too nice. I'm a little worried about the keyboard.

Leo: Ah, me, too.

Steve: I've been reading reviews.

Leo: Yeah.

Steve: Yeah. And they say that it's a little, you know, the little rocky, high...

Leo: There's not much travel.

Steve: Exactly.

Leo: Yeah. I am, I'm going to let Lisa get one and watch. Maybe you and Lisa and watch.

Steve: I think I won't preorder one. I'll wait to go and look at in the store. Wait, when will they be in the store?

Leo: April 10th.

Steve: They won't be in the store till the 10th?

Leo: Well, nobody's...

Steve: I wonder if I could see one before.

Leo: You could play with the Force Trackpad now because it's in the 13" MacBook Pro. But I think you're going to have to wait till the 10th.

Steve: I think I'm going to love that regardless. That just sounds like a complete win.

Leo: Everybody's raving about that.

Steve: Yeah, the haptic feedback trackpad.

Leo: It's also a slow processor. I'm not - I'm really not sure. You can get the haptic feedback in other ones.

Steve: I'm not doing 3D modeling or DNA research or anything.

Leo: No, that's true. You know what I got?

Steve: I'm checking my mail.

Leo: You know what changed my attitude on all this, I got the new Chromebook Pixel.

Steve: That's the Pixel 2 or the, right, the new Pixel.

Leo: Pixel 2, I guess, yeah. Two years down the road it's much faster, snappier. Boy, what a gorgeous screen. And it does do everything I want, pretty much. There's a couple things. I can't do photo editing. But again, the MacBook, I might not want to do photo editing.

Steve: Does Apple have their stuff on Android yet? Like their Office Suite stuff?

Leo: Yeah, there's web-based versions of Microsoft Office and iWork, Apple's solution.

Steve: I'm just thinking of Jenny. This might really be the right thing for Jenny.

Leo: Boy, from a security point of view, it's incredible. First of all, it's got a TPM chip. They sign everything. You can't modify, you can't access many of the files. You can't just kind of browse around in the directories. It's locked down. And if anything bad goes wrong, they've got this Power Wash, it just wipes it all clean.

Steve: But it's not cheap, now, is it. It's a thousand dollars?

Leo: It's $999, yeah.

Steve: Yeah.

Leo: But you know what I realized is that, you know, I've played with every Chromebook. And by the way, I hated Chromebooks. I've hated the whole idea. Didn't get it. Until I started using them. And I'm starting to get it. And I'm seeing students use it. And I'm starting to get it. But then what happened was a lot of the reasons I didn't like it, because they were cheap hardware, and it was junky. And but you put it on nice hardware, I use the Chromebook more than anything else now. I'm using it all the time.

Steve: Well, and I think it makes sense for somebody who just wants to get their work done; who doesn't want to, like, live for the sake of having a computer, but just wants the computer to let them do what they need to do.

Leo: You know what's wild, it has SSH. So I moved my RSA key over, my private key over, imported it into the SSH module, then deleted it.

Steve: Client, yup.

Leo: And now I can SSH to a Linux box. So I've got - anything I can do on a Linux box, I can do pretty snappily. So I feel like I'm got everything. I've got a chatroom in there.

Steve: There's probably an OpenVPN client, too, I would imagine.

Leo: Oh, I'm sure there is, yeah. Well, and Google also has a remote access solution that you can access any computer remotely. So if I need a Mac, I could access. I think it's getting better and better. Anyway, just - I know. I know. People are going, you're crazy, Leo. You need a real computer. A manly computer. That's actually the one thing I miss. I cannot really do programming. I can only do web-based programming. I can't really do real programming.

Steve: True.

Leo: You know, I can SSH to Emacs and run Lisp. But that's about it.

Steve: No. Then, I mean, your hair that has just come back would fall out.

Leo: I actually - I like Emacs. I used to be a VI guy. I'm kind of coming around.

Steve: Yeah, well, you know that when all of the texture of the left and right paren keys has been smoothed off...

Leo: You've been doing Lisp.

Steve: That you've been doing Lisp.

Leo: I love Lisp. I'm starting to come - I'm coming around on functional languages. Screw these imperative languages. Anyway, Mike Robles in Wauconda, Illinois wonders about anti-keystroke logger protection: Steve, long-time listener. I'd like to get your opinion of something called Zemana AntiLogger, particularly the free version, Zemana.us. According to Zemana, this software not only protects against keyloggers, but they also claim it prevents SSL intrusion - protects SSL, prevents MITM, and monitors fake CAs. It got good reviews. Have you looked at this? What do you think? By the way, thanks a lot for all you do to promote security awareness. Sincerely, Mike Robles.

Steve: So, you know, if it could wash the dishes and my car. I sort of tripped over this one because these are the sorts of claims which are difficult to really honor. For example, you know, okay, first of all, I should say this thing's been around for a while. This keystroke logger is, kind of this anti-keystroke logger functionality that it professes, is sort of funky. Because inside of Windows there's a long chain of processing that keys go through which is mostly a testament to the age of Windows. As one thing or another, or someone had a new good idea, it had to keep all the other previous good ideas and add this one to it.

And so, as a consequence, there's about 10 different API levels where you can access keystrokes. And they're all targets for attack where a keystroke logger could get itself. So their claim to fame is they insert something at the very end, or I should say they insert something at each end. They put something way deep down in a driver in the

kernel close to where Windows is finally so tired of processing this keystroke that it gives up, and it says, fine, here it is. And way up at the front, right underneath the keycap itself, so that it catches it before anybody has had a chance to mess with it. And then they say they encrypt it. So it's like a little encrypted link from the key top down into the kernel, just before, like, the original primal NT function gets it. And in between is gibberish.

So if any of these keystroke loggers link into this chain anywhere there, they're going to get gibberish. And, I mean, this is how they describe it. I think, okay, this is gibberish. You know, all the keystroke logger has to do is link in ahead of them off the top end or below them at the bottom end. It's not like they have some privileged position. And maybe the keystroke logger is smart enough to do that, or maybe that's just how it operates. So the strongest claim you could possibly make is we saw a keystroke logger once that we could prevent. Okay. Maybe two or three of them. I don't know. But not all of them ever. It'd be like saying, here's the antivirus solution for all the antivirus today and forever. Just install this, and you're good to go because we solved that problem. It's like you can't solve the keystroke logger problem. They're claiming to have done so.

And so this is sort of a - I don't mean to pick on these people. There's a class of these. They all sort of feel the same. It's like, you know, somehow they didn't invoke military-grade encryption anywhere in there. They avoided that catchphrase. That's also typically a sign of a problem. But anyway, so, Mike, maybe it's good. But it just feels like more gunk in my computer, you know, something, one more thing to go wrong, sort of like power windows. I don't know, I'm just grumpy about these things that say, okay, this prevents all of these, because nothing can prevent all of anything.

Leo: Wow. That's an epigram for the centuries. Maybe put that on your tombstone. "Nothing can prevent everything."

Steve: Nothing can prevent all of anything, yeah.

Leo: I tried, but nothing can prevent everything. Here's one from Chris in Tokyo. And he's got that question I had about USB Type-C: Steve, with BadUSB and the inherently insecure nature of USB looming, what do you think of laptop manufacturers, even Apple, using USB Type-C for power? As far as I know, nothing has been done on the specification side to address security issues. Do you know anything about this? If I were a bad guy, man, I'd be planning where to offer "free" power to commuters, travelers, and coffee sippers looking to top off their fancy new laptop batteries.

Steve: Yeah. And the industry is reacting similarly. BadUSB is still fresh in our mind, and we're now seeing, for example, famously, the MacBook that has only a USB connector. Which means you'll have to connect to USB in order to get power. And I was looking at the little, at my maglock, how it's got those four little gold dots. And it's like, oh, I miss you because all you could be is DC.

Leo: I miss you, MagSafe.

Steve: Yes, I do, because you're so innocent. Look, four little connectors. Instead, you

peer into that USB-C, and it looks like an octopus has just taken over. So...

Leo: The good thing is, as I've been told, the Apple power adapter doesn't have any firmware. So you couldn't actually put BadUSB on an actual legitimate Apple adapter. The issue is going...

Steve: Ah. And, see, of course. And that's not the attack. The attack is how many times have any of us, as people propose, ever asked to borrow somebody else's power adapter in a pinch, or plug into the seatback in the airlines.

Leo: Well, exactly, yeah.

Steve: Yeah, or, you know, or there's, like, power outlets all over the place now that are USB. And presumably at some point in the future they'll be USB Type-C. Which is why I said somebody needs to make a condom. And as soon as somebody does, let me know, and I'll tell everybody.

Leo: How would that work?

Steve: The idea would be, it would be a tiny little thing which has a male USB-C on one end and a female on the other. And all it has is the power wires...

Leo: Oh, smart.

Steve: ...going between them.

Leo: Oh, that's easy.

Steve: Yes, it is. It's trivial. Someone needs to make one.

Leo: You could make one yourself. You just solder it together. Of course. Because if there's no data, if it's just power wires...

Steve: Then you're back to, oh, look, it's like my maglock. It's like my cute little maglock.

Leo: That's brilliant, Steve. I wonder if somebody's selling that? If not, let's do it.

Steve: Yeah, like I said, as soon as someone - now everyone knows, make a condom. As soon as someone has, I'll tell everybody, and then we'll...

Leo: You should do it. You should call it the SpinRite Power Condom. Seriously. It'd be trivial. You'd just need a bunch of Type-C connectors, male and female.

Steve: Male and female.

Leo: Yeah, because you have to have an innie as well as an outie, and just wires. It's like how the power wires are. There's probably none.

Steve: Yeah, we're on Question 7. By the time this podcast is done, there will be a project on Kickstarter.

Leo: Kickstarter.

Steve: I mean, and the reason I'm not doing it is because everyone will, and it's really not, I mean, it's worth having one, but it's not worth taking the time to do it. There are lots of other people who can do plastic and...

Leo: You could have $5 million right now in your pocket.

Steve: The USB Type-C Power Condom. We need a condom.

Leo: Why didn't I think of that? Brilliant. How about this? Look at that. There's already one for USB Type-A adapters.

Steve: Ah, good.

Leo: It's called Charge Safely, the SyncStop.

Steve: Perfect.

Leo: Look at that. Protect your data.

Steve: Have them do one for the next one.

Leo: It's just - they call it a USB Condom. Oh, man. Yeah, they've got to be doing a Type-C next, of course.

Steve: No doubt, no doubt.

Leo: "A condom that stops trojans, what an irony," says Wistful in the chatroom.

Steve: I will mention that, with any, well, first of all, your new beloved Pixel is also USB Type-C.

Leo: Yes.

Steve: Right?

Leo: Yup. It's got two of them, one on the...

Steve: So those do.

Leo: One on each side, which is kind of cool because you can charge it either way. You could have video out either way.

Steve: Oh, that is very nice.

Leo: Isn't that nice? I wish Apple had done that.

Steve: I love that it itself is reversible. And now it's even side-to-side reversible. So that's double nice. And let's hope that we're not going to run across the Firewire-style problem that Thunderbolt also had, where it's a DMA-style interface. Presumably - I've not had any chance yet to look at USB. Presumably they understand that they need to be providing some protection. But again, remember that in the BadUSB case, it was the device you were plugging in that was the problem. So the idea would be, you would be plugged - you think you're just getting power, yet you're getting power, and there's also a drive hidden there which is screwing around with your computer without your knowledge and permission, which is why you need the power condom, to say I want power and nothing else.

Leo: Great idea. All right, Steve. Let's see. We've got a couple more questions before we wrap this puppy up here. And Question #8 comes to us from John Hughan in Austin, Texas. He's wondering about TPM. We were just talking about TPM in the...

Steve: Yup.

Leo: I don't know if the original Pixel had TPM. I bet it didn't. That's awesome.

Steve: They've been around for so long.

Leo: Yeah, it just adds cost.

Steve: They may have had it.

Leo: But, see, that's what you get when you spend a little more, like $999. You get things like a TPM module, which means...

Steve: Well, you get nice hardware.

Leo: It is nice.

Steve: You can't really do really nice hardware for 200 bucks.

Leo: No. The screen.

Steve: You know, you're going to get some green frog laptop.

Leo: Yeah. Steve, I was thinking about your point that having a device that can decrypt what you want decrypted makes a lot of sense and makes a system inherently - oh, wait a minute, I'm sorry. That having a device that can decrypt what you want decrypted makes a system inherently [in]secure because the key has to be on the system. That's what we were talking about last week. But in that case, why do we trust TPM to provide our security for things like whole disk encryption keys? If we conclude that TPMs aren't secure, are we using them for sheer lack of a better alternative? It stands for the Trusted Program Module; is that right?

Steve: Platform, platform.

Leo: Platform, that's it.

Steve: And so he adds, are we using them for sheer lack of a better alternative? Yes.

Leo: Oh, really.

Steve: Yeah.

Leo: But it is a secure store itself; right? I mean, you can't - it's hardware.

Steve: Well, yeah. But this was the point that I was making last week was that, in the

same way that a DVD player is hardware, we were able to reverse-engineer the keys. And there's a fundamental problem when you are relying upon a local secret to be kept. The secret is in this thing I'm holding.

Leo: You got it, yeah.

Steve: And so all I have to do is pry it out of there. And in fact, prying it out of there is what's often done. The lids are popped on these things.

Leo: Physically, wow.

Steve: Yes. So I did want to mention, which I didn't before, is that because that's a known problem, things that attempt to protect their physical security are made deliberately vulnerable. For example, I've seen systems where they use a long-life battery and then run fine wires through, like, the air, and then pot this thing in black plastic. So that, if you're trying to dissolve the plastic or etch away at it or file it away, you'll break a fine wire, and it'll lose power and forget what it knows. I mean, the point is a lot of effort has gone into the recognition that the physical security of a small piece of electronics has now become crucial.

And the key of the Trusted Platform Module is very similar to what Apple has done with the Secure, was it called Enclave? I can't remember what they called it, the secure element in the iPhone. And that is, it will never export its secret. You can ask it to use its secret in order to prove something or to obtain decryption, but it will never export the secret. So you give it something, it does the work and says here's the result. But you never get the actual work, only the work product. So the answer is yes. TPMs are better, I mean, as a local repository for secrets, which they're as safe as we've been able to make them. And the manufacturers of the TPMs make them literally brittle so that any attempt to tamper with them will fracture them and cause you to lose the secret.

Leo: That's interesting. And then you'd know - or if not, you'd at least know. Or whatever.

Steve: Yeah, yeah.

Leo: And I don't know if Google ever said there was a TPM in the original Chromebook Pixel, but a teardown reveals an Infineon chip that is likely to be a TPM module.

Steve: Yup. Yeah.

Leo: Why not?

Steve: You know, TPM has been around so long, yeah.

Leo: It's cheap. And again, for $999, you can do anything you want.

Steve: Yup.

Leo: Bubba Mustafa, who's waiting for DerbyCon 5.0 - I'm not sure what...

Steve: I had no idea what that was. I thought maybe you would know what the reference is [hacker security conference in Louisville, Kentucky, September 25-27, 2015].

Leo: No, no idea. Worried about SSL/TLS hijacking/proxy: Steve, after listening to the episode about Superfish and understanding that enterprises will proxy the traffic to do deep packet inspection and other security defenses, doesn't that mean the whole SSL/TLS model is fundamentally broken? Can't my ISP, Big Brother, Chinese, Wookies, whatever, pop in on one of the hops and play man in the middle? Does HSTS fix this? Granted, of course that's only if both sides support it. Can HSTS be proxied, as well?

Steve: Okay. So I did like the addition of Wookies as a possible attack...

Leo: Wookies. Never know when the Wookies are listening.

Steve: That's right. Okay, so I wanted to sort of just make sure we hadn't sort of gone overboard with Superfish. First of all, HSTS won't fix it because all that does is force you to use encryption. That's all it does, force you to use encryption. But if the encryption itself is broken in some way, being forced to use it, well, that doesn't matter because the encryption itself is broken. For any of this to work, your browser must trust that man in the middle. Your browser must trust that proxy. That's done in the case of Superfish because Superfish has installed itself in your computer and given it a certificate, that is, given your browser, you know, remember the 400-some certificate authorities. Well, now there's 401. And no longer is the Hong Kong Post Office number one on our suspect list. Now Superfish is in there also.

So the point is an ISP cannot use, cannot break SSL and TLS, nor can Big Brother, well, technically, except I'm sure that the NSA can mint any certificate they want to, like, on a whim. It has to be that they control a certificate authority, among all those that the browser trusts. It has to be. But not an ISP. An ISP would have to require that you load a certificate in your browser in order for your browser to trust them. And I have mentioned that it's a bit of a chill that ISPs might in some horrible future become ballsy enough to say, oh, yeah, sign up for us, and here's how you install our certificate in your browser so that we're able to protect your security.

So if that day comes, oh, that's going to be a dark one, the idea that people might not have a choice but to install an ISP's certificate because what that would mean is the ISP is doing a man in the middle, maybe in order to do the kind of caching which SSL/TLS prevents, in order to lower their bandwidth. But I sure hope we never see that day. But anyway, for what it's worth, it's not easy to do this. You need to get the browser to trust

the proxy. And that requires adding a certificate that the proxy provides.

Leo: Finally, #10, Alan Figgins in Auckland, New Zealand had a question about Target-style security breaches and SQRL: Steve, I have been a listener since the day you netcast Episode 1 of Security Now!, a grateful user of ShieldsUP! since the Nineties, and owner of a SpinRite license since the early Noughties, so a huge thank you. In Episode 493 there was a question regarding SQRL and, in particular, whether there might be a potential security issue if, well, Target, to continue using them as an example, is compromised, and their SQRL database would be exfiltrated. Your response to that query was it wouldn't really matter too much, since the bad guys would only have gotten a list of users' public keys, and there is no particular value in those to any other person or website.

I agree with that statement, but I am wondering if there might be another security issue whereby a bad guy gets into Target's SQRL database and, rather than exfiltrating it, they replace a user's public key to be the matching half of a public-private key pair that they, the bad guy, have generated. Now the bad guy can log into Target's website using SQRL, sign the nonce with the private key that they have which now matches the public key held by Target, giving them access to the user's account. Is that correct, or have I misunderstood how SQRL works?

Steve: Alan, you understand it perfectly well. And, yes. If somebody were able to alter a database of accounts, then there is no way to prevent that. And you could argue that there's theoretically no way to prevent that. That is, any authentication system would fail that level of breach. And in fact, if bad guys get in, it's not really even clear to me, like, that that's what they would do. It's like, if they have that level of control, to alter the account database of some target like Target, then they've got free run of the place. So you're absolutely right. SQRL won't protect nor - that would be like changing someone's password to something else. So now the bad guys can log in as them with a password that they have changed. Or, I mean, like any authentication system. If you change what it is that the destination service has that identifies a person as some other person, then it's going to identify the new people.

Leo: It doesn't matter if it's a password or a SQRL, it's...

Steve: Doesn't matter if it's anything. I mean, nothing even theoretically can breach that. So there was another - this is sort of along the lines of let's explore this all the way. And I'm glad to explore it all the way. We have explored that. And it's like, yeah, nothing can withstand that sort of attack. So it's not - there's nothing I can do. I mean, there's absolutely nothing. So I don't consider that a problem or even a characteristic of, I mean, it's sort of like the nature of authentication is something is identifying the accountholder. And if you change that, then they're going to be identified as somebody else.

Leo: Steve, we've done it again, 10 questions.

Steve: Yes, and Episode 499 is behind us.

Leo: Uh-oh.

Steve: Here comes 500 next week.

Leo: You know what that means. Next week we're going to do something. I'll have to get you a cake.

Steve: I don't eat cake.

Leo: Set off fireworks. I'll get you a meat cake. It'll be all protein.

Steve: Give me a carrot.

Leo: A carrot? You're easy. Yeah, 500 episodes. Isn't that great?

Steve: Yeah, love it.

Leo: Five hundred great episodes, I might add.

Steve: I'll have something to say next week.

Leo: I'm glad you're feeling better. I am so happy.

Steve: That's what we're going to do for 500.

Leo: Okay, good. Kind of a special little thing.

Steve: I have some feelings.

Leo: I'll eat cake. You watch.

Steve: Perfect. Just don't choke, don't spit it out when you hear what I have to say.

Leo: Uh-oh. It's not going to be your last episode; is it?

Steve: No, get some Kleenex.

Leo: Oh, oh, no.

Steve: No, no, no, no, no. No, it's not, not bad.

Leo: Okay.

Steve: It's just how good this is.

Leo: I'm glad you're feeling well.

Steve: Me, too.

Leo: We do this show, and I guess you don't want to miss the next one. We do this show every Tuesday. Now, it's been a little confusion because people say, did you start early? Well, the problem is we start daylight savings time early. Saving time.

Steve: We strange people in California.

Leo: Yeah, well, the U.S. We start it...

Steve: Oh, wait. Our daylight savings time changes at a different time than other people's daylight savings?

Leo: Yeah.

Steve: Oh, my lord.

Leo: So they haven't gone to summertime yet in Europe. So we moved the podcast without telling them, kind of. We are still at the same time as far as we're concerned, which is 1:00 p.m. Pacific daylight time. That's, of course, 4:00 p.m. Eastern daylight time. But it is a different time - actually it's now 1:30, isn't it, 1:30 and 4:30 daylight time in the U.S. But it's a different time UTC because UTC does not change. If only we were UTC. It does not change. And you need to calculate the offset from UTC to your local time, and then this will all make sense. So, just so you understand, we are at 2030 UTC. So you have to figure out your own offset, from now on until we go - we move. That's the problem, we did, we moved on you. And let's just blame the stupid - blame Ben Franklin. We don't need it anymore, but we got it.

Steve: It's annoying.

Leo: It's annoying. It doesn't save energy. People die as a result of it.

Steve: Four heart attacks.

Leo: I almost had a heart attack. And I know I have crappy sleep. I think you got sick because it just changes your circadian rhythms for no reason. It doesn't save energy. The farmers don't like it, and that's a canard. There's no value to it whatsoever. The reason it doesn't save energy is it's a push. We use more air conditioning and less fuel oil or vice versa. So it's a push. So there's really no value to it. And I do predict that sometime in our lifetime this time change will end. I really think so.

Steve: I don't know. We're pretty old.

Leo: Well, that's how optimistic I am.

Steve: Okay.

Leo: I'm just an optimist, Steve. As Alex Lindsay says, I just want the sun overhead at noon. I don't care about anything else.

Steve: How about sometime during this podcast daylight savings time will change?

Leo: That'd be good.

Steve: That'd be good.

Leo: Yeah. Thank you, Steve. Always a pleasure. You'll find 16Kb versions of this show, show notes, transcriptions, everything you need at Steve's site, GRC.com. You'll also find SpinRite there, the world's best hard drive, maintenance, and recovery utility, and lots of freebies. And everything you need to know about SQRL so you can implement it on your website. If you can't watch live, and you want video, we have that, too. We do high-quality audio and video at our site, TWiT.tv/sn. It is also available wherever podcasts are aggregated, including iTunes. You can get the TWiT apps on almost any platform and watch that way, as well. Thanks to our third-party developers for those. But whatever way, you don't want to miss an episode. And if you're going to miss an episode, you definitely don't want to miss next week's Episode 500 of Security Now!.

Steve: Ooh, boy.

Leo: And one last week to vote in the Podcast Awards. Make sure you vote for Steve. PodcastAwards.com. Vote early, vote often and often and often. Vote all the time. Don't vote for any other show. Don't dilute your vote. Vote for Steve.

Steve: There is no other show.

Leo: There is no other show. There's one show.

Steve: Even the people competing with us in that section have voted for us.

Leo: There's no reason you shouldn't win this every year. But I guess we have to go through the motions. So please vote.

Steve: I just decided this one I want. Then I'll be quiet from now on.

Leo: Well, that's kind of how I feel. It's like, you know, I could have it every year because we have such a devoted audience, and they're very active, and they participate, and they're big.

Steve: It's a great audience.

Leo: There's 70,000 listen to this show. If each and every one of you voted every week...

Steve: Yeah, we'd break their web server.

Leo: It'd be like, okay, well, let's see who - are you in the technology, or what category?

Steve: Technology.

Leo: Well, let's see who won technology. Let's see. Oh, here's a podcast with 300,000 votes. What's the next biggest one? Oh. I guess he won. Thank you, Steve.

Steve: Thanks, my friend. See you next week.

Leo: See you next week for Episode 500 of Security Now!.

Steve: Yay.