The iOS 11 Security Tradeoff

Description: This week we discuss the details behind the "USB/JTAG takeover" of Intel's Management Engine, a rare Project Zero discovery, Microsoft's well-meaning but ill-tested IoT security project, troubles with EV certs, various cryptocurrency woes, a clever DNS spoofing detection system, a terrific guide to setting up the EdgeRouter X for network segmentation, last week's emergency out-of-cycle patch from Microsoft, a mitigated vulnerability in Apple's HomeKit, Valve's ending of Bitcoin for Steam purchases, finally some REALLY GOOD news in the elusive quest for encrypted email, a bit of miscellany, some closing-the-loop feedback with our listeners, and a look at the security sacrifice Apple made in the name of convenience and what it means.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-641.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-641-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson's here, the penultimate episode of 2017, and oh so much to talk about. We've got lots of security news. There's an out-of-cycle patch update from Microsoft. That usually means something's going down. Some iPhone exploits. And at the end we're going to address this issue of iPhone security. Is it getting better, or is it getting worse? All coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 641, recorded Tuesday, December 12th, 2017: The iOS Security Tradeoff.

It's time for Security Now!, the show where we protect you and your loved ones and their security and teach you a thing or two with Mr. Steven Gibson, our hero of the hour. He is the man in charge at Gibson Research Corporation, GRC.com, creator of the first spyware, antispyware tool. He also...

Steve Gibson: Yeah, make sure you say "tool" and not...

Leo: Yeah, tool, anti, anti, not spyware, antispyware. In fact, he's revealed much spyware, has exposed it in many cases. And he is also a great guy to have on your side in a knife fight. No.

Steve: Well, I'm not sure.

Leo: In a bot fight, for sure.

Steve: Maybe yes, maybe in a hacker fight or something, yeah.

Leo: Hey, Steve.

Steve: So as promised, we're going to finally discuss what we just didn't have a chance to dig into, which is this ElcomSoft kind of rant about a change that Apple made in iOS 11 security which their position is just collapses the whole security foundation and structure that Apple has built. And so we didn't have a chance to talk about that last week. I promised we would this week, and we will.

But we've also got now, thanks to Black Hat Europe happening, the details on that much-discussed here USB, or JTAG over USB compromise of Intel's Management Engine. We now know the details of that. And remember for weeks we've sort of been scratching our heads saying, well, we're just going to have to wait to find out because it can't really be as bad as it sounds. And it isn't. I mean, what they've done is true; but, boy, did they have to jump through hoops. We've got a rare Project Zero discovery from the Google team. Microsoft has a well-meaning but ill-tested IoT security project that I want to touch on because maybe we'll hear something from it in the future, but I'm not sure.

Some interesting troubles with EV certificates, which we all know I'm a fan of. But as they become more popular, of course people are going to start screwing around with them. We've got various cryptocurrency woes, a clever DNS spoofing, like spoofed site registration detection tool. A terrific guide appeared for setting up our favorite router, our favorite little consumer router, the $49 EdgeRouter X, for network segmentation. Somebody did a beautiful how-to walkthrough for how to take advantage of all the power in that cute little powerhouse router. We have to talk about last week's surprise emergency out-of-cycle patch from Microsoft that our Windows listeners may have noted. It's like, wait, what? It's not the second Tuesday. In fact, today is the second Tuesday.

Leo: I'm installing it right now. I just noticed it, yes.

Steve: Yes. We've got a problem with Apple's HomeKit, which thanks to the nature of the problem they were able to mitigate. But it does mean that we're going to go to 11.2.1, probably this week, it's expected. Valve announced that they're ending support for bitcoin purchases of Steam stuff, which is sort of interesting. We're going to talk about that briefly. Finally - and this is going to be of big interest to you, Leo, because you're a big PGP proponent; but also really good news for all of our listeners that really want email security, even though that phrase itself is an oxymoron. But we've finally got some really good news in this elusive quest for encrypted email.

Then we've got a little bit of miscellany, some closing-the-loop feedback with our listeners, and then I've read and reread and reread this ElcomSoft blog. It wasn't well written, but I think I understand what it is this guy's point is, and I can see what Apple did and that it does really create a problem. So I think 641, today's podcast, is going to be another goodie.

Leo: I'm just doing a little math here, multiplying, let's see, it's $17,431.20 for a bitcoin, and you have 50. You have $871,000 in your wallet. I'm just curious if you've made any progress finding it.

Steve: Haven't looked. I think I know where.

Leo: C'mon. C'mon.

Steve: I think I know where it is. But it's almost better to sort of have it and not know that I don't have it because...

Leo: Well, yeah, because if you didn't have it, that would be really disappointing.

Steve: I did do...

Leo: All you need, I found out, is the wallet.dat file. That's where the private key is stored. It's a small file. And what I was able to do is install the bitcoin core, the current bitcoin core, get it completely synced up with the blockchain 150 gigs later - unbelievable.

Steve: I know.

Leo: And then take the wallet.dat, the generic wallet.dat it created and move that out, and move my old wallet.dat in. And it recognized all - I have almost eight, what do I have, 7.85 bitcoin. It recognized it all. Unfortunately, I had the good sense - I was half smart - to encrypt it. You know, it has built in, the bitcoin wallet has an encryption capability, I presume strong encryption, I hope not, but I presume, and then promptly forgot the password. I didn't put it in LastPass, which tells me it's something I use all the time, like my PGP key. So I'll find it. I'll find it. But I have sitting here, right here in this little laptop, that's a mere $100,000. You've got almost a million dollars in there. Will you not look for it?

Steve: I've got some boxes in the way. I'd have to move them.

Leo: But you know the machine you used to mine it; right?

Steve: Yeah.

Leo: And you haven't erased that machine. So it's there.

Steve: So what happened was, it was on this little i7 cube that I built for the podcast.

And I just sort of had it next to me, and I left it running overnight, and I woke up, oh, what do you know, 50 bitcoins.

Leo: Fifty. Five zero, kids.

Steve: Five zero.

Leo: This was in the good old days.

Steve: That was when one successful hash gave you 50 bitcoin. And it wasn't hard. I mean, this wasn't GPU. There was no Freon cooling or steam coming off of it. It was just I left the machine running. So then I think I remember moving it to a different machine because I thought, well, if 50 is good, 100 would be better. So I kind of think I know where it is, and that I just sort of forgot about it. It's like, okay, fine. And then that machine stopped, that has like a RAID and a bunch of stuff, it sort of - I turned it on once, and it didn't boot. And I thought, okay, well, you know, and I have other machines. I haven't worried about it ever since. Now I'm kind of thinking about it a little bit more lately.

Leo: Steve. I don't know how well off you are. But I think even Bill Gates, knowing that there's a hard drive with $871,000 on it, would look for it.

Steve: But Leo, look what's happened. I mean, I'm not in a hurry, either, because this shows every sign of, you know, it staggers a little bit. It hit $20,000. Then there was some profit-taking that dropped it down to 15. Now we're back up to 17 something. I mean, believe me, I've got the bitcoin ticker on every device that I own. And every so often I go check in. It's like, oh, okay, that's nice.

Leo: So that's actually really, that's the first question is what, if you found it, would you sell it immediately for cash? No, you wouldn't. You'd hold it.

Steve: No, I wouldn't. No, absolutely.

Leo: So there's no urgency.

Steve: Yes.

Leo: No urgency at all.

Steve: I'll leave those boxes where they are.

Leo: Yeah, I kind of thought about that, too. Now, here's the one thing you might want to do. There are, as you know, two forks. There's bitcoin gold and bitcoin cash. And if you take your bitcoin and put it in an exchange that supports the other bitcoins, you'll get the equivalent amount in their bitcoin. Which I think one of them is $1500. So you've got $75,000 just free money. And I, see, I would - that's what I would do if I got access to mine is I'd get the bitcoin gold and cash and get out of it because who knows how long the forks are going to survive, and keep the main bitcoin.

Steve: Well, and of course the moment I realize this in a true currency, I owe the U.S. Internal Revenue Service half of this.

Leo: Well, that's what I thought. But a number of people have said, "Oh, Leo, don't be foolish."

Steve: Yeah. I really like not being behind bars, Leo. Doing the podcast and having bars up in front of my face, that would really put a damper on my security recommendations, I think.

Leo: Although, I have to tell you, if you are in a prison cell, you might want to get a Ring Video Doorbell for it.

Steve: Ah.

Leo: That way, if the jailer comes a knocking, you know it's him. This is not what Ring has told us to say about the Ring Video Doorbell.

Steve: There's a segue we will never repeat. You know that every week we have a picture. And sometimes we go a little - wander off the beaten path. But for anyone who's an Amazon customer or has been doing ecommerce, this just tickled me. And so, though this is not security related, it's just too fun.

So for those who are listening, we have a woman sort of in her middle age, kind of frumpy, with her shopping basket in her hand, and she's standing in the market, looking at the display. The upper shelf says "Customers who bought this item," and then it shows some bananas. And then down below, "also bought," and I don't know, looks like a jar of mayonnaise, some carrots, a loaf of bread, and some milk. Anyway, she just sort of looks bemused. She's like, what? And of course all of us are used to the nature of our transactions being electronic and the system knowing us. And I have to say, sometimes I find that information very useful when I'm trying to...

Leo: If it were my Amazon account, it would just have five more bunches of bananas below it. They always recommend stuff I've already bought.

Steve: Yes. Yeah, well, in fact, when the Healthy Sleep Formula was happening, people

have stopped talking about it now, although apparently still many people are using it.

Leo: They're all asleep.

Steve: I don't know, about 2,000 visitors. Yeah. When they would put one of the ingredients into Amazon, it immediately brought up all of the rest of the ingredients. And so many people sent me snapshots of that.

Leo: That's cool. That's cool.

Steve: Saying, hey, other people have been here, yeah. And like sometimes you'll want to know, like, what batteries this thing takes. So you'll select that thing, and then there's like, oh, people who bought that thing also bought these batteries. And it's like, okay. So it's sort of a shortcut for you. So certainly handy. Anyway, not about security, but I just got a kick out of this moving from the cyber world into the so-called brick-and-mortar world.

Leo: Yeah, yeah.

Steve: So, okay. We've been talking many times and for several weeks, ever since the guys at Positive Technologies informed the world that they had managed to get a hold of a JTAG interface over USB to, what was it, the last seven years of Intel motherboards and chipsets. Which was like, what? Because we explained the JTAG is the industry standard debugging interface which is a serial interface that allows you to halt the processor, single-step it, examine its registers, examine the state of the system, read and write to memory, I mean, it's everything. So the question was could Intel have actually been, possibly have been so irresponsible as to have deliberately exported this incredibly powerful debugging interface on their USB ports. And the answer is no.

So now, thanks to Black Hat Europe, which has happened, we've got the detailed research paper where they take us through what they had to do. And it was exhaustive and exhausting. First, we talked about this a while ago because we've been talking about the worry with the Intel Management Engine, the IME stuff, for some time because bugs are beginning to surface in that, and it is like ring -3. It's super powerful. You could do everything with it.

So the first breakthrough was that the firmware was compressed using a simple but powerful compressor known as Huffman coding, where you look at chunks of bytes, and you assign it a variable bit length token which is inversely proportional to how many times that chunk of bytes appears. So that is to say, the more times a run of bytes occurs, the shorter the token you give it; and the fewer something occurs, the longer the token.

As a consequence, you end up with, first of all, removing the byte boundary. You turn it into bit encoding, which is going to be inherently more efficient because you're not, like, having to round everything up to the next byte size. And the things that happen more often are encoded in fewer bits. This of course is what Morse did when the Morse code was created. The things most often sent in the English language were given the shortest representations, dot and dot dash and so forth; with the things that occurred less often

being more lengthy to send. Very clever.

Leo: Etaoin shrdlu. Right?

Steve: What?

Leo: Etaoin shrdlu. You would know that. I bet you would know that; right?

Steve: That's right. Right, right, right.

Leo: E-T-A-O-I-N S-H-R-D-L-U. Those are, in order, the most common letters. The only reason I know that is it's great for crypto, right, for...

Steve: Well, and you just passed your ham license not too long ago, also.

Leo: Yeah. I didn't need to know Morse code, though, thank god.

Steve: Oh, no kidding. That's gone?

Leo: Yeah, they dropped it. Oh, yeah.

Steve: Okay. So some people managed to reverse-engineer the Huffman tables, which had not been published. Once that was done, then the firmware could be decoded. The problem was, until relatively recently, the management engine controller was not x86. It was known as the ARC family. ARC is kind of similar to ARM, but different. It is a RISC family. It's IP. It's intellectual property in the same way. You're able to build basically your own controller. They manage and groom and evolve the architecture over time, exactly the way ARM does, but this is ARC. And so that's what Intel had historically been using. The problem is there are no really mature reverse-engineering tools for that wacky RISC engine, so hackers weren't that interested in digging around in there.

Well, Intel said, hey, why are we licensing this ARC thing when we've got x86 stuff coming out of our pores? So they switched the IME some time ago to x86. Well, that made it much easier for security researchers to reverse-engineer because now they had all of the mature code analysis tools that the research and the hacker community have developed over time, automatically breaking out subroutines, following things along, figuring how things are interlinked, giving you beautiful traversal graphs of how the flow of control goes. So that made it much more easy to do. So that allowed them to extract the firmware, decompress it, understand what it was being written in, and apply mature code analysis. They found a stack overflow vulnerability.

But Intel had a stack cookie, which we've talked about in the past. The idea is that you put little sentinels, they're called "stack cookies," on the stack, normally generated by something. And in fact there's a hardware random number generator that generates the stack cookies so they don't repeat. And the idea is that, if any buffer overflow occurs,

and the stack is overrun, it will stomp on the cookie. And so before control is returned to the data on the stack, the cookie is checked. It's validated to make sure that the stack has not been smashed. So props to Intel for really having a lot of protection down in a system that they would never have expected anybody to be digging around in because it's just hard, they made it so hard to get to. But it's there. So there's stack cookies.

And so then they figured out, okay, how to get around the stack cookie. But then it turns out the stack is nonexecutable. So again, props to Intel. They said, hey, stack is for data. Code should not be running on the stack. So mark the stack segment as nonexecutable. Well, one thing that's missing, and I'm not criticizing Intel because they went to a lot of extent in order to do this, is Address Space Layout Randomization. There is no ASLR. What that means is that all of the location of the modules of code are in known fixed locations. That means that they can avoid the problem of needing to execute from the stack by using so-called ROP, Return-Oriented Programming, where instead you jump to known code that is executable, being code, and get it to do the stuff you need.

So it turns out that there are file system components that are used during the boot-up process, the so-called BUP, B-U-P, that we've talked about before, that can be replaced to allow unsigned code to run. So they went through all of this in order to arrange to bypass Intel's across-the-board code-signing protection, the idea being that the system absolutely is designed not to let any non-Intel signed code to run. They figured out a way around that, got their own code to run, then managed to turn on a mechanism in the PCH.

The PCH is the Platform Controller Hub. That's the chip where this all runs, where the x86 is, I mean the x86 engine that runs all this. The way they explained it in their paper, the platform controller hub is god of the motherboard. It has direct access to about 50 different little itty-bitty devices all over the motherboard that are all about just motherboard level stuff, way below where the main motherboard processor is running. And so, if you can get into the PCH, into the Platform Controller Hub, and run your own code, you can do anything you want. I mean, it is supremely powerful.

And it turns out that all of those devices all over the place, including that x86 Management Engine Processor itself, support JTAG. And so by manipulating something known as the PCH Red Unlock, they were able to obtain JTAG debugging access to all of the devices on the platform, including the x86 Management Engine Processor itself, and then were able to export that over one of the USB interfaces to the outside. But only after all of that other work.

So I give them serious props for having achieved this. This is the kind of research where the result only makes the platform stronger. Certainly Intel will immediately move, if they haven't already, to shut this down and resolve the vulnerability that these guys managed to find. But they certainly had to work for it. And by no means does it mean, as we were worried, that anybody can walk by any of the Intel-based, Intel chipset-based motherboards made in the last, what is it, 10 years I think, since 2007, and just stick a thumb drive in and take over your machine. Not even close. It requires serious deep level surgery in order to make that happen.

So Intel really did a good job of working to prevent this from happening. They missed a little spot, so I'm sure that will be fixed in firmware updates to prevent it, and for sure moving forward. So anyway, that closes that chapter and pretty much resolves the worry we had. These guys did a great piece of forensic reverse-engineering and security work, which is how the system is supposed to work.

We have a couple more stories that are sort of similar to that. And in fact the next thing

I'm going to talk about was that last week Ian Beer, who is one of the Google Project Zero security researchers, tweeted something very tantalizing on December 5th, exactly a week ago. He tweeted: "If you're interested in bootstrapping iOS 11 kernel security research" - meaning, okay, if you're a researcher who wants to play with rootkit-y kinds of things. He said: "...keep a research-only device on iOS 11.1.2 or below." And then he said: "Tfp0 release soon."

Well, tfp0, that's a special nomenclature known to iPhone kernel hackers, stands for Task For PID, P-I-D, Process ID 0, which is the kernel. And so what he was saying was, again, like he was foretelling something coming soon that was going to be huge for the iOS jailbreaking and kernel research community. And he was saying do not move to 11.2, which is what happened last week. So stay at 11.1.2 or below deliberately.

Then yesterday he dropped the other shoe, saying: "iOS 11.1.2 now with more kernel debugging." There's a link in his tweet to the bugs.chromium.org, which is where the Project Zero repository is. And basically all the details for getting a kernel debugger, which he produced a proof-of-concept local kernel debugger, which runs on all the machines he has to test on - an iPhone 7, a 6s, an iPod Touch 6G. And he says that adding more should be easy. There is a link to what he calls the "async_wake" exploit attached to this, with full details. He worked with Apple to responsibly disclose this. This was closed in 11.2.

So this has been fixed. But this is significant for people wanting to poke and examine the internals of iOS 11 or, yeah, iOS 11 and earlier, even 10, because Apple, as we know, has been working frantically and successfully to lock down, to increasingly lock down iOS. The flipside is that researchers have also been locked out, as is necessary. And so for a while, because it's been increasingly difficult to get jailbreaks on iOS, what that also means is that valid research has been pushed out. And so there sort of isn't any way to have it both ways. You can't allow researchers in because there's no way to keep those secrets away from the bad guys.

So Apple, in order to continually lock down iOS more and more, the problem is that it prevents valid research which may discover and responsibly report problems, just as Ian has, and did. I mean, he found a way on 11.1.2, that was what we had two weeks ago, he found a way in. And we haven't had that for a long time. So anyway, in their most recent 11.2 update, Ian was responsible for I think it was - I had it here in my notes - a large portion of the updates. He is Project Zero's iOS specialist. Thought I had it here; I don't see it. Something like five of the last 10 updates were from him. So anyway, so the research community is excited because they've not been able to look in and poke around for a while. Maybe they'll find some other things that Apple has missed and responsibly reported that have not been fixed as this was.

Oh, here it is. I said: "In the latest security bulletin for iOS 11.2, five of the 15 iOS 11.1.2 vulnerabilities that were patched were discovered and reported to Apple by Ian." So he's been very useful and valuable to Apple in continuing to find problems and responsibly disclosing them, thus allowing them to continue, essentially, the attempt to lock him out, which they continually seem to be unable to do. But in the process...

Leo: It's funny that he's telling them his jailbreaks, and they're fixing them. He must just enjoy the game.

Steve: Yes, exactly. And, I mean, he really, as a part of Project Zero, the goal of course, Google's goal is not to only fix Google's things, but to fix everybody's everything by

responsibly reporting. And remember, they do start that 90-day clock. So if you don't jump on this and fix it fast, it goes public after 90 days because they're not screwing around. If they say, look, here's something bad, the clock is started. In fact, there have been cases where Tavis found something, remember, after one of his famous showers, reported it to the LastPass folks, and before he was finished toweling off they had this thing fixed. And so the 90-day clock didn't even get down to 89 before LastPass had something that Tavis Ormandy found fixed. So, yeah, I mean, hats off to Google for having this project. It just makes everything a lot more secure.

Now, that's all the doing it right. Microsoft has a project called Sopris, S-O-P-R-I-S. That is, that's the Microsoft Research branch. And the good news is that this project SOPRIS is exploring the very worthwhile idea of making future IoT security better by designing it in from the start, by integrating security into the underlying chip and architecture and everything, thus effectively completely eliminating its cost. That is, we could argue that the reason it isn't being done with light bulbs is that you can get a cheap processor that doesn't have any security in it that works, and the light bulb's able to phone home or do whatever it needs to do, and security is just not there. So their correct argument is, if we build it in, if it's just always there, then it'll get used because it's there.

So the bad news is, I would argue, and I do, that they're going about it all wrong. They proudly came out and created a hacker challenge. They organized it through HackerOne and called it the Project Sopris Challenge. And over on HackerOne they announced that the Microsoft Research Project Sopris team invites the security research community to test our most recent experiment, the Sopris Security Kit, by applying to the Project Sopris Challenge. And then there's a link in this announcement.

And they wrote: "A primary goal of The Challenge [in caps] is to broadly engage with the security community towards sharing and learning: 150 security experts have been selected to receive and test a Sopris security kit through The Challenge. The application period closed at midnight on April 14th Pacific Daylight Time. Applicants have been notified if they are accepted to participate in The Challenge by email by April 21st. Devices were then distributed by early May, with the challenge ending on July 12th. It is free to participate, and there are no fees for the hardware." So then I thought, okay, this is interesting. And so I dug a little bit deeper.

And so Microsoft then, Microsoft Security, boasts on their SOPRIS page: "Project Sopris Security Challenge. Thanks to all the skilled hackers who participated. Over 150 hackers pounding on their Project Sopris boards for 60 days. Microsoft Research Sopris Challenge completes as a great learning experience, with no verified exploits. Stay tuned for updates from the Project Sopris team as we continue to work with the security community to explore devices secured from the silicon up."

And so I'm thinking, okay, all that sounds good. So they released something. They gave everybody this information, the kits and things, and then said find some problems. Well, then I found a quote from one of the - it was Wired's coverage - one of the hackers who participated who goes by the handle HexDecimal, and said: "It's stupidly easy to hack most IoT devices, but this was very different. The chip was definitely built for security from the ground up. One of the noteworthy things would be" - wait for it - "the lack of information." And I thought, what? He says: "The board and its web server were very closed off, nothing that would hint at an exploit." He says: "I only started to get a foothold after decompiling one of the setup tools that came with it. But I never managed to find anything, and neither did anyone else in the challenge."

And so I thought, okay, what? That's the dumbest thing I've ever heard. Microsoft gave these guys an undocumented black box and said, "Try to hack it."

Leo: That sounds like fun.

Steve: Well, except that - yes, fun, except what you want is something completely documented, completely open, completely known. And that isn't what Microsoft gave them. Galen Hunt, who's the managing director for Project Sopris, says, quote: "The team was actually disappointed that the penetration testers didn't find any flaws." Okay, so give them documentation.

Leo: The ultimate security through obscurity.

Steve: Exactly. This exactly was pure obscurity. It's like, here's something. See if you can break it. But we're not going to tell you anything about it. We're not going to tell you what it does. We're not going to tell you what it supports. It's just like, what? So anyway, I would say give them total documentation, give them source, give them some schematics, give them everything because that's what the world will ultimately have. It's, I mean, it's exactly like what Intel did not do. They tried to hide their Intel Management Engine under layers of compression with undocumented Huffman tables and just buried it away. And so of course researchers managed to reverse-engineer it, as researchers are always going to do.

So it's absolutely ridiculous to give people a black box and say, okay, you have 60 days to crack this, and we're not telling you anything about it. Give them 60 days with the source code and see how well you do, Microsoft. Maybe someday, but not now. They'd rather parade around and say, wow, look, nobody can figure out why what we told them nothing about they were unable to penetrate. It's like, okay. Well, fine.

Leo: Steve "Tiberius" Gibson has been - somebody said: "Steve doesn't drink caffeinated coffee, does he?" I said, oh, yes, he does. Oh ho ho. That is not a misnomer when we said Steve's off to caffeinate.

Steve: I reused some Starbucks cups the other day. I made a latte here and took it back to my other house. And Lorrie saw the side of the cup that Starbucks had originally marked up, and it showed - it had a six in the window of shots of espresso. And it's like, yes, that's the hex, the venti hex latte.

Leo: That was literally an eye-opener for Lorrie; right?

Steve: Indeed.

Leo: What? [Crosstalk] what she's got into.

Steve: Yeah, she does. So extended validation is now the subject of people poking at it, as was almost predictable. Leo, go to stripe.ian.sh in Safari, if you've got Safari as your normal machine in front of you: S-T-R-I-P-E dot I-A-N dot S-H. And what you will see - or you can just click the link in the show notes, if you have them. What you will see...

Leo: Unfortunately, I don't have Safari. This is Chrome.

Steve: Ah, okay. Well, Chrome works, too. You should see an...

Leo: I see it says secure, yeah.

Steve: It says secure.

Leo: In green, yeah.

Steve: And probably does it say Stripe, Inc. in the U.S.?

Leo: You know, Google has done a bad thing. They now hide the certificate. Whoops. Clicked too many back. They hide the certificate from you, which I find very frustrating.

Steve: Yeah, we have talked about that, how they took that away, essentially.

Leo: Yeah. Let me see if I can get back to the page in search. All right, stripe.ian.sh. And I think I remember that I - what do I do? I click the settings? I can't even remember anymore.

Steve: Yeah. They have come back a ways to make it a little more visible. But anyway, under Safari, what someone who goes to stripe.ian.sh would see is a nice, green, centered in the URL, they don't even - Safari doesn't show you the URL, it just shows you what the certificate - and it says Stripe, Inc., and then, parens, (U.S.). Same thing in Firefox. I see this happy glowing green Stripe, Inc., as in the United States.

Leo: Yeah. It says secure via HTTPS, valid certificate. Let's look at the certificate. Stripe.ian.sh, Stripe, Inc. But wait a minute, it's "Ian." It's not "Stripe."

Steve: Correct. Exactly.

Leo: Now, does it have anything to do with the fact that it's Comodo?

Steve: Well, no. But he did get a Comodo cert. What this researcher did was he incorporated Stripe in a different state. And so that cost him $100. And then he purchased an EV certificate from Comodo for his valid firm, Stripe, Inc., but which has a name collision in a different state of the U.S. In the United States you can't have two incorporations of the same name and the same state, but you can incorporate the same name of a company in a different state.

So the point is, here is a social engineering hack on extended validation, which is like, oops. The problem is, if you're Stripe, Inc., you're well known. You're sort of famous. Somebody seeing this in Safari or in Firefox would just assume, oh, okay. And if you clicked on a link, and if you didn't take the trouble to inspect the URL carefully, and you really can't even - you don't even see the URL in Safari. But you have learned to trust the extended validation greenness. It says Stripe, Inc. So you immediately think, oh, I am at the right place. No, you're not. And in fact another researcher got an extended validation for the firm Identity Verified.

Leo: Brilliant. Brilliant.

Steve: Yes. And it's green. And it looks wonderful. And it means nothing, unfortunately, except that you went to a site of a firm who got an extended validation cert, Identity Verified. And it's all warm and fuzzy. And unfortunately, it doesn't mean what any user would think it means. So we're beginning to see some hacks against the underlying extended validation implied meaning, which kind of were inevitable, but it just demonstrates how difficult it is to do these sorts of things. Yes, we have EV certs. But yes, there are ways around what the EV is trying to say so that showing you Stripe, Inc. incorporated in the U.S. isn't really a guarantee that you're at the Stripe, Inc. you think you are, or is being asserted by the certificate.

So anyway, this has stirred up some dialogue in the CAB Forum with the guys that are active, our CA Browser guys, trying to figure out what to do about some of these things. So this is a good conversation to have, but it just demonstrates that just the fact that we're involving more humans in the loop for certificate validation doesn't automatically solve every problem. And we could also argue there probably is no solution to every problem.

So I ran across, in doing some research for a mistake that Apple made in allowing an app to get into the app store, I ran across one of the premier websites for managing cryptocurrency. It's called MyEtherWallet. And Leo, if you go there, and this time it doesn't matter what browser you use, M-Y-E-T-H-E-R-W-A-L-L-E-T dotcom, what you get is this big annoying clickthrough, and with a bunch of apology, saying "Welcome to MyEtherWallet.com. We know this clickthrough stuff is annoying. We are sorry." Then it says: "Please take some time to understand this is for your own safety. Your funds will be stolen if you do not heed these warnings." Which hopefully will get their attention and beg some more patience from them.

And then they say: "We cannot recover your funds or freeze your account if you visit a phishing site or lose your private key." And then basically they go through a long series of intercept pages, each of which is annoying to someone who doesn't like these things. But what I was immediately put in mind of is the problem I'm already understanding I'm going to face, I and the SQRL community, with SQRL because...

Leo: This is actually good because it does explain what the hell's going on. And I think a lot of people just launch into this without any idea.

Steve: Yes, exactly, Leo. So, for example, to give our listeners a sense, they say: "What is MEW?" which is their acronym for MyEtherWallet. And they say: "MyEtherWallet is a free, open source client-side interface. We allow you to interact directly with the

blockchain while remaining in full control of your key and your funds. You and only you are responsible for your security." Then they say: "MyEtherWallet is not a bank." So they try to draw the distinction.

"When you open an account with a bank or exchange, they create an account for you in their system. The bank keeps track of your personal information, account passwords, balances, transactions, and ultimately your money. The bank charges fees to manage your account and provide services like refunding transactions when your card gets stolen. The bank allows you to write a check or charge your debit card to send money, go online to check your balance, reset your password, get a new debit card if you lose it. You have an account with the bank or exchange, and they decide how much money you can send, where you can send it, and how long to hold on a suspicious deposit, all for a fee."

And then they differentiate themselves from all of that, saying: "MyEtherWallet is an interface. When you create an account" - and blah blah blah. I won't drag everyone through it. But "We do not charge a transaction fee. We never transmit or store your private key," blah blah blah. So they're working to educate the consumer who's got themselves all revved up over bitcoin and blockchains and so forth without any idea what they are, but they want some. They're trying to say, look, we want to help you, but we want to help you help yourself. So there's, like, a limit to what we can do.

And so they're attempting to explain the responsibility that doing this has because they don't want people to lose their money. But they want to explain that they are not a place where you can go for recourse. They can't get your money back. They can't help you if you get spoofed and somehow your private key gets grabbed by scammers. They can't help you.

And so of course I was reminded of this because of course SQRL is very similar. SQRL gives the user, requires the user to have responsibility for managing their identity. There is no one to complain to, no third party to say, oh, I forgot my password. Please send it to me in my email. No. That doesn't exist because, if it did, it could be hacked. And that's how identities are lost, and that's how Mat, we talked about years ago, lost his Twitter account, and then they got everything else and got all of his stuff.

And so I've been, as our listeners will understand when we do our podcast on SQRL, there's all kinds of - like we've done everything we possibly can to help people not get hurt and to recover from anything they do. But ultimately, very much as with bitcoin, the ultimate responsibility is the user. And we're seeing the consequence of that because people aren't used to that. They're not used to, like with your credit cards, wait, that's not my charge on my card. And the bank says, oh, okay, and they take it off. They reverse the charge. And so these guys, I really liked what they did because right upfront they said, "We're sorry for what we're going to make you read, but please pay attention to this because we don't want you to get hurt. But if you hurt yourself, we're sorry, but we can't unhurt you."

Leo: It's funny you should mention this because this is the wallet that was spoofed on iOS.

Steve: Yes, exactly.

Leo: Completely unrelated, but...

Steve: Apple, after all of this care and concern and worry, Apple allowed a knockoff version of one of the world's biggest crypto wallets, this one, the MyEtherWallet app, onto the App Store. And it went right up near the top of the most popular Ether wallets. It's an imposter. MyEtherWallet got onto the App Store. TechCrunch wrote: "The app rose to the number three spot" - that is, this fraudulent, this spoofed imposter app - "the number three spot in Finance of the App Store last weekend as part of the bitcoin frenzy that saw the Coinbase exchange top Apple's free download list in the U.S." So a huge amount of interest.

"However, in this case it's important to note that the app is not official. So users," TechCrunch wrote, "should avoid downloading it." The app developer of the spoofed app, who's listed as Nam Le, has three other apps with Apple. Two are panda fighting games. And he has no history of crypto or bitcoin services. So the legitimate creators of MyEtherWallet immediately posted a tweet saying that they have contacted Apple and asked to have it removed, and today it is gone. But so it was in looking into this story that I went over to the MyEtherWallet and was so impressed with the work, the degree that they went to and the care that they showed to try to help people protect themselves, yet here was a spoofed version of their app that got onto the App Store, and thousands of people downloaded it.

So unfortunately, I don't see how we solve these problems except by being as responsive as possible to them. And props to the MyEtherWallet people for understanding that they need to explain what this is because they're doubtless feeling the same sort of frenzy over Bitcoin and Blockchain and Ethereum and everything, and recognizing that people are jumping into this without appreciating what this is and that the nature of the anonymity, the nature of the security that this offers also holds responsibility. I mean, here you and I were talking earlier at the top of the show. It's like, oh, where are those bitcoins? Oh, I don't know. You know?

Leo: Let's put them in this wallet on iOS. What could possibly go wrong? Yeah.

Steve: There's no one I can email a complaint to and say, hey, could you please send me my 50 bitcoins? I'd like to have them now because suddenly they're really valuable. Whoops. Nope. So, I mean, I understood that. But today other people are not going to. So we need - that education has to happen, and there's going to be some pain in the process, inevitably, I think.

Leo: Yeah. Of course whenever there's a bubble like this you get a lot of people who are just greedy and not particularly savvy. They get taken advantage of.

Steve: Yes. So a very cool project on GitHub that I know that some of our listeners are going to find interesting is known as - it's called DNS Twist. And it is GitHub.com/elceef, that's the guy, and his project is /dnstwist. And he wrote: "See what sort of trouble users can get in trying to type your domain name. Find similar-looking domains that adversaries can use to attack you." His project can detect typosquatters, phishing attacks, fraud, and corporate espionage. He says: "Useful as an additional source of targeted threat intelligence."

So get a load of this. "The idea," he writes, "is quite straightforward. Dnstwist takes in your domain name as a seed, generates a list of potential phishing domains, and then checks to see if they are registered in the DNS." He says: "Additionally, it can test if the

mail server from MX record can be used to intercept misdirected corporate emails, and it can generate fuzzy hashes of the web pages to see if they are live phishing sites."

So it's just very cool. It was developed under Ubuntu Linux, so that's the primary development platform. So if you're a Linux user, it's easy to run it. macOS users can use Homebrew in order to host it. And it's got a Docker container, so you could use that. But so what this does, and the page shows it running, he has got an animated command screen that you kind of see it going. You give it, for example, Amazon.com, and it generates an incredibly large number of lookalike Amazon.com -esque or -ish domains and then emits DNS queries to all of them, looking to see if anything that looks like Amazon.com was registered, and tells you, if so. And of course you put your own domain in and see if anybody has bothered to register things that look like it. So it's simple, but it's just a very cool idea. He supports Unicode, so he's doing all kinds of Unicode character lookalike attacks. And anyway, just a very cool spoof test. And it's free. So if that idea...

Leo: Installing it now.

Steve: ...jumps in, well, I think it's a cool - yeah, see what happens with TWiT.tv.

Leo: You want me to run it on that? Oh, geez, I know. I don't even want to know. I know. Yeah, let's try it.

Steve: So in the meantime, someone sent me a link to a very cool piece of work. The links are now permanent on GRC's Linkfarm page, so everyone can find it. All you have to remember is that GRC's Linkfarm has it. They're also in the show notes here. It is a beautifully assembled how-to for using the Ubiquiti EdgeRouter X, setting it up for network segmentation, specifically for IoT devices. I mean, I don't know that this guy is a listener to this podcast. But if not, the coincidence is extreme because all of this assumes exactly what we want, that we've talked about here.

The Ubiquiti EdgeRouter X is this cute little $49, incredibly powerful router that our listeners are using, many of our listeners are using. The problem has been that I've never had time, essentially, to do this. I would love to have time, but as everyone knows, SQRL first, SpinRite 6.1 next, and so forth. But that exists now. So it is a - don't remember how many pages, 60-some I think it was, page PDF.

So you can grab it from his GitHub page or get the link from the Linkfarm or from the show notes here for this Episode 641. And it takes you through the entire process of how to create separate segments and even separate SSIDs on separate WiFis so that all your light bulbs and your door locks and your Apple and Amazon and all that IoT stuff can be on a segment of your network in its own address space with no ability to touch your computers or your high-value things, and how you can, you know, each port of the five ports of the Ubiquiti can be its own network.

So thank you for the guy who put it together. And for anyone who's been wondering if they set theirs up right, or maybe got an EdgeRouter X and then just immediately came to a standstill when it's like, ooh, boy, I'm not sure how to do this, well, now there's a perfect guide for answering that question for our listeners. So I'm delighted for that.

There was, last week, an update across the board for Windows. It was in response to a

CVE, a common vulnerability that Microsoft was made aware of; and it was so bad they had to immediately patch it across the board. They said: "A remote code execution vulnerability exists when the Microsoft Malware Protection Engine does not properly scan a specially crafted file, leading to memory corruption. An attacker who successfully exploited this vulnerability could execute arbitrary code in the security context of the LocalSystem account" - which we know is as bad as it gets. I mean, LocalSystem account is root, essentially, in Windows - "and take control of the system. An attacker could then install programs; view, change, delete data; or create new accounts with full rights."

So anyway, this was a very bad vulnerability that was found. And we've talked before about how antimalware systems can themselves, by their very nature, create an expanded attack surface. If they're not implemented correctly, since they're looking at everything coming in from the outside, they have to be perfect. And they are, also by their nature, they're interpreters, and we know how difficult it is to get interpreters exactly right.

So I would argue, and we have been, that once Microsoft has a good enough malware protection facility, it makes more sense to use Microsoft's than a third party's because we have seen third parties having problems discovered in theirs. Microsoft is able to and does respond immediately to push patches out to their systems. So I think they did everything they could. I still feel best using Microsoft's built-in solution for protection. And hopefully it won't be needed very often. But they are keeping it updated as quickly as they can. And they've got a facility for immediately pushing out any changes.

So that's what happened last week, prior to today's big second Tuesday of the month update. And I'm using a Win10 system for my Skype link, and so one of the things I do when it's the second Tuesday of the month is turn it on very early in the morning and immediately go to check for updates and, oh, look, there is something, what do you know. Now we know it's because there's going to be a rollup every month. So I get that done, and I reboot a few times, and I go back and check to see if it thinks there's anything more. So I make sure the system is settled down well before the time that we have to make a recording so that in no way is this going to be updated. But one has to do that these days with Windows 10. And so that's why I was a little surprised to find something last week. It was Microsoft jumping on a problem.

And speaking of jumping on a problem, Apple was also notified of a problem for which there is no current information. 9to5Mac had the reporting on this, and they said a HomeKit vulnerability in the current version of iOS 11.2, so that's not the one that Project Zero found the problem in last week, where we were at 11.2.1, but rather the update later last week to 11.2. So a vulnerability was found in 11.2, which we're all using at the moment, and was demonstrated to the guys at 9to5Mac, which allows unauthorized control of accessories, including smart locks and garage door openers.

They wrote: "Our understanding is Apple has rolled out a server-side fix that now prevents unauthorized access from occurring while limiting some functionality and an update to iOS 11.2 coming next week." And they wrote this five days ago, so last Thursday. So an update coming next week, meaning this week, so we're already expecting once again iOS 11.2 to be replaced, probably by iOS 11.2.1 this week, will fix the problem and restore that full functionality, which was reduced.

So they wrote: "The vulnerability which we won't describe in detail was difficult to reproduce, but allowed unauthorized control of HomeKit-connected accessories, including lights, thermostats, and plugs. The most serious ramification of this vulnerability prior to the fix," they wrote, "is unauthorized remote control of smart locks and connected garage door openers, the former of which was demonstrated to 9to5Mac. The issue was

not with smart home products themselves individually, but instead with the HomeKit framework itself that connects products from various companies." So of course we know, as always, bugs are not good. But in this case the centralized nature of HomeKit's architecture allowed Apple to immediately neuter the undisclosed vulnerability at their end while whey prepare a proper fix, push it out, and then reenable the reduced functionality.

So as a consequence of the architecture that HomeKit uses, Apple again was able to respond quickly, prevent there from being any consequences, and give them a little breathing room to fix this. And it wasn't clear what functionality was reduced. But I imagine anybody who had a vulnerable front door lock and garage door would just as soon have the vulnerability prevented, even with some loss of functionality, until Apple gets it fixed, which apparently is going to happen this week.

And finally - well, not finally, but next - I thought this was interesting. I just ran across Valve announcing that they're ending support for Steam purchases made with bitcoin. And it's like, okay, what's the problem? Well, two things. It turns out that, well, in a post on Steam, Valve wrote: "In the past few months we've seen an increase in the volatility in the value of Bitcoin" - yeah, no kidding - "and a significant increase in the fees to process transactions on the Bitcoin network."

They said, "For example, transaction fees that are charged to the customer by the Bitcoin network have skyrocketed this year, topping out at close to $20 per transaction last week," they said, "compared to roughly $0.20 when we initially enabled Bitcoin," which was sometime in 2016. "Unfortunately," they wrote, "Valve has no control over the amount of the fee. These fees result in unreasonably high costs for purchasing games when paying with Bitcoin. The high transaction fees cause even greater problems when the value of Bitcoin itself drops dramatically."

Valve also explained why surges like that can have an impact on purchases through Steam. They wrote: "When checking out on Steam, a customer will transfer X amount of Bitcoin for the cost of the game, plus Y amount of Bitcoin to cover the transaction fee charged by the network. The value of Bitcoin is only guaranteed for a certain limited period of time, so if the transaction doesn't complete within that time window, and with Bitcoin value fluctuating wildly, the amount of Bitcoin needed to cover the transaction can change. The amount it can change has been increasing recently to a point where it can be significantly different."

So anyway, Valve said that those price discrepancies would normally result in a refund or an additional payment from the customer when more traditional, more stable currencies are involved. But in the case of bitcoin, high transaction fees themselves can make those subsequent resolutions expensive. So they concluded, saying: "At this point, it has become untenable to support Bitcoin as a payment option. We may reevaluate whether Bitcoin makes sense for us and for the Steam community at a later date."

And it's interesting, that made me dig into a little bit like what is this about transaction fees? Like what's happening? And so the way the bitcoin bit chain system works is that the bitcoin block is deliberately limited to a megabyte of data. And a block is completed every 10 minutes. So only a megabyte of transaction data is allowed to be processed every 10 minutes. So this creates a transaction rate scarcity within the entire bitcoin ecosystem, and those wishing to push higher value transactions through the system have ended up creating a bidding war to get their transactions into the next block to be computed.

So what's happened is there's now essentially an auction system in getting transactions

into the blockchain for validation, which has pushed the price up from what used to be just a minuscule fee to now $20. And the other reason is that the transaction fee was in a made-up unit of satoshis and some fraction of a bitcoin. But once upon a time bitcoins weren't very valuable. Now bitcoins have been floating around $20,000 each. So what was once a micropayment has become a mega-payment because the bitcoin price has gone up, but the fraction of satoshis has remained fixed and is fixed by the system, so it's become expensive.

So anyway, I just sort of thought - I thought this was an interesting snapshot into some other unintended consequences of both the high value of the bitcoin and the fact that the bitcoin value is volatile and that it's also become so popular that this limit on block size has created a constraint in the rate at which transactions can be processed, which creates rarity there. And that then further drives up the price, as people bid essentially in an auction in order to get their transaction processed. So, eh. I don't know. I mean, Leo, I guess I know I'm glad to have my, well, somewhat ephemeral bitcoins or the promise of them. But it does seem like the system is becoming a little bit a victim of its own success.

Leo: Well, that's why there are those two forks, you know, partly because the blockchain got so big, but also because of the speed of transactions and the transaction cost.

Steve: Right.

Leo: And the bitcoin board really seemed reluctant to make any changes at all to the algorithm.

Steve: Yeah.

Leo: So, yeah, I wonder; you know? I think you and I should sell our bitcoins quickly.

Steve: Well, it's not clear that they're ever going to go down. They may just stop going up. But, I mean, it's like people are, I guess - now [crosstalk] speculate.

Leo: They can go down, though; right? I mean, it's not - yeah. I mean, it's all speculation. Come on. The Winklevosses are bitcoin billionaires.

Steve: Yeah. Well...

Leo: And they're so bullish that that's all imaginary because they're not going to cash them in.

Steve: Right. And in fact one of the things I had in my miscellany, I'll talk about it now because it's on point, is it's a little disturbing to wonder if those cryptocurrency or the

cryptomalware people that we talked about for years, who were asking for payment in bitcoin, had they kept them...

Leo: Oh my god, yeah.

Steve: Yes, yes.

Leo: Could have been a big difference, yeah.

Steve: It was 400 bucks for a bitcoin for a long time, or 350-something, I remember. So people were like, oh, well, I guess it's worth a bitcoin to get my files back. Well, if those guys kept their ill-gotten funds in bitcoin, well, now they may have a lot of, I mean, they may be nefarious millionaires.

Leo: For me to exchange my bitcoin for dollars would require that somebody buy them; right?

Steve: Yeah, I mean, there are exchanges where this is happening.

Leo: It's an exchange. And the price is determined by what people are willing to pay.

Steve: People are buying, yeah, yeah.

Leo: So it could go to zero if nobody wants them.

Steve: Yeah, you should grab - Bitcoin Ticker is my favorite iOS app.

Leo: I don't want to follow it that closely.

Steve: It's just interesting.

Leo: Wow.

Steve: So the good news, the good news for people who want encrypted mail. Yesterday ProtonMail announced Bridge. So I finally have truly good news about email encryption. In their blog posting they said: "Today we are officially launching ProtonMail Bridge, which brings easy-to-use email encryption to desktop email clients." And the reason I'm sharing this with our listeners is I'm bullish about this. These guys got it right. They wrote: "Ever since the day that we first got the idea to create ProtonMail, one of the most enduring challenges has been how to do email security right while simultaneously

making encrypted email easy enough to use for normal people. Since our early days working from the CERN cafeteria, we have been working tirelessly to address this specific problem.

"In the years since, we have made many great strides towards creating usable encrypted email, first with ProtonMail's webmail interface and then with our award-winning iOS and Android secure email apps. However, one of our goals has always been to bring easy-to-use encrypted email to the desktop. The problem is formidable. Desktop systems encompass multiple operating systems with dozens of popular email clients with their own adherents, and virtually none of them natively speak PGP, the email encryption standard upon which ProtonMail is built. Around two years ago, we created a small task force to tackle this challenge. Today, we are finally ready to present ProtonMail Bridge."

I'll skip a lot of the stuff, and jumping down they said: "Furthermore, after the technical documentation of the ProtonMail Bridge code is done, we will be releasing the source code of the Bridge, so that you can even compile it yourself instead of getting the binaries from us, so there is even less need to trust us. This is an important step in our work to eliminate ProtonMail itself as a threat vector. Currently, the officially supported email clients are Thunderbird, Apple Mail, and Outlook, on both Windows and macOS. Linux, they write, "is coming in spring of 2018." So, what, a few months from now. "However, in theory, any IMAP email client can work with the Bridge; and, in our beta testing, many were shown to work. If you are a ProtonMail user, you can immediately get started."

Okay. What is it? Well, it is simple and clever. It is a local mail server. So you run this app on your Windows or your Mac machine. It opens an email port on your local system. So rather than aiming your email client at some remote SMTP server, for example ProtonMail or Cox.net or whatever, you point your email client, whatever email client you're using - they're officially saying Thunderbird, Apple Mail and Outlook, but it could be anything because what they've done is they created a local IMAP and SMTP server running on your local machine, that is, on localhost.

So your non-secured, non-encrypted email client talks to the local server running on your desk. And then that server, which is currently what they haven't yet published the source for, but they're going to, it uses the Proton Mail API to connect securely to the ProtonMail mothership and exchange all your mail over PGP securely, encrypted and with authentication and all the benefits of PGP. But the point is, from before it leaves your desktop, before it leaves your system, it is wrapped in PGP protection and on its way to ProtonMail for store and forward to whoever you want to have secure communications with.

So this is beautiful because it solves the problem of complex configuration, of email clients not natively supporting or easily supporting PGP. The problem is that people keep - I see this problem. I know that there's an interest among our listeners because I'm constantly getting email from people talking about this or that email system that is secure. And it's like, yeah, except that the whole problem hasn't been solved. These guys have a nice solution. By setting up a local server on your machine, you're talking essentially within the system on your own network stack. Your client connects to the ProtonMail server running locally, establishes a remote connection, and you're secure all the way through.

So yay to the guys at ProtonMail. And I know that we've got a lot of users of ProtonMail who are arranging to use it through whatever means predates this. I wanted to make sure everyone knew, and people who are interested in this who could become ProtonMail users, that you can now use your native system email client and get all the benefits.

Leo: I'll stick with GPG. It makes absolutely no sense to me. I don't understand it at all. GPG works fine, and it doesn't need to set up a server on your system, and you don't need to go through some third party. You just use GPG.

Steve: Okay. Cool.

Leo: Yeah, I don't get it. I frankly don't get it at all. But okay. Why do we need to do this? I don't even understand why we need to do this. Why are we setting up a mail server and then routing it through ProtonMail? I guess if you're a Proton Mail user, maybe.

Steve: Right. And so it allows you to use any client that you want to.

Leo: But I can use, well, I can use all the clients that they currently support with GPG, so I'm not sure what I gain. Anyway, I'm sure we'll find out.

Steve: Well, actually, I already talked about the annoying idea that ransomware miscreants could have seen a huge value inflation in their ill-gotten gains. I did want to mention that I had a breath-holding moment as I moved my one most critical domain, but then the second most critical one, from Network Solutions, where it has been since 1991. December 18 of 1991 I registered GRC.com, and Network Solutions has been its registrar. It is now at Hover because, of course, December 17, its annual expiration was coming up. And I said, okay, I refuse to remain at Network Solutions any longer. I am now at Hover.

I just wanted to say that it went perfectly. No one even knew it happened. No outage, no glitch. I was a little nervous about it because it's of course the most important domain I have. And it went perfectly. So now I've got all of the services that Network Solutions either didn't offer or wanted to charge extra for, which is annoying, like privacy and lock and so forth, all there, all turned on, and GRC.com is happily at Hover, probably happily ever after.

Leo: I just have to mention, even though I know it has nothing to do with [crosstalk]...

Steve: Oh, yes.

Leo: ...that Hover is a sponsor of some of our shows.

Steve: Yes.

Leo: But that's not why Steve's choosing it. And this is [crosstalk].

Steve: No, in fact I...

Leo: He's paying for this.

Steve: In fact, didn't my raving about Hover precede them being a sponsor?

Leo: No, actually. They've been a sponsor for many years.

Steve: Oh, okay. Well...

Leo: I think, if I'm perfectly fair, there might have been a little bit of a feedback loop because you tweeted what's the best one to use, and I imagine many of your respondents were listeners who already were [crosstalk].

Steve: Ah, well, I certainly do agree because I've been raving about it ever since, yes, yes.

Leo: It is good, yes, right, I agree, yeah.

Steve: So in another fun example of success with SpinRite, Al in Portugal sent me a note saying "SpinRite Sound Recovery," dated the 10th of December, so a couple days ago. He said: "Hi, Steve. Am still running Win8 here. Today the sound stopped playing back." He said: "I checked the sound troubleshooting area in the Control Panel, but it said it found nothing wrong with the sound. Brilliant.

"So," he says, "I dug around the sound dialog from Control Panel to see if I could refresh the playback by disabling and reenabling things, but nothing worked." He said, "In fact, when I clicked the playback tab, sometimes it would bring up the small spinning blue circle that never stopped spinning." He said, "I had to terminate the rundll32.exe to close the sound dialog.

"This made me think that perhaps there was something wrong with Windows being unable to read some files somewhere. So I thought, why don't I give SpinRite a go to see if it fixed things. I first tried to reboot twice to see if that might help, but it did not."

He said: "So then I started running SpinRite on Level 2 and let it run for just the first 1.5% of the 500MB disk drive. I only did such a small amount because I heard that sometimes people had only run SpinRite for a little while, and it had fixed various problems. I think it fixes some stuff in the filesystem. Anyway," he says, "I fired up Windows after that, and the sound was working again. So I think it must have been SpinRite finding and fixing the problem. Thanks very much, Steve, for SpinRite, and for the Security Now! podcast, too. Al in Portugal."

And of course we have talked a bit from time to time, in fact recently, about how sometimes, even if SpinRite runs across something it can't apparently fix, when people reboot their system, sure enough, it got the job done anyway. And so in this case Al just deliberately ran it for a little while, 1.5%, and that was enough to get the job done. And

of course we know that, when you set up Windows, when you install Windows from scratch on a blank drive, it's probably going to be at the front of the drive. That's where the files will go. Windows will sort of probably start allocating itself from the beginning.

And so that's one of the reasons that Windows-based problems can get themselves resolved most easily. And in fact it's also why that tends to be where the damage occurs is that the hard drive head will generally be hovering around those areas most read. And so if something happens, if you bounce the head off the surface or you trip over the power cord when it's in the middle of writing, it may glitch the data there. So that generally is where the problems tend to accrue, toward the beginning of the drive. And so it actually is where SpinRite generally finds the most problems to fix. So running it just for a while can often solve the problem.

So a couple of closing the loops with our listeners. Jim Clary said, or asked: "Any reason for 'I'm not a robot' check on a logon page?" He said: " I understand the usefulness for account generation, but not login." And he said: "Humana uses it, making it more difficult to use a password manager." And Jim, I have to say I think you're right. I can't really see why, I mean, I guess a bot that was doing guessing would be thwarted if it were trying to log on.

Leo: There are better ways to do that, like rate limiting.

Steve: Yeah, yeah, exactly. And of course there is a little bit of a problem, too, because remember from a privacy standpoint the presence of that means that the supplier of the "I'm Not a Robot" checkbox is seeing that you are logging in every time you are.

Leo: I see CAPTCHAs more and more often. It really bothers me. I signed Abby up for the Affordable Care Act yesterday because the deadline is December 15th, don't forget. If you're going to use ObamaCare, got to do it now. And it had a CAPTCHA at the end. And I think maybe healthcare providers, maybe it was - might have been through the healthcare provider. I can't remember.

Steve: Might be mandated, you mean?

Leo: Well, yeah, but I wonder why. I don't understand the security benefit to it.

Steve: Interesting. Yeah, no...

Leo: You make an excellent point. It's just routing it right through Google.

Steve: Yeah, exactly.

Leo: I mean, what robots are signing up for healthcare? No, I'm just kidding. I'm kidding. I'm kidding, robots. Not talking to you.

Steve: So all the bots could use some healthcare.

Leo: Maybe.

Steve: We often discuss on this podcast. Peter Griffiths asked if a branded router...

Leo: Not his real name, by the way, I think. He's a character on a TV show.

Steve: Oh, Peter Griffiths? Okay.

Leo: Maybe it is his real name. I don't know. I shouldn't make fun of him.

Steve: He said: "Is a branded router always best?" He said: "Visiting the folks and, given recent router security awareness, wanted to update their firmware. Linksys router, no firmware update available newer than 2013. He says: "Time to buy a new one?" And, okay. So that's a good question. I guess the question would be whether that particular firmware had known problems. But you would expect, then, if it did, that there would still be newer firmware available unless Linksys was just taking down firmware for older hardware. But I don't really see a reason to do that, and four years is not that long.

So what you really want is you want a router from a company with a reputation for really maintaining the currency of their firmware. And frankly, something like, I mean, we have seen Ubiquiti having some problems, but they're immediately putting up firmware patches and responding pretty quickly to them. And the router's not very expensive, and it's very capable. So maybe. I of course like the idea of running something like pfSense on a small platform and having essentially a non-branded router which runs and is reliable.

So I don't know. I think it's kind of six of one, half a dozen of the other. But certainly sticking with a company that is going to be maintaining the firmware moving forward makes a lot of sense, I think. And also it's worth noting you can always put a router inside of a router. It's sort of the - that was the whole "three dumb router" concept. But you can certainly just do two. And that prevents the inside network from exploitation from the outside. It doesn't prevent malware from running on your router. But it does at least prevent them from looking into and getting access to your devices directly. So that's maybe something to consider, as well.

Leo: And it was Peter Griffin, so not Griffith, that's in "Family Guy." So I apologize, yes.

Steve: Ah. So Patrick Hogan said: "Steve, do you think it's annoying that some ISPs don't enable custom DNS servers to be configured in their customers' home routers?" And he said: "@SkyUK are dragging their feet with this." And, yeah, that would be annoying. I don't use a router that my ISP has control over in the states. And in fact it was in that context, of course, there are two things you could do. In most of the DHCP config of our systems, you're able to - the Obtain IP Address Automatically also has a subsidiary of get your DNS servers also from DHCP. In Windows, at least, you can turn

that off. And I've seen that option over on Macs, and I'm sure it's there on Linux.

So one option is just to not have your systems where you have control of it. Of course you don't have control with light bulbs and so forth. But where you do have a desktop UI, you could override the system's DNS so that it's not getting it from the router's DHCP. And that's another place where you could put a router inside of your router, that is, a router inside your network that you do have configuration control over, even if you may not have configuration control over the one your ISP provided, and then set it up to provide DNS to everybody within your network. So that's also a possibility. So with a little more configuration responsibility, you can probably get around it. But, yeah, it would be annoying if you had to go through all that, rather than just having control over the router that your ISP has provided.

Jonathan Harris said: "I believe I've found a bug in a brand of networking equipment my company uses. I have tried contacting them regarding the issue, but receive no response. How can I verify this problem and how should I proceed?" Well, okay. Without any more details, it's impossible to provide any advice. But I will remind our listeners of security.txt as the new file which is probably too new to actually exist. But you know how robots.txt has been around since forever, which is a file that well-behaving robots look at in the root of any domain in order to get essentially marching orders to tell them to stay away from certain areas in the domain. We now have security.txt. And there is a site, securitytxt.org, where they document the format of the file and are working to publish this.

So one thing you can do, Jonathan, the only thing I can think to do, is to go to the site for that networking equipment and put security.txt, I'm sorry. Go to the site and go to /security.txt as the URL and see if you get a file. If so, then that company - and certainly if they're a networking equipment company, the chances are hopefully more likely that they would have a file that they would offer. There they may tell you who to contact in order to support security problems.

Leo: I asked Patrick Delahanty to do it on our server. I don't know if he has yet. But, yeah.

Steve: Good, good. That's a brilliant...

Leo: Everybody should do that, just to reassure security researchers it's safe to report flaws.

Steve: Yes, exactly. It's safe, and we will not...

Leo: [Crosstalk], yeah.

Steve: Exactly. We will not hurt you. And here's where you send your report to, yes.

Belwig asks: "Long-time Security Now! listener. I don't recall you ever going over the right and wrong ways to handle password resets. Are the token-based links in emails secure for resets?" Then he said: "Waiting patiently for SQRL." And so, okay, first of all, we all know the wrong way is for you to press a button saying I forgot my password, and

then they send it to you. [Buzzer sound] Wrong. You also don't want them to send you a temporary password. That's wrong, too, the idea being that anybody who is able to, I mean, the big problem with all of these email-based resets is that email is not as secure as you would like it to be. But unfortunately, email is what we've got. And so nobody likes the idea of using email to reset passwords.

But certainly the only thing we've got, if email is your lowest common denominator, I mean, and if you've got two-factor authentication enabled, then you want to be able to tell the person, okay, you need to provide your second-factor code. But otherwise, clicking on a link that you receive in email, that then takes you to the site, that then requests any other information that you may be able to provide like a second factor, if you've got that registered, is the best that we're able to do with email, and arguably sending you your password means that, among other things, they didn't hash it, and so that they're able to provide you your password back in plaintext, which should terrify anybody.

But, yeah. Unfortunately, unless there is some other means of authenticating you other than email, the lowest common denominator is a link that you click on that takes you to the site, where you're then able to provide a new password. And that's the best we've got at the moment. And, yes, I'm also waiting for SQRL.

Yosef Berger asks: "When storing passwords by hashing and salting, why is it considered bad practice to use the user's name as the salt? If we want to make it so different users with the same password don't end up with the same hashed password, why is a unique username not as good as" - now, he said an RSG, and I guess he means just a randomly generated token. And I agree, Yosef. A hash does not need to be secret. It's best if it's unique. But it's really good enough if it's probably unique.

So I don't see a downside for using the user's name as the hash. You don't want one hash globally because then that allows, I mean, you want hashing at all so that you can't use pre-computed hash tables. A global hash would allow a computation of precomputed hashes to be made once and then applied to everyone. So a per-user hash means that the hashing needs to be done for every single user. But a username as the hash salt is going to be almost user-unique. Maybe you could have two people with the same name; but, okay, that really doesn't provide someone who wants to brute-force the hash much benefit.

So I agree. I mean, it's probably better just to use a randomly generated token. And why is that hard? It's as easy, essentially, as using the person's name. But if there were some reason, if someone wanted to make an argument for it, it's like, yeah, I don't really see that being much of a problem.

And, finally, Javier - wow, Javier, I don't know how to pronounce your last name, Matusevich I guess - said: "You've said it's incredible WINE works at all," referring to the fact that the guys under Linux have managed to get essentially a complete Windows environment running under Linux. And then he said: "Running your DNS Benchmark on my Mac. Flawlessly works." And it's no coincidence because I want my Windows stuff to have as much reach as possible. And so when I was nearing the end of the work on the DNS Benchmark, I made sure that it would run under WINE. And of course WINE is also available on the Mac, so that allows the DNS Benchmark to run on the Mac and on Linux machines.

And I will assure people that a lot of the time and attention has been going into making sure that SQRL also runs under WINE, and it does. So that will allow, until there are native clients written for Linux and for macOS, will allow GRC's SQRL solution to be run

across all those platforms by having it hosted on WINE. So we're taking the time to make that happen; and we've got a bunch of WINE users, Linux users, over in the SQRL forum who are providing valuable feedback on the clients' operability under WINE.

Okay. And finally, what we promised: iOS Security versus Convenience. What is it that Apple reportedly, well, definitely did with iOS 11? Now, okay. This is coming from the ElcomSoft guys who blogged about this last week, calling it the "Horror Story of the Rise and Fall of iOS Security." And I've got to take that with a little bit of a grain of salt. First of all, these are the guys who do phone hacking stuff. So they're actually not unhappy that Apple made the change that they made because it gives their forensics tools much greater reach.

And so as I'm reading this, and they're complaining about it, it's like, what? Really? You're happy. I mean, they published something called the iOS Forensic Toolkit. They've got the ElcomSoft Phone Breaker with the Keychain Explorer and the ElcomSoft Phone Viewer, all of which are forensics tools which have just become more valuable as a consequence of a decision that Apple made to back off on the way backups are protected. As I said, I read through this thing, this kind of poorly written blog posting that rambles along and gives lots of examples, but never really gets to the point, which was why this was sort of difficult to nail down.

But so to sort of drive the point home, what they said, what this guy wrote in his blog was: "With the release of iOS 11, Apple developers made too many assumptions" - okay, I don't really agree with that - "breaking the fragile security-convenience balance and shifting it heavily onto convenience side."

They write: "Once an intruder gains access to the user's iPhone" - so they have to have physical possession of the iPhone - "and knows or recovers the passcode," so those two things, "there is no single extra layer of protection left. Everything" - and, they write, they mean everything - "is now completely exposed. Local backups, the keychain, iCloud lock, Apple account password, cloud backups and photos, passwords from the iCloud Keychain, call logs, location data, browsing history, browser tabs and even the user's original Apple ID password are quickly exposed. The intruder gains control over the user's other Apple devices registered on the same Apple account, having the ability to remotely erase or lock those devices

"Finally, regaining control over the hijacked account is made difficult as even the trusted phone number can be replaced. Why," he writes, "Apple decided to get rid of the system that used to deliver a seemingly perfect balance between security and convenience is beyond us. Once someone has your iPhone and your passcode, you are no longer in control of your device or your Apple account."

So, finally, what can you do to protect yourself? Since the passcode is now the one and only safeguard left, make sure you use at least six digits, or of course switch to the full alphanumeric one, and use one that's robust. Four-digit PINs are no longer secure. Other than that, we'll just wait and see if Apple can fix it.

So that's ultimately the takeaway is the passcode is your last line of defense. So then in trying to understand exactly what it was that happened, it boils down to Apple's change, which they did make in iOS 11, of allowing password recovery for backups because backups are where everything lives. And until iOS 11 - and we've talked about this in the past. If you lose your backup password, your only recourse has been to reset your device and start over. So they write in this blog posting for backup passwords in iOS 8, 9, and 10, in these versions of iOS one could protect their backups by specifying a backup password in iTunes. One would only need to do it once.

Once a password was set, all future backups made on that computer and any other computer with no exceptions would be protected with that password. The password would become the property of the iDevice and not the PC or the copy of iTunes that was used to set the password. You could connect your phone to a different computer and make a local backup with a freshly installed copy of iTunes, and that backup would still be protected with the passwords you set long ago. Any attempt to change or remove that password must pass through iOS, which would require the provision of the old password first. Forgot the original password? There's no going back. You're stuck with what you have unless you're willing to factory reset the device and lose all data in the process.

He writes: "If you ask me, this was a perfect and carefully thought-through solution. Want to protect your data against an attacker? Set a long and complex backup password and don't store it anywhere. Forgot that password? You can still make a cloud backup and restore your phone from that backup. Even your passwords in the keychain would be restored if you rolled out the cloud backup onto the same device you made the backup from, or used iCloud Keychain if that was to be a different device."

And then he finally concludes with this, saying: "A perfect system? Apparently, it was not to everyone's liking. The users whined. The police complained. The FBI complained. And Apple gave up." So under iOS 11, he writes: "Stripping Backup Passwords. In iOS 11 you can still specify a backup password in iTunes, and you still cannot change or reset it through iTunes if you don't know the original password. However, this means very little as you can now easily remove that password from iOS settings." He quotes Apple. This is what Apple has to say in its Knowledge Base.

Quoting Apple: "You can't restore an encrypted backup without its password. With iOS 11 or later, you can make a new encrypted backup of your device by resetting the password. Here's what to do." And Apple says: "One, on your iOS device, go to Settings > General > Reset. Two, tap Reset All Settings and enter your iOS passcode. Three, follow the steps to reset your settings. This won't affect your user data or passwords, but will reset settings like display brightness, home screen layout, and wallpaper. It also removes your encrypted backup password. Four, connect your device to iTunes again and create a new encrypted backup."

And then Apple concludes: "You won't be able to use previous encrypted backups, but you can back up your current data using iTunes and setting a new backup password. If you have a device with iOS 10 or earlier, you cannot reset the password," writes Apple.

So ElcomSoft says: "That's it? That's it. You have just removed the backup password. You can now make a new backup or, rather, extract information from the device." He says: "Don't rush, and do make sure to specify a temporary password - '123' always works - before you make that backup. A password-protected backup will allow you to decrypt the user's passwords, credit card data, health data, and other things that would be otherwise inaccessible. So set a temporary password, make that backup, decrypt it with" - and then here of course they're selling or promoting their ElcomSoft Phone Breaker or just use Keychain Explorer, a tool in the ElcomSoft Phone Breaker, to access that user's passwords, authentication tokens, credit card numbers, and other interesting things. Oh, and their pictures, too.

So essentially it must have been that there was some pushback, I mean, that's the only rationale I can see is Apple decided with iOS 11 to allow people to sacrifice their previous backups when they've forgotten their password, but to reset the password which was not resettable before and make a new backup. So what that means is that if someone does acquire your phone and has your passcode and can get your passcode, because that is required in order to perform this reset, then you are allowed to reset your backup

password for subsequent backups.

So then the person doing the forensics work can choose a trivial password, produce a backup and then, knowing that password, get decrypted access to the backup using any of the iPhone forensics tools that are available and essentially dump the contents of your phone. So that's what that was about. That was the argument that this guy was making about the tradeoff that Apple, they argue, made. And it does appear like Apple must have deliberately consciously decided they want to allow people to reset their backup password. You cannot do it prior to 11, and you can now.

Leo: Well, there you have it, folks.

Steve: Yup.

Leo: I mean, so to summarize, is it fair to say that Apple has backed off a little bit on its once extreme security...

Steve: I think it has.

Leo: ...in favor of convenience.

Steve: This allows you to get to your keychain, which, I mean, and a lot of other things that you could not otherwise get to.

Leo: So as poorly - as confusing, and I tried to read it, too, as ElcomSoft's article was, it does make a valid point.

Steve: I think it does, yes.

Leo: That was my sense, too, but I [crosstalk].

Steve: Boy, it was a journey to get to it, I know.

Leo: Yeah. Good. All right. Mr. G., you've come to the end of another fabulous episode of Security Now!. We do the show every Tuesday, right after MacBreak Weekly. That usually ends up being around 1:30 p.m. Pacific, 4:30 Eastern, 21:30 UTC.

Now, let's see. This here is December 12th. We'll be back on the 19th. And then we have a couple weeks off; right? I'm trying to figure out what we're doing here. On the 19th, and then the next one will be the 26th, that's going to be our special episode.

Steve: Right.

Leo: Kind of a how - can I say what it is?

Steve: Well...

Leo: We'll surprise you.

Steve: Yeah.

Leo: Although I think there'll be a lot of interest in it, let's just put it that way.

Steve: I do think it's - yes. Yes, yes, yes.

Leo: And then we will be back doing the show for the new year. So we have one more episode in 2017.

Steve: Yay.

Leo: Before our holiday break. But you're invited to come by and say hi. Just tune in TWiT.tv/live about 1:30 p.m. Pacific, or go to YouTube or Ustream or Twitch. We're on all of them. You get your choice at YouTube, rather at TWiT.tv/live. If you do go in live, join us in the chatroom. Nice bunch of people, always kind of have a good fun attitude about the whole thing at irc.twit.tv. See, they're all saying thank you. Some people are suggesting maybe a bitcoin mining option in SQRL to raise the attention there, kind of a marketing thing. You can also get it on demand.

Now, Steve has a great version of it. He has not only the audio, 64Kb audio, at his website, GRC.com, but he also has the transcripts of it. So if you like to read along while you listen, or you just want - it's really most useful for searching for concepts and ideas: GRC.com. While you're there, pick up a copy of SpinRite. That's Steve's bread and butter, the world's best hard drive recovery and maintenance utility. You can also get lots of other free stuff. It's like the Old Farmer's Almanac of Tech. It's just chockful of fun and interesting stuff: GRC.com.

We have audio and video, believe it or not, for the show at our website, TWiT.tv/sn. And possibly the easiest thing to do would be get your favorite podcast client on whatever device you carry around in your pocket and subscribe so you don't miss an episode. You just get it automatically, right after we finish on Tuesdays. I think that's everything, Steve. Thank you so much.

Steve: My friend, my pleasure. See you next week for the last episode, the last new episode of the year. And I think our listeners will enjoy the holiday special. And then we're back in 2018 to do some more.

Leo: The holiday special: Twas the night before SQRL.

Steve: Ooh, yeah.

Leo: That's a good idea, though, huh?

Steve: Thanks, buddy. Bye.

Leo: Bye.