Listener Feedback #214

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-511.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-511-lq.mp3

SHOW TEASE: It's time for Security Now!, the day of Patch Tuesday, so we'll talk a little bit about Microsoft's updates. We also have a bunch of questions and answers. We'll talk about keyless entry systems, a couple of ways to secure them. It's a great Security Now! ahead, next on TWiT.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 511, recorded Tuesday, June 9th, 2015: Your questions, Steve's answers, #214.

It's time for Security Now!. Steve Gibson is here, our Explainer in Chief, the inventor of the Apple II light pen. Something we don't talk a lot about.

Steve Gibson: The LPS II, the Light Pen System for Apple II, yup.

Leo: He's also, of course, the guy behind SpinRite, the world's best hard drive maintenance utility. He's here every week to talk security and really explain so much stuff. And it's great because there's this trickledown phenomenon that happens, Steve, because you explain this to me; and then I go on the radio show, and I try to put it in human speech.

Steve: Perfect.

Leo: In English. And I think, for instance, when we were talking about the keyless entry systems on cars, you did such a great job explaining that. And you talked about your baguette, your little RF-impenetrable bag. I then actually talked about -

yeah, there you go. And I talked about the same thing on the radio show because I think this is something that's really of mass importance, general interest. And I got a lot of emails and eventually got a call from somebody who said, "Where do I get this baguette of which you speak?" And I had to explain, well, that's a generic term. Although it turns out there is a company called Baggallini that makes RFID protection bags.

Steve: Oh, my lord. And it turns out PU, I was calling it the "PU Leather," and that's an acronym for some type of...

Leo: Oh, how funny.

Steve: ...polyethylene something or other.

Leo: Oh, how funny.

Steve: Or polyurethane or something.

Leo: Yeah, PU Leather.

Steve: That's what "PU Leather" meant. It wasn't the brand.

Leo: Well, in any event, what are we doing today? What's the show about?

Steve: So we're going to do a Q&A. We skipped one week before last for Logjam because I felt that we needed to give that some immediate attention because everyone was interested in the details of how another problem with the TLS protocol occurred. And then last week, where there was so much stuff to talk about at the top of the show, we ended up running out of time and didn't even get to our normal batch of questions. So I thought, let's do another Q&A. I have a number of neat topics queued up for the future, and we'll do one of those next week. But today I thought let's do another Q&A. We don't have too much to talk about this week, but some interesting stuff. And I have another mind-blowing demo of SQRL to show.

Leo: Oh. You blew my mind last week.

Steve: I'm going to do it again.

Leo: Holy cow.

Steve: Yup.

Leo: Great. That's exciting. And we're going to talk about "Mr. Robot."

Steve: Yup, we have some media to discuss, a little update. And, yeah, so lots of good stuff. So while I was putting the notes together there was no news from Microsoft on Patch Tuesday. So I assemble them, create PDFs, post them everywhere, note on Twitter that they're posted. When I settled down just pre-podcast, I refreshed the page, and there they were. Also my Windows 7 machine, which is what I use for Skype, it knows about them. There's nothing super notable. There were two of the packages that were marked critical, one for IE and the other one for Windows Media Player. The one for IE is a little disturbing because one of the things it fixes, they say, is the ability for a malicious website to obtain your browser history. And that sort of seems like a bad thing for a website you visit to be able to do. Now, no one uses IE anymore, of course, so it's not such a big deal. But IE's components are deeply wired into the operating system and used for all kinds of purposes.

So the standard second Tuesday of the month. Update as soon as you can. And it will require a reboot of your system because there's a bunch of kernel driver problems which are privilege elevation exploits and stuff. So Microsoft's doing their standard, this is like sort of a medium one. It's not super tiny, and it's not the end of the world. Nothing mission critical. But as you'd expect, worth keeping your systems patched.

One thing I noted, I think it was late last week, that was just sort of a little bit of good news. And that is that there was an appropriations bill in the House that had an amendment proposed for it. And the amendment would block the funding for the National Institute of Science and Technology, NIST, you know, N-I-S-T...

Leo: I thought of you when I saw this, yeah.

Steve: Yes.

Leo: This is Sensenbrenner's; right? I think it was Jim Sensenbrenner that proposed - anyway, yes, yeah.

Steve: Good. And so what I loved about it was - and so what this would do is block the funding available for NIST to work with the NSA and the CIA on undermining or backdooring our encryption. And of course that's one of the ways Congress has of working its will is it controls the purse strings, since it's in charge of the flow of money. But the point was that the vote was 383 in favor, with only 43 against.

Leo: Wow.

Steve: So it was an overwhelming support for the idea of not having this.

Leo: This comes back to where the RSA, or the NSA influenced the crypto algorithms that the National Institute for Standards put out.

Steve: Right. That was the dual elliptic curve deterministic bit generator, which it ended up, as you mentioned, in RSA's package and was used by default, even though it was like the worst one available. I mean, it was just inexplicable that this happened. And as listeners to the podcast will remember, RSA received payment for including this broken random number generator that people have serious reason to believe was a means of essentially compromising encryption because we know, we talk about it all the time, how crucial high-quality randomness is for encryption. The keys are typically - the actual keys we use are randomly generated, and then that key is encrypted. And so our password or whatever, the secret that we agree upon, that's the decryption of this randomly arrived at secret. Well, if it's not randomly arrived at, if there's a known bias, that is, some people know the way it's not random, that allows them a huge leg up.

And so all the people deep into this are, without positive evidence, they're as sure as they could be that there was this kind of influence. But lord knows the House of Representatives has no idea about what I just said. They just don't like the idea. And they don't like it in 383 to 43. So for me, what was heartening was this wasn't a close vote. This was - wow. So, and in exploring this a little bit further, some comments were made, it's like, well, yeah, this is hopeful. But we'll have to see because in the past there has been legislation which, when it finally got down to the point where the final agreement was going to be made, pressure from the intelligence agencies removed some of this language. So it's not a done deal. But I was heartened by just the sentiment that the bias in this vote demonstrated.

Leo: And let me give credit where credit's due. It was not Jim Sensenbrenner, it was - it's called the Massie Amendment. It's Congressman Thomas Massie; Zoe Lofgren, who is our Democrat from California; and Ted Poe of Texas.

Steve: Nice. Yeah, I would have recognized Sensenbrenner. I didn't remember his name in the coverage.

Leo: Yeah, no, you're right, yeah. So very good. They passed the Massie Amendment.

Steve: Nice.

Leo: Nice.

Steve: So we have to talk about the 4.1 million record breach that the U.S. government suffered after last Tuesday's podcast. Not that that had anything to do with it, of course. OPM, the Office of Personnel Management, which is the U.S. government's personnel office, which it was revealed hackers believed to be operating from China - and again, attribution to these sorts of things is always touchy. And of course the Chinese government has adamantly, completely denied any relationship to this at all. And I'm sure everyone has already heard about this because it made lots of headlines. And then it's like, okay, you know, another breach in the government. This, of course, at 4.1 million records, is the worst in history.

Now, in digging around, I ran across a quote that just - I had to look up who this person was that said this. Her name is Donna Seymour. She's the CIO, which as we know is the

Chief Information Officer, for the OPM, the organization that was breached. And she explained that encryption - oh, I forgot, I just stepped on the lead. This was all in plaintext. None of this was encrypted. The 4.1 million records were not encrypted. And so when asked, wait a minute, what, she said, quote, "Encryption and data obfuscating techniques are new capabilities that we're building into our databases."

Leo: That we just discovered.

Steve: It's too new. You know? And how old is your PGP key, Leo? It's, like, two decades old; right?

Leo: Well, yeah.

Steve: Yeah. But, you know, we need to move slowly and carefully here in the government. And encryption, you know, we're still trying to kind of figure that out. And then I just think, yes, and these are the people that want us to turn over a spare set of keys to all of the encryption that we use every day.

Leo: Boy.

Steve: They haven't figured out how to, like, plug the cords together yet. But they're not happy that we're way ahead of them, and we know how to do encryption out here in the private sector.

Leo: You've just nailed it. You know what? That's what it is. They're scared because everybody else understands this stuff, and they don't.

Steve: Yeah.

Leo: That's exactly it.

Steve: Leo, it's too new.

Leo: It's too new.

Steve: It's brand new. It's still wet.

Leo: We're still studying the idea.

Steve: Yeah. The toner is, you know...

Leo: Oh, no. No toner. It's a dot matrix. I promise you.

Steve: Wow, yeah. So NBC News had some coverage of this. I mean, everybody covered it. But I saw this, I caught one little piece that had some fun quotes from people, from private sector contractors who work with people like dear Donna on her security at OPM. A guy named Richard Blech is the CEO of a cybersecurity firm, Secure Channels, and they work with many federal agencies. He said: "It's not even the money as much as the process involved. Everything gets caught in government glue," is the way he phrased it. And he said...

Leo: That's a good name.

Steve: Government glue. "I've worked with these guys, and you have to go through layers and layers of groups and committees to get anything done." He said: "It practically takes an act of Congress to change the computer system." And asked about them not being encrypted, he said he was mystified that the data in these federal breaches are not encrypted. And then, independently, he said, sort of editorializing, he said: "Government networks," the way he characterized them, "are an exponentially growing sprawling mess, and they are nearly impossible to protect. The Einstein intrusion detection system senses network behavior events. But if you don't know what's normal, how can you ever detect what's abnormal?" And so...

Leo: There you go.

Steve: Again, we get this, you know, it shouldn't surprise anybody. Have you been to the DMV? So you know, this is the way the government is. And so there was some rush a decade ago to computerize. Oh, we have to go online. The Internet seems to be real, like it's not a fad after all. And so all this happened before privacy became an issue. And as you probably know, Leo, if you've followed any of the stories, this 4.1 million records goes back decades. This is not, like, today's records. This is old stuff that was moved online, never encrypted.

And now the system is just - it's essentially too big to manage. They can't, they have a difficult time detecting intrusion because it's such a sprawling mess that, as this guy said, has grown exponentially, that they're really not sure where the pieces are and what information is flowing over what wires. It's just, you know, it grew with no regard for security. And in fact I read elsewhere someone saying that what often happened is security was then an afterthought, and it was almost too late to fix it. We, of course, are all familiar with the phrase "too big to fail." This is too big to secure. It's just like, eh, no. We just hope nothing bad happens.

Leo: Security afterthought.

Steve: So our discussion of hard disk drive rootkits has generated a huge amount of interest, just because it's so creepy, the idea that the hard drive firmware could be infected, which puts it at a level out of reach of any tools we have. So I found a 20-year-old hacker. And as I'm reading this, on his page he links to a 46-page presentation

showing in detail what he had to go through. And I found myself thinking, you know, if you aimed yourself at a creative pursuit, I mean, he's having a ball doing this. But imagine what he could do...

Leo: Just imagine.

Steve: ...if he had guidance. If someone said, hey, here's something, like here's SQRL. How about, you know, bringing it up on Raspberry Pi? We don't have anyone doing Raspberry Pi yet, for example. Instead, what he did was he said, you know, I'm going to do this. I'm going to figure this out.

So paraphrasing a little bit from what he posted, he said: "Since I got into firmware hacking, I've been working on a little project behind the scenes, a hard disk firmware-based rootkit which allows malware to survive an operating system re-install or full-disk format. Unfortunately, I can't post a proof of concept for many reasons. People have been contacting me just to tell me not to post it. So instead I've written a presentation overviewing and explaining the rootkit, which I've dubbed MT," because his site is MalwareTech, it's MalwareTech.com. So MT-SBK. SBK stands for Superpersistent Boot Kit.

And he says: "The general purpose of MT-SBK is to provide a framework for my previous project, TinyXPB, a demonstration bootkit. This new firmware framework enables my TinyXPB to be stored and loaded from within the hard disk firmware, preventing it from being removed. Antiviruses, operating system re-installs, or even full-disk formats, nothing gets rid of it. This rootkit is designed for a major brand of hard disk and can infect the firmware from within the operating system with no physical access to the drive required." You do need admin privileges, though. "It's also completely undetectable to software running on the host computer. Once it's installed," he writes, the only way to remove MT-SBK is by replacing that hard disk's printed circuit board or connecting an EEPROM programmer directly to the flash chip and re-flashing it with the original firmware."

So what we've got is the way this industry works. We've seen it before. In the wild, evidence of this already being done by we don't know who was found. We've never found it in a drive because you can't. Once it's in, it's able to protect itself, to block its own visibility. Firmware cannot be read from the interface on purpose because companies consider it proprietary. And what this guy did was to hook equipment directly to the little eight-pin chip on this brand of hard drive and directly read out its contents. He learned ARM machine/assembly language. He figured out how to debug this thing, how to set a break point so he could check his code.

And this is where I'm saying, boy, you know, if only he was aiming at something creative rather than hacking. I mean, this is, you know, he's having fun. But he's clearly an extremely capable developer. So we went from evidence that this can be done, that news runs throughout the industry, and then independent people start doing it because it can be done. And so he's got it running. Thankfully, he's not posting it as a proof of concept because the reactions to this when we've talked about it before demonstrated this is really bad. I mean, this is something you can't get rid of, you can't even know you have.

What his code does is it replaces the master boot record only the first time it is read. So when the system comes up and boots, it reads the master boot record. His code watches that come into the cache, essentially the RAM cache in the processor on this hard drive motherboard. He replaces that sector on the fly, before it heads out through the

computer system. So it's a one-time replacement. All successive reads return the normal MBR. So any check, any look at the master boot record is fine, except the very first time it is read from the operating system. It ends up being that he's using this to bootstrap the other rootkit that he wrote which lives outside of the drive. So essentially he's moved his rootkit, or bootkit, as he calls it, from where it would normally be into the firmware. So you can't format it. You can't get rid of it. And it's working. And now we have...

Leo: You'd have to throw the drive out; right? I mean, what would you - I guess you could rewrite the firmware.

Steve: If drives still cost a thousand dollars, then, yeah, you might take the time to rewrite the firmware. But if you absolutely knew this was bad, I think you'd just toss it. Drives are so cheap these days, it's like, this is an evil drive. We're not going to use this. But you certainly could flash over the existing firmware.

Leo: Does any existing antimalware software look at firmware?

Steve: No, it can't. The firmware is...

Leo: Oh, it's prior to the software.

Steve: The firmware has no commands that allow it to be read out. And so you reverse-engineer it by accessing the ROM directly. You read the ROM contents directly. And that's the Achilles heel of the current design. The firmware is sitting in a little separate eight-pin chip, standard available EEPROM chip, when what they need to do is move the firmware into the processor so that now it's an integrated unit, and so you don't have a bus between the external firmware memory and the processor. If there's a bus, you can tap the bus and see what's going by. We need to move this memory on chip so that then it's a closed black box processor, and then we're safe.

But none of the technology is doing that now. It's just cheaper right now for them to use - it's using a Marvell multicore ARM processor and this external memory. And so it just - it pulls from that when it boots up. In the ARM is just the get-the-firmware-from-the-external-chip code, which it pulls in, and that allows firmware updates to happen because manufacturers want to be able to update their firmware. But they update it by writing it. They are unable to read it. So consequently, no AV software is able to detect it. The only thing you could do, I mean, if you suspected this, would be to arrange to be the first read of the boot sector. But that's just this one particular instance.

Essentially what we have is an industry of very vulnerable drives to this kind of attack, and so they don't all have to just work by replacing the MBR. They could do more fancy stuff. And there were some things he was never able to figure out. The firmware is compressed with a Lempel-Ziv compression, which is then Huffman encoded. And he was unable to reverse-engineer that. So he said, eh, fine, and he, like, came up with a way that didn't need him to. So there were some limits to what his own - how much time, probably more than anything else, he wanted to put into it. Whereas it's certainly the case that a state-level agency has no limitations of time and resources.

So anyway, the point is it's gone from, oh, look, we found in the wild some code that

apparently does this, to now we have a 20-year-old kid with no budget. He mentions that at one point. There was, like, something he had to do because the wires he had were too thick, and he didn't have any thin wires around.

Leo: Wow.

Steve: Being able to perform this reverse-engineering and succeed. So what we need is we need our hard drive manufacturers to address this because this is...

Leo: How would you suggest they address it? Write ROM; but, see, they'd want to be able to patch it.

Steve: They're not signing their firmware. All they have to do is sign their firmware.

Leo: Sign it, oh.

Steve: Yeah, because you can modify the firmware, and nothing is verifying. There is a - he called it a "sum eight," so maybe it's just a simple byte sum to generate a checksum to make sure that it was read correctly. But that's not a security measure, that's a reliability measure.

Leo: Maybe they're reluctant because they fear there'd be a bypass around that, and they want to make sure they do something even more robust.

Steve: Yeah. We need, I mean, the right answer is to invest in moving the firmware on chip.

Leo: Right.

Steve: Don't have it be a separate chip, where it is available for having it sucked out into a code analyzer. And he used IDA to disassemble it and create a flow graph and, you know, just sort of sat there learning ARM assembly language and figured it all out.

Leo: Wow.

Steve: Wow. And the other thing, too, is that manufacturers need to understand, if they're considering this proprietary, well, this kid has a disassembly of their proprietary firmware. So it's not very proprietary.

Leo: Yeah.

Steve: So there are a lot of reasons I can see that hard drive manufacturers want to fix this. The problem is maybe you could retrofit this by doing one final firmware update. Like I don't think you can add code signing. That would have to be built to be secure and unbypassable. It would have to be in the boot code that lives in the ARM processor so that it has burned into it the private key that allows it to verify the signature - oh, no, I guess it could have - well, no. Anyway, there are many ways you could do it. But it would - I don't think you could do a field upgrade to foreclose this unless it was like, this is the last - no. As long as the chip is outside, you can't. Someone could always suck out the contents and change it, unless they have code signing built into the ARM processor. The point is, all the drives...

Leo: Then it gets expensive.

Steve: Well, I mean, it doesn't cost much in volume. But my point was that all the drives that exist today are vulnerable. That is, this could happen in two years, as everyone comes up to speed and fixes this problem. But that doesn't help us for, like, the entire install base of hard drives that we have today.

Leo: You'd have to be, I mean, this is kind of a nontrivial thing. And I guess you could...

Steve: Yeah, this is not script kiddie grade. On the other hand, what we've seen is we've seen communities gather around these. So this guy is being ethical and not posting proof of concept. But many other people have nowhere near the same sort of ethic. They say, oh, information should be free. And so they'll start putting out firmware. And, I mean, this is, you know - watch. Six months from now we're going to see a lot more of this going on. Just makes sense. It's a tasty target.

Leo: It sure is. Sure is, yeah.

Steve: It's something new for people to play with. So I've not yet had a chance to get a full analysis of what happened at the Worldwide Developers Conference on Monday. But I did note that, in iOS 9, Apple is further strengthening the passcode input by requiring six digits.

Leo: How important is that? I mean, it's a million different options instead of 10,000, but 10,000 is still a lot.

Steve: 10,000 is a lot.

Leo: I mean, given that you have to enter it by hand, and there's a setting that says after 10 tries erase the damn thing.

Steve: Yeah. Although, for example, we've seen hacks where - and I'm virtually certain they'll fix this in iOS 9. Remember the hack where, if you put it in wrong, it powered the

system down before it could write into nonvolatile memory that you had just had a mis-entry. And so that allowed you to do the 10,000 passcodes without ever getting your phone wiped.

Leo: Right, right.

Steve: Also, four seems a little lean, just because of the other traces that are left behind. Anybody, for example, who's using the standard little 10-key pad, you'll see that pattern in extra finger grease on the screen. And so if there's only four digits, that's not a lot of ambiguity if you mix in any other information like taking a look at fingerprints on the screen. What I do is I always switch mine into alpha mode, and then at least I'm not dealing with a 10-key pad. I still have a relatively short phrase because I don't want to burden myself, and I maintain pretty good control of my phone, and I have Touch ID, which is what I normally use.

But of course after a power cycle or an update you need to give it your unlock code again. And so what I do is just use the full alphanumeric keyboard, and that way you're dealing with a much larger alphabet than just a 10-digit alphabet. But still, to me, this seems like a good - it's simple. Minimal inconvenience. And of course this is on devices that increasingly have Touch ID, so people are having to enter that less often.

I'll never forget watching you and Sarah talking about this on iPad Today, and she was saying, "I hate having to enter my - every time I want to buy an app," which she said, "I'm doing all the time, having to keep reentering my Apple ID." So she was really looking forward to just being able to put her thumb on Touch ID in order to authenticate a purchase. I loved, of course, that iPad's going to get split screen. I mean, it'll be interesting to see it. I like more features. And the keyboard improvements seem like an interesting thing, too, being able to turn the keyboard into a trackpad just by putting two fingers down. And it looks like there's some editing stuff - bold, italics, and underline - that are right there, too, which seem like good things.

And then I got a tweet this morning from Bill, who is from Twitter. @BillyInDallas is his handle. And he says, "As with all previous OS X releases, 10.11 defaults with the firewall off." So it's like, well, that's too bad. It would be nice if the Macs started shipping with that on. But maybe they're concerned...

Leo: It's off. And I think, yeah, I mean, I turn it on the first thing I do. But I also turn on hard drive encryption first thing I do.

Steve: Right.

Leo: But, you know, it would cause problems. People's stuff wouldn't work.

Steve: Something would break.

Leo: Yeah.

Steve: And the industry is getting a new certificate authority. Amazon has applied...

Leo: What?

Steve: ...to become a CA. Yup. There's an existing organization. The Amazon PKI, the public key infrastructure, will be run by Amazon Web Services. And I guess, you know, it makes sense that Amazon, that is developing such a web presence, would want to offer certs. However, it's not just for their properties. Because Mozilla and Android root stores and the Mozilla and Android processes are public, Amazon is publicly known to have applied to be included in Mozilla and Android root stores. Microsoft and Apple applications are not public, but it seems almost certain that Amazon has applied there, too. So they're intending, in their little statement of application, they've said that they plan to offer standard and EV certs, which will be able to function for server authentication, client authentication, email signing and encryption, and code signing. So sort of the core main certificate roles that we see most often. And so before long we'll be talking about Amazon as a new CA.

And speaking of new CAs - oh, and these are all available to the general public without any restrictions. We don't know, have no idea yet what their timeframe is, what the schedule - it's not been officially announced, as far as I know. It's just been people picked up on the submission of the application to Mozilla and Android and said, ah, interesting.

Leo: Why not? [Crosstalk]

Steve: Amazon's going to be a CA. Exactly, yeah. And, I mean, it fits in with their whole AWS stuff.

Leo: Yeah, yeah.

Steve: Then last Thursday a signing party was held by LetsEncrypt.org. They generated their root and intermediate certificates. Our listeners will remember that Let's Encrypt is this interesting effort that is getting ready to launch. At the beginning of the year we were talking about it being, eh, you know, sometime in the summer of 2015 was when it was expected. And this is another step in that direction.

So this is them generating their - because Let's Encrypt will also be a certificate authority. And that's the one that uses this automated agent to work with an on-the-wire protocol so that your server, a web server, is able to say to Let's Encrypt, I need a certificate. Let's Encrypt sends back a challenge and says prove me you own this domain that you're asking for a certificate for by putting the following blob on your root or wherever. And then the server that is in control of the domain does that, says okay, it's there. Let's Encrypt verifies its presence, verifies that it has control of that domain, and says, okay, good. Here's your certificate. And again, the agent does all of the configuration and cert installation, making this just virtually painless, in addition to being free.

So for all this to happen, the Let's Encrypt certs have to be trusted by all the browsers that will be connecting to servers who are getting their certs from Let's Encrypt. So they

generated a root certificate for themselves, and that's a cert which is self-signed. The root cert is self-signed. And in following standard best practices, they then generated what's called an "authority," that is, an intermediate certificate. And that one is signed by their root certificate. And that intermediate is able to issue certificates. That allows them to put the root in deep freeze, safe, so it's absolutely protected and is never exposed. And essentially the worker certificate is the intermediate one, which then is used to sign the certificates of the websites that apply.

The problem is, on like day zero, we don't trust this root. It was made last Thursday. Nobody even knows about it. So what they did to solve that problem is they cross-signed. IdenTrust is a well-known sort of industrial certificate authority that is already in everybody's root store. So IdenTrust also signed this intermediate certificate so that, not only is the Let's Encrypt root signing it, but so does IdenTrust. So what that means is that, from day one, the actual Let's Encrypt root is propagating out into all of the root stores of the various entities that need to trust it. And this is not just browsers. Browsers are of course what we think of for web surfing. But encryption is used by appliances and all kinds of things that have root stores.

So the idea is we need those things, maybe that have their root store updated much less often than our consumer browsers do, which is almost on a daily basis, we need them to be trusted. So IdenTrust gives that intermediate certificate some credibility from day zero. And then over time the Let's Encrypt root will get propagated out into our browsers. So they're moving forward. Again, no clear indication of a timeframe and schedule, but this is obviously a necessary step. I talked, I guess it was last week or the week before, about the license that a website is agreeing to. And so now we've got the certificate that would sign the certs that the websites would receive. And we'll have a podcast here before long on the protocol, which is ACME, an acronym I don't remember, but A-C-M-E is their automatic certificate management something.

Leo: Yeah, Wiley Coyote used to use them, yeah.

Steve: Yeah. And I got a kick out of this. It turns out - this was inevitable. But in pursuing this story of the U.K. selling off, now beginning to sell off unused blocks of its own IPv4 address space, a broker surfaced. So the story, the BBC carried the story that the U.K. government has started selling off Internet address blocks, that is, IPv4, the existing space. And we've talked about before how it's vanishingly available. It's getting scarce. Various - was it Egypt that was buying them? I can't remember now. But there is now a market that never existed before, when you could just ask your local Internet IP source, RIPE or InterNIC or whomever, for a block, and they'd say, oh, yeah, here you go. How many do you need? Okay, here you go. Well, obviously you're not going to pay for them because they're free. And it is a weird sort of thing that what was once given freely is now a profit center. And I don't quite get that whole logic. But, you know, okay.

So the first group of 150,000 IPv4 addresses were snatched up by a Norwegian firm called Altibox, for which they paid about 600,000 pounds. And I didn't convert that to dollars, but that's a lot of money. And apparently, if the U.K. government sells off all of the surplus addresses it owns, it's estimated that, at today's IPv4 value - I mean, it feels like we're talking about bitcoins because, I mean, it's ephemeral.

Leo: Alexa says, by the way, that's $917,000.

Steve: $917,000, okay, so just shy of a million for this 150,000 IPs.

Leo: It's pretty cool. I asked Alexa, "Alexa, how many U.S. dollars is 600,000 British pounds?" See if she can answer it again.

ALEXA: Sorry, I can't find the answer.

Leo: It's funny, I must have misstated it that time. But she said it once before.

Steve: Cool.

Leo: I'm sorry. Dr. Mom is mad at me because I keep - her Alexa wakes up, too. Sorry. Go ahead.

Steve: Everybody put a towel over your Alexas while the podcast is going on. So anyway, so the U.K. government could get up to 15 million pounds. So that sounds like, what, like maybe $20 million, something like that. Okay. So here's the background. In 1993 - so what's that, 22 years ago - they obtained a Class A network. That is, the Department of Work and Pensions in 1993 - so, okay, it's 22 years ago. That's a reasonable - that was the beginning, a Class A network. I didn't look up, like, what their number is. But what a Class A, of course, is, is that first number, you know how IP addresses are a.b.c.e - wait, a.b.c.d, where that first one is the Class A network. So they own all IP addresses, that is, this one, the Department of Work and Pensions in the U.K. owns all of the IPv4 addresses starting with some number, whatever it is.

And so they did an analysis that showed that they're only using about 70% of those Class A addresses. And a Class A network, since an IP address is 32 bits, and the first byte removes eight, that leaves 24. So 2^24 is 16,777,216 IP addresses. They're using 70% of those. And that leaves the difference, about 5 million, free for disposal. Now, I don't know. My take is this is too early to sell. We're going to see an escalation in IPv4 costs because people really would rather pay than switch. They just don't want to go to IPv6. In fact, I ran across one quote that said that companies were consolidating their use of IPv4 space and selling off the excess in order to finance the upgrade to IPv6, which I thought was wonderful.

But in digging around in this, I found a company called the IPv4 Market Group. And their website is very slick and professional looking, IPv4MarketGroup.com. And this is an international broker of IPv4 addresses. So we now have, I mean, I'm sort of wondering how this gets - who, you know, do you bid for this? Are they auctioned? Who sets the price and so forth. Turns out there's an organization, at least one, a broker. Sandra Brown is the president. And she was quoted as saying: "Regional caches of IPv4 addresses have all but run dry, meaning many firms have to look elsewhere for them. Trading in IPv4 has been brisk in Europe because the organization that oversees 'Net addresses in the region had approved policies that allowed for transfers. In the busiest months, about 2 million IPv4 addresses were being traded in Europe."

And she says: "Supply has met demand, but we're reaching a point where supply is about to fall short, and we have seen prices escalate because of that." And single IP addresses sold in volume, or IP addresses sold in volume were going for about $11,

about 7 pounds each, although quantity discounts are available for large blocks. And then the same Sandra Brown said that the sense of the industry is that another five to 10 years will be needed for most of the conversion from IPv4 to IPv6 to take place, and at this time only pilot programs are underway.

So even now, I mean, this is classic. We're dragging our feet, kicking and screaming. We'd rather buy somebody else's IPv4 IP addresses, if we need more than we have, than bite the bullet and move over to IPv6. So I really think we're going to see, as unused IPv4 space becomes increasingly valuable, we're going to see the price increase. And of course, as we said before, this represents a real technical problem because what we want is blocks of IP space to be owned by regions. So, like, for example, we want all of that Class A network's IPs at the Department of Work and Pensions to be in the U.K., used by the offices of the Department of Work and Pensions.

The reason is then everywhere else in the world, all the routers on the Internet that route these things, when they see that first byte is X, whatever that is, they all know that's as far as they have to look. X is the first byte. Send it to the U.K. And so they figure out what interface of the router goes in that direction, in the direction of the U.K., which is what the routing table also tells them, and they're done with it. But when a chunk of that Class A network is sold to Norway, suddenly it's like, oh, oh, wait, wait, yes, the first byte is X, but there's been some partitioning of this Class A network, some subpartitioning. So now we need to look deeper into the IP address in order to figure out who owns a subnet of that, and send that in a different direction.

So what's happening is, as a consequence of this, sort of the last gasp of IPv4 and this natural inclination to sell off what you don't need because it's now got a cash value, and it may help finance your painful move to IPv6, is we're seeing a rapid fragmentation of the routing tables, an explosion in the size of routing tables that has a lot of people in the industry worried. Routers have had problems before. They've been upgraded, and they've got more RAM, and of course everything's faster than it used to be. But this is a concern. And we can see now what the problem is because it's no longer as simple as, oh, if the first byte matches this, send it over there. Now we have to look deeper. And that means many, many more entries in the routing table in order to see where these small pieces of larger networks may be aimed. Really interesting, I think.

I ran across a really interesting configuration page on the Mozilla.org's wiki, which I commend to everyone who's interested in setting up TLS on their servers, web server people. I think you could probably google "Mozilla server side TLS" because the Server_Side_TLS is in the URL, it's wiki.mozilla.org slash Security slash Server_Side_TLS. And I can't tell here whether it's underscores or spaces. Must be underscores. Anyway...

Leo: Yeah, it's underscores.

[https://wiki.mozilla.org/Security/Server_Side_TLS]

Steve: Anyway, beautiful page, full of tables and charts and comparisons showing what servers offer which features. They have of course tackled the whole issue of the order in which the cipher suites are listed. They explain their logic that, okay, we gave priority to the key length of the key agreement. Then the next in line for priority is the encryption or authentication or whatever. It's all laid out there, not only what Mozilla's recommended configuration is for the cipher suite ordering, but why they chose it, how they chose it.

Anyway, I just think anybody who's - I know there's been a lot of interest in my own list that I created for my Windows server, which is, unfortunately, due to a 1K character constraint, which is really annoying. I had to, like, go through and remove things I wished I could keep because I had to stay within this 1K total string length for all of the names of the server suites, and they're not small names. They're long, you know, multi-hyphenated things. But that's how I curated my own list for my server. Anyway, I just think people would enjoy it.

A real quick note: We talked about the various forms of energy for powering vehicles, and I mentioned supercapacitors. In the news again was a different article talking about a breakthrough in the lab, still not commercial, but that's where these things start, of a graphene-based supercapacitor. And of course many people are looking at graphene because it allows for such a, well, a high-capacity capacitor, essentially.

What this article had that I just wanted to share was some numbers that were lacking from our discussion last week. Today's lithium-ion batteries, which we would like to meet or exceed, are around 200 watt-hours per kilogram (Wh/Kg). So that's the energy, watt-hours is how many watts you can pull for an hour, or how many hours you can pull a watt, essentially. So 200 Wh/Kg for today's best lithium-ion. All previous supercapacitors have been about on the order of the high 20s or low 30s in watt-hours per kilogram. So compared to 200, you know, better than a tenth, but still falling far short of what we need.

In the lab, these guys have built a prototype graphene-based supercapacitor which is delivering 131 Wh/Kg. So while still not where lithium-ion is, really getting close, 131 versus 200 for lithium-ion. So, and of course I'm a proponent of supercapacitors, as we know. They are ageless. They do not die with recycling. And as fast as you can pump power into them, they can take it. So potentially solves a lot of problems. If we got another - so this is four times better than the previous record. And if they could double it once again, then we're at 262, doing better than lithium-ion. And then if they can get it into production that's - of course it's one thing to be in the lab and something else for it to actually be on the floor, under the floorboards of our electric vehicle. But I thought it was fun to see some numbers.

Okay, so media. Leo did not have the chance yet to watch "Mr. Robot." I know what you will think. I know what you'll think when you do, mostly because the response from our listeners has been truly nothing short of phenomenal. I mean, this was a show made for this podcast. So for anyone who didn't get around to it, if you want more feedback than just mine, I mean, our listeners have been raving about it. So I'm sure, Leo, at some point you'll, I mean, it starts running, the series starts running on June 26 on USA Network. And I heard you mention on MacBreak Weekly that it was available, I think it's - I know it's available through Amazon's...

Leo: iTunes has it, yup.

Steve: And iTunes, yeah. They made it...

Leo: It's on the USA Networks site, too.

Steve: Yes. And I think even YouTube, maybe. I mean, they basically sprayed this thing everywhere.

Leo: Everywhere except outside the U.S.

Steve: Ah.

Leo: It is, after all, the USA Network.

Steve: Ah, true, yes.

Leo: So you're going to have to use TunnelBear or something if you want to watch it outside the U.S.

Steve: Okay. I also tweeted the news of an interesting-looking new Netflix series. Netflix is doing their own productions, famously "House of Cards." They've done three seasons. And what everyone likes is that they release all the episodes at once when they do this, and so people binge watch. In fact, I was watching your feed a few months ago, and one of your guys had stayed up all night watching Netflix on one of these Netflix binges. So this was released last Friday. And it's sci-fi only if we spell "sci" P-S-Y. And the trailer was immediately interesting. And when I saw the trailer, I tweeted to my followers, "Hey, Netflix has a new 12-episode series, 'Sense8.'" It's Sense, and then the numeral 8.

Now, I've seen all 12. Some early feedback from my tweet, before I had started to see it, said, "Whoa, adult content warning." And people who are homophobic will have a problem with it. Okay. So for what it's worth, I liked it, although I liked it a lot more after they calmed down, after the first two episodes. It was just gratuitous sex, I mean, sort of like I was reminded of the first season of "Game of Thrones," where it's like, okay, you know, let's get on with the plot. But the plot is interesting. And this is not a spoiler because the trailer says this, and everyone knows this when you start watching. Eight people around the world are linked telepathically and are able to share aspects of each other. And I really liked it. I mean, again, you have to put up with the first couple episodes because - and this is the - you pronounce it Wachowski Brothers? Or Wachowskis?

Leo: Wachowski, yeah.

Steve: Wachowski.

Leo: They're not brothers anymore, though.

Steve: Right. And I was going to say that I think it's one of them and his wife.

Leo: No, no, it's his sister.

Steve: Oh, his sister, okay.

Leo: Well, it was his brother. She's transgender.

Steve: Ah. See, now, Leo, I'm so glad you're tuned into the pop culture. And in fact, now that you've said that, a lot of this...

Leo: Makes sense.

Steve: A lot of this series makes a lot more sense.

Leo: Yup, mm-hmm.

Steve: Because the other thing I noted is only the gay people are having all the sex. Somehow the straight ones never get around to it.

Leo: Well, I don't know if she's gay, but she's transgender. I don't really - that's a separate status.

Steve: Well, we have some of that in here, too. So, I mean, it's racy. For what it's worth, if that doesn't offend anybody, I liked it.

Leo: I'll have to check it out, yeah.

Steve: I thought it was a great series, and it developed really well. However, I will tell you, I wanted to finish it for the podcast, so last night I watched the last two episodes. Halfway through, and this is just a teaser to see if anybody gets it, halfway through the final episode was the biggest plot mistake.

Leo: Oh.

Steve: I could not believe. In fact, I backed up and watched it happen again. And I thought, how could the writers do this? I mean, it's very complicated. We're essentially developing eight different plot lines and weaving these lives together. And the way they depict the telepathic connections is done perfectly until they completely blew it in the middle of the last episode. I just - I was stunned that they did something which, I mean, they broke all their rules that they had established. So people who are curious have a little tease for the last one.

And I did want to mention that a book that we have talked about often here is turned into a movie by Ridley Scott. And of course, Leo, you know what I'm talking about.

Leo: Yeah, "The Martian." You've seen the trailers; right?

Steve: Yes. Matt Damon? In fact, it's funny because my best friend, who's got a great sense of humor, he said they should have called it "Bourne in Space."

Leo: Oh, yeah, because it's Matt Damon, yeah, yeah, yeah.

Steve: Because of Jason Bourne, yeah.

Leo: Yes.

Steve: So anyway, yeah, there's like the official trailer came out, I think yesterday. Then there was this other weird sort of getting ready to launch, he's sending like a postcard back to Earth. And so there's two different things on YouTube. I have the links in the show notes, if anyone can't find them. So I think it's...

Leo: There's a trailer, and there's a viral video.

Steve: Ah, okay.

Leo: Yeah. I mean, I don't know how that's different, I guess.

Steve: Yeah, but it's officially produced, so it was somebody...

Leo: The video postcard's a viral video, and then the rest is a trailer, yeah.

Steve: Ah, okay.

Leo: "Bourne in Space."

Steve: And everyone knows that "The Martian" is the story of an intrepid astronaut who gets left behind, and his story of survival. So, and many people are worried that the movie will not live up to the book. I think it's clear...

Leo: It's going to be different.

Steve: Yeah. So, and I may have heard you say, and I completely concur, read the book first.

Leo: Yeah, yeah.

Steve: I mean, I will, you know, it just goes without saying. Now I always take the trouble of reading the book first because it's always so much better. "Jurassic Park" the book? And in fact, when I was watching the movie, it's like, wait, wait, wait, wait, wait, you left that out? You can't. We need that.

Leo: You have to. You've got two hours.

Steve: Yeah.

Leo: At most.

Steve: Yeah. Yeah.

Leo: But it's still fun to see it come to life.

Steve: I now want to explain something which I confused people with. Last week I used my phone to log into your computer through SQRL.

Leo: Right.

Steve: By letting the phone see the QR code which you were transmitting to me over Skype.

Leo: Right.

Steve: Which has a cool use case. But a lot of people said, wait a minute.

Leo: Why would I want to do that, yeah.

Steve: Why would I want to do that, yeah. That was one. And the other is, but I don't have a smartphone. This thing needs a smartphone? I have to carry a phone with me? I don't use a phone, or I don't have it with me most of the time. So I said, okay. We need Part 2 of mind-blowing SQRL demo. So here's my iPad.

Leo: Okay.

Steve: And now here's my iPad turned on.

Leo: Okay.

Steve: And here we have a...

Leo: I see a QR code, yes.

Steve: The QR code that you would see on any SQRL-based website.

Leo: Right.

Steve: I do this.

Leo: You tap the QR code. A little SQRL pops up.

Steve: Yes. Now it wants to confirm...

Leo: I can't quite read what that says.

Steve: It's confirming that I want to log onto GRC.

Leo: Okay.

Steve: And I am now logged onto GRC.

Leo: Nice. And that's how most people will use it.

Steve: Yes. So what that essentially means is what I showed before we call "cross-device login," where the device you are authenticating with, your smartphone, is not the device that you're actually logging into. So it's cross-device. And this is Jeff Arthur's same client. It can scan a QR code from any screen where it's displayed, or you tap the QR code. Because we wrap the QR code in an href tag, turning...

Leo: By the way, you can also click it with your mouse.

Steve: Yes.

Leo: People are saying, "But I don't have a touchscreen." Okay. Click it with your mouse.

Steve: Yes. And in fact, I wanted to demonstrate that user experience. I've been writing a Windows client. So it lives in Windows. And what happens is, as Jeff did there on iOS -

oh, and by the way, that also works on the iPhone. You tap the QR code on the iPhone, and you're logged in, which is the way it's all, I mean, 99.9% of the time that's the way you're going to be logging in.

Leo: It's just like clicking a button and saying, "I'm here."

Steve: Yes.

Leo: And it somehow identifies you.

Steve: Yes. And you and I will be going over that in detail when we do the full demo. But what I realized is, the reason this almost seems magical is that, unlike all the other, I mean, every other system to strengthen authentication, like second factor, they all make it more complicated. They all increase the number of steps that you have to go through. And the reason is you can't have one factor if it's not secure. But what SQRL is, is secure single-factor authentication. You don't need any other factors because it cannot be cracked, as far as anyone knows, using standard crypto.

So what this does is it actually simplifies the process of logging in, rather than making it more complicated, by being just one secure factor. And as you said, Leo, when you're using it, normally at your desktop, you simply go to a website. Oh, they support SQRL. You click your mouse on the QR code. Up pops a little confirmation dialogue to make sure that's the site you want to log into because people can't read QR codes. So we need the server where it's going to authenticate to, to tell us who it is in order to verify. We say yes, and we're logged in.

Leo: How is that different from a saved password?

Steve: Oh, my god. For example, the site loses all of the passwords in its database, and now you're compromised.

Leo: Right. But, I mean, a saved password in my browser. That's what somebody's asking in the chatroom: Well, that's the same thing I get when I log in now because my password's saved in my browser.

Steve: Okay. From that experience, in that case it is. But you've also given the website a secret and trusted it to keep the secret.

Leo: Hmm.

Steve: SQRL gives websites no secrets to keep. There's nothing for the web. You're not requiring them not to divulge any information. You're known as just a pseudorandom tag in the website. Anyway, we'll go over this in detail. I didn't mean to get too sidetracked.

Leo: I shouldn't have asked you.

Steve: No, well, there's lots of...

Leo: Lots of questions from the chatroom, yeah. But we'll do this soon, I'm sure.

Steve: Yes. So I did get, as I was going through the mailbag for the Q&A, found a tease-y subject line, "SpinRite Saved Newspaper Publisher's Hard Drive Data," that I wanted to share, just to give another example of SpinRite functioning. This was John McCarthy in Buffalo, New York. He said: "Steve and Crew. As the Chief Stationary Engineer" - I'm not really sure what he means by that, but he said - "a.k.a. head boiler room guy, for a major metropolitan newspaper in western New York State and a proud owner of SpinRite for many years, I've got a testimonial that's one for the books.

"Yesterday I received a rather panicked call from the head of our company's IT support crew. Usually such calls would involve cooling issues in the server room, a heat complaint in his office, or even worse, a plugged toilet in the men's room. Lo and behold, it was neither. Instead he asked if I had, quote, 'that recovery program that you told me about.' 'Sure. Why? What's up?' I responded, as I always carry a copy of SpinRite with me, though not on my tool belt.

"He went on to explain that the paper's publisher/president's Windows 7 laptop refused to boot past the splash screen. Remembering my glowing recommendation of SpinRite, he hoped he could borrow it for a last-ditch effort to recover the boss's data, as none of their utilities could fix the problem. Confidently, I gave him the CD, with the agreement that, if successful, he would purchase the corporate version of your magic disk. He agreed.

"Two hours later, bang. It worked like a charm. He was able to boot off the previously unresponsive drive, back up all the data, and even make an image of it for later transfer to a replacement SSD. The long and short of it is you should be hearing four yabba dabba doos soon, as the entire IT department was simply astounded by your excellent product."

Leo: That's the business license.

Steve: Yes, four copies of SpinRite allows - we call it the "site license" - allows it to be used within a single site in a corporation. He says, "as the entire IT department was simply astounded by your excellent product. And he finishes, "I've been listening and promoting Security Now! and the TWiT network for many years. And while I have the opportunity, I want to thank both you and Leo for allowing me to not be the dumbest guy in the server room, even if I'm the only one holding a pipe wrench."

Leo: I love it.

Steve: And he says, "Next week I'll clue them in on Harry's shavers. Thanks again. John."

Leo: Thanks, John. That's great.

Steve: Thank you, John.

Leo: Okay, Steve. To the questions we go, starting off with Louis in Chicago. He needs to hand his smartphone over for repair. I'm always nervous about that, ever since you told me that we got emails from, was it valet car parkers, that say they'd immediately go through the cars, the key rings, look for USB keys for storage in the car, get all the music, everything they need. When you hand over something for repair, there's always a little bit of a risk.

Steve: Yup.

Leo: Longtime listener. Thanks for a great show. I hope one day you'll pick up one of my questions. We did, Louis. I'm a proud owner of an Android-based Sony Xperia Z3 Compact smartphone, Z3, and I have some hardware issues with jack stereo output. No worries. It's under warranty. They'll fix it for free. But in order to get it fixed, I have to give my phone, not to Sony, but to a third party.

Steve, this has all of my Gmail accounts configured on it, apps, private pictures, your podcast, and many other things. How would you recommend preparing a mobile phone for a warranty service? My issue is hardware-related, so it might get fixed, or maybe I'll have my phone replaced with a new one, while the old device disappears somewhere with all that private data on it. I can reset it to the factory setting, but I know it's not enough. Any other good practices? Tools? What should I do? Louis in Chicago.

Steve: So this is closely related to a topic we've talked about with SSDs. And the trick is what's - I see that over in the acronym world it's called FDE, specifically full-disk encryption. And it's been available, I think since Android 3. But the problem is there has not been hardware assist for the encryption.

Leo: So the Nexus 6 does this, when it came out with Lollipop on it. And everybody complained because the hard drive, the storage access speed was so slow.

Steve: Right. So I was wondering, Leo, for an update from you on the state of the Android hardware platform. Apparently the Qualcomm chips used in many of these smartphones has an encryption engine in it. But for whatever reason, Android is not using it. So there's acceleration available which is not being taken advantage of. Over in the iOS space, of course, we know that Apple put in hardware encryption of the mass storage so that everything is encrypted as it's written and then decrypted as it's read. And the advantage of that, and the same advantage if you were using Android with full-drive encryption or full-disk encryption, is that, when you wipe the disk, and you wipe those keys, even if there was residual data there, it's gibberish. It's noise. It's not useful to anyone else.

Leo: Yeah. So I'm sure Google will address this in M, Android M, because they're obviously aware of it. The Nexus 6, which is a Google platform phone, has encryption turned on before you even get it.

Steve: By default, right.

Leo: By default. And in fact there's no way to turn it off, which is one of the things people hate about the Nexus 6 because it really does slow down disk access without hardware support.

Steve: Wow, interesting, right.

Leo: I mean, it's terrible. But I make it a matter of habit, whenever I get a new phone, to turn on encryption immediately because of course, if you don't, there's always the risk...

Steve: Before, yup.

Leo: Yeah, that something might leak.

Steve: Right, yeah. The wear leveling that's going on could swap out an unencrypted piece of memory that's still technically there. So ideally, exactly as you said, Leo, the first thing a smartphone user wants to do is turn on full-disk encryption so that all of the customization that they do of their accounts and usernames and everything is being encrypted. One thing Louis could do, now...

Leo: Well, let me tell you an answer here real quickly. Because if you use a Samsung phone - and by the way, Samsung phones are the first phones to be approved for government use, I think with the DoD. And Google has said we're going to borrow this technology. We're going to license it from Samsung. They have this Knox feature which does in fact - you kind of want a TPM; right? What Microsoft did with Windows, the Trusted Platform Module.

Steve: Right. You would like your keys to be kept really safe.

Leo: Right. And that's one thing Knox does, is it has secure encryption. It has hardware-assisted trust zone integrity management, trusted boot, secure boot. I don't know, I'm not really clear looking at this, and maybe if you look at it you could tell...

Steve: If full disk.

Leo: ...if the full-disk encryption is hardware supported. All Android devices will do full-disk encryption. But what you've said, and I agree, is you want hardware support for this so it doesn't slow you down. But I would bet it does. I would bet it does. They've done - this is so strong what they've done. And it's really, I mean, down to the point where you can't modify the firmware on the phone without tripping the Knox lock. And it'll say, hey, you can't, for instance, use Samsung's secure payment stuff. Much like Apple does. But I can't - yeah, on-device encryption. Cipher algorithm. I don't - you want to see it being hardware; right?

Steve: Yeah. And you want to say, like, full-disk encryption as opposed to some acronym we're not seeing.

Leo: Yeah.

Steve: Well, and so for Louis...

Leo: So I suspect there are solutions, but I just can't verify.

Steve: Yeah. For Louis, for what it's worth, Louis, if by some chance you had - if your phone wasn't - well, first of all, how old is the Sony Xperia Z3 Compact?

Leo: It's brand new. It's pretty new.

Steve: Oh, okay. So would you imagine it defaults to being encrypted?

Leo: No. And I think that this was the big deal. Nexus 6 was the first Android phone to default to being encrypted. And nobody's done it since.

Steve: So it was recognized as a mistake.

Leo: Yeah, yeah.

Steve: So Louis, your phone is probably not encrypted now. One thing you could do would be to turn on full-disk encryption, which is going to scramble the entire drive. It's going to overwrite all of the plaintext, all of the stuff that's in the clear, with its cryptographic equivalent, essentially, in order just to make it illegible. It's a nice way of wiping, cryptographically wiping, secure wiping existing storage that is not encrypted. Then, if there is a wipe function - do Androids have like a reinitialize, forget all of my keys?

Leo: Yeah. I don't know if it's a secure erase, but they do have a wipe.

Steve: Well, it doesn't have to be because, if he's encrypted it, it will certainly erase the encryption keys. And that's all you want.

Leo: Right. Turns it into gibberish, yeah.

Steve: You want to encrypt it, yeah, encrypting turns it into gibberish. And then erase the encryption keys means the gibberish remains gibberish. That's better, I mean, if there is a secure wipe, that would be something you could also do to prep it for untrusted parties having access to it. But if a secure wipe is not available, encrypt it, then do a secure wipe. If by any chance it is already encrypted, then just do a wipe, and it'll be safe.

Leo: Of course, this is something Apple does do is encrypt it right out of the box, so you don't have to worry about it. I remember Google saying they were going to do this. I just can't remember with which version of Android they turn this on. My guess is it's M, not the current version, Lollipop.

Steve: And we know that, to not be a performance hit, it'll need hardware assist in order to run at the same speed, yeah.

Leo: Right, right. And they probably wouldn't do it until then. So, yeah. Let's move on to the next question, which comes from Funchal, the Madeira Islands in Portugal, where of course Madeira wine is made. Marco Silva wonders about Apple's routers: Steve and Leo, listen to the show. Love it. Listening since the very beginning. We've been hearing almost weekly about security problems on cheap commodity routers.

Would Apple routers, like Airport Express, be a good alternative to flashing a normal router with OpenWRT, DD-WRT, or Tomato? I don't remember hearing Apple routers mentioned on your detailed explanations of the various problems affecting several router brands. And should they be affected by a problem, I think Apple would provide an update almost immediately. Keep up the good work for the next hundred years. Marco. Thank you, Marco. Yeah, I mean, Apple does keep their routers up to date, absolutely.

Steve: It was really an interesting question. And it caused me to dig down because it is true. I'm not, in my normal course of seeing what's going on and gathering information and news for the podcast, I'm not encountering it.

Leo: No.

Steve: So I found a history of CVE exploits and vulnerabilities; and, sure enough, the Apple routers are solid. The last known problems were I think several years ago, and those were noncritical. They were just, yeah, we ought to fix this kind of thing, not horrors like open ports of, like, management being exposed to the Internet. And before that it was five years ago. So I thought Marco raised a really good point. If you're somebody who wants a turnkey solution, not comfortable, for whatever reason, doing flashing of firmware and updating to the various open source alternatives where we know

and believe that they are substantially more secure than, well, I mean, than manufacturers are just dumping into their commodity hardware and not caring about, the Apple Airport routers, I think...

Leo: That's what I use.

Steve: ...really make a great, secure choice.

Leo: They also work better.

Steve: Probably a little more pricey.

Leo: Oh, it's like 200 bucks. But the thing is, Apple's not in the router business. They're in the hardware, you've got to buy everything Apple makes business. So they're going to keep that up to date. They have a strong interest. Whereas Linksys, hey, we got your 40 bucks. You'll be back. But Apple, you know, they're more expensive. And so they patch them, for sure. They also eschew some of the common router technologies that have caused problems like WPS. There isn't a pushbutton WPS on their Apple router. They use a different system. They don't use UPnP. They use another standard, but it's I think much more secure. So I think they do it right. I certainly feel a lot safer using an Apple router.

Steve: Yeah. And I had never thought about it. But when I went looking, it's like, I'll be darned. It's like, you're right. Apple is - there are, like, no exploits.

Leo: You never hear about them. On the other hand, you can't put OpenWRT or DD-WRT on them. They're not...

Steve: Right, so the idea would be leave it the way it is.

Leo: Yeah. You can rely on Apple for your updates.

Steve: It's fine, yeah. And so no fancy features that those other firmwares may offer.

Leo: It's a pretty good router. I don't think, yeah, I mean, I don't think their AC version does beamforming. It is MIMO. I mean, you know, I think they're pretty good. I've been very happy with them. Very reliable.

Bob in Pennsylvania has a question about the NetUSB bug. Actually, it's one of two: I have one of the vulnerable Western Digital routers with a USB connection port on a LAN segment inside my home. I connected one of my PCs to the router, ran the probe test you run at ShieldsUP! for port 20005. It showed up as being stealth. Whew. I think it's because the vulnerable router is one of two I have behind my

WAN-connected router. Does my WAN-connected router protect my USB-enabled router from this vulnerability? Thanks for your weekly podcast. I've been listening from the first episode and usually learn something new every week from your efforts. I have an interest in staying safe on the web and appreciate what you do.

And that goes with Charlie Kelly from Granbury, Texas. It's near Fort Worth. He says: Listener since Episode 1. No, I haven't listened to them all, but most. After all, this is a 10-year commitment we're talking about here. That's true. That's true.

I have one of the mentioned routers with a USB port. I'm using it as a NAS with a 1TB portable unit attached. You guys have made me quite conscious of network set-up and security. What if I put my NAS wireless and wired router behind another router? Would that obviate the vulnerability, while allowing me to keep my NAS? I'm a high school math and physics teacher, so I have limited resources. Golly, Charlie. That's not how it should be, is it.

Steve: No.

Leo: You should have unlimited resources.

Steve: Yeah. We need him.

Leo: Mm-hmm. Thank you, Charlie.

Steve: So the answer to both Bob and Charlie is yes. We've talked at various times about the idea of putting NAT routers in series. And if you do have known vulnerabilities, as both Bob and Charlie believe they may, on a router, where for whatever reason they have features, or like in Charlie's case he's using it as network-attached storage with a terabyte of USB-connected storage, and it's not safe on the 'Net, put it behind a router. It will get an IP address from the router that's on the WAN, and it in turn will give IP addresses to everything behind it. So NAT routers operate in series without any trouble at all. I would say...

Leo: Don't you usually want a bridge, though? I mean, you're still protected if you bridge. You don't have to do double NAT, do you?

Steve: You would be protected, although you can't bridge the WAN side.

Leo: No, no, you've got to bridge the internal one, yeah.

Steve: Yeah, exactly. So although you have to make sure that the NAS function, like, worked in a bridging mode. I don't know whether it would or not.

Leo: Right. I think it would. You're still assigning addresses, it's just being done by the first router, the external router, not the internal router, yeah.

Steve: Correct. Correct. And I would say to Charlie...

Leo: I was told you should never double-NAT, that it's a bad idea. That it's always better to have only one, and only one device doing DHCP on your network.

Steve: I don't know why that would be because we've talked about the benefits. There are, like, various security benefits for, like...

Leo: Yeah, you make a triangle.

Steve: ...creating isolated networks and things.

Leo: Okay.

Steve: We have a lot of listeners who are double-NATed.

Leo: Well, double-NAT, by all means. All right.

Steve: I would say to Charlie that you can verify whether that router is actually vulnerable. He didn't specifically say whether he had done the port probe of port 20005.

Leo: He has to do it on the outside, though; right? Yeah.

Steve: Yes. So while it's connected to the LAN, if it is, you can use the ShieldsUP! facility at GRC.com to probe that port, 20005. If you're not vulnerable, or if you have a feature to turn that off inside your router, then maybe it's exposed on purpose, and there's a way of disabling it. So again, I'm sort of being sensitive to Charlie's expression of limited resources. Maybe he doesn't need to spring for another NAT router, if the one he's got can be disabled or isn't vulnerable in the first place. And the port probe at 20005 will tell you that.

Leo: Yeah. Don Filupeit in Scottsdale, Arizona, found his manufacturer's PKES Vulnerability Mitigation. Oh, boy: In reviewing my owner's manual for my PKES - is that the keyless fob?

Steve: Yup. Passive Keyless Entry System.

Leo: Got it. Passive Keyless Entry System-equipped car. I discovered there's a technique to shut off the radio part of the key fob for the purpose of conserving battery life in the fob itself. The method is to press and hold the lock button while simultaneously pressing the unlock button twice. A tiny LED in the corner of the fob - I've got to try this. Wait a minute. I can get my fob out here - flashes four times to signify the procedure was successful. I tried this, then walked up to the car with the fob in hand. It did not respond to me touching the door handle for unlocking. I presume the button still works. I hope so. To cancel the procedure, you press any button on the fob - oh, I see - and it returns to normal operation. So the idea is this is a temporary fix.

Steve: It puts it to sleep.

Leo: Yeah. This seems to be a more convenient means of mitigating the range extender vulnerability than either removing the battery or obtaining a Faraday pouch and its attendant management problems. The car is a 2015 Toyota Highlander. What do you think?

Steve: So this was meant to be a proxy for many people who tweeted or wrote, who said...

Leo: Yeah, it works. Well, at least it blinked.

Steve: You're kidding.

Leo: It blinked an LED. I don't know. My car is not nearby.

Steve: And it's not a Toyota Highlander.

Leo: This is an Audi. It's an Audi.

Steve: I'm sure it's not a Toyota Highlander.

Leo: But I can't actually be sure that it worked because it blinks the LED anyway.

Steve: Oh, okay. So, okay. So this question was a proxy because, as a result of our podcast, a number of people went to the owner's manual. And lo and behold, the manufacturer had considered this. Maybe not, I mean, they weren't doing it for preventing this particular hijack of their system. But they had a "how to conserve your battery."

Leo: It's in the manual.

Steve: If you're not - yes. If you're not comfortable with the whole...

Leo: What a concept.

Steve: ...radio self-unlocking thing.

Leo: Yeah, yeah.

Steve: Other people said, hey, I went through my screens in my car's UI, and I found an option that turns that off. And it's like, okay. So I just wanted to say to people, RTFM. It turns out that, if you've got a fancy car, there's I would say better than a 50-50 chance that the manufacturer knows how to help you disable this, probably on a basis that you like. Now, frankly, I'm not sure that I think holding down the lock button and pressing unlock a couple times is easier than sort of - the idea of it being in the Faraday bag I just sort of like. It's just, okay, now it can't hear anything.

Leo: Or get a purse or a briefcase that's lined with that stuff. And then when you throw it in your briefcase or your purse, there it is.

Steve: Yeah. And if you're leaving your keys at home, that's where you leave the little pouch, and you just toss them into the pouch and sort of close the open mouth. So mostly I wanted to say to people, hey, it didn't occur to me when we were discussing this on the podcast. But from the feedback I've had, many, if not - I can't say "all." But people are discovering, I'll put it that way, when they looked more closely, there were features. And they may have noted them, but not cared because they figured, hey, this is secure. Now that we know it's not secure, maybe go back and turn that feature off, if you have the option.

Leo: Yeah. While they say it's to conserve battery, it is interesting that they offer this option.

Steve: Yeah.

Leo: I wonder if they maybe - they kind of knew.

Steve: Uh-huh.

Leo: I mean, all you have to do is think. I mean, if you did a thought experiment, oh, you know what, this could be insecure maybe.

Steve: Yeah.

Leo: From the Twitter, Jerry Voelker, @jvaudio, writes: Hello @SGgrc. What is the "Elephant Diffuser" that Windows had and no longer has? What?

Steve: Okay. So as it happened, I've been reading about this. So when I saw the tweet I thought, oh, that's an interesting topic. This was a technology in BitLocker which Microsoft somewhat controversially removed. And it was a security feature that the original designers put in which, similarly to what we were just discussing with full-drive encryption on a system that has no hardware assist, this increased the security of BitLocker, but had a substantial performance penalty overhead.

So here's what's going on. We've talked about how, when we do encryption, we must encrypt and then authenticate. So, which we've talked about several times. The idea is, when you're encrypting, you do the encryption, then you add authentication information which allows you on the decryption end to verify first with the authentication that there have been no changes made. And then if it authenticates, you decrypt. The reason is that bad guys are tricky. And it turns out that it's actually possible, and we've discussed this in the not-too-distant podcast, to make changes to the encrypted data which will have a known effect when it's decrypted. So even though the bad guys don't have the key and aren't able to see the plaintext, there are ways to maliciously alter the ciphertext to achieve an end when it is decrypted. Thus we authenticate.

The problem is authentication increases the size. It's something you tack on the end, extra, sort of like a checksum-y thing, extra data which you then apply against the original encrypted content. Now think about a hard disk. We're encrypting sectors or clusters. We're encrypted fixed-size units. There's nowhere to put authentication data. You can't, like, borrow a little of the next sector because you can't alter it. So authentication on full-drive encryption has always been a problem.

And what the Elephant Diffuser is, is an answer to that. It is a layer of additional complexity which makes that particular attack, the unknown key change the ciphertext to make a planned change in the plaintext, it renders that far more difficult. It also takes time. And Microsoft decided, eh, we're in a hurry, so we're going to take out the Elephant Diffuser. So BitLocker no longer has it. It's not a crucial flaw, but it does weaken what they had previously. And again, I'm not going to do any conspiracy. I'm just going to talk about the technology. So that's what it is.

Leo: Question 6, CyborgX in Bangkok reacts to Steve reacting to Google's Project Soli. That was one of the advanced technology projects you liked.

Steve: Ooh, that's the wave your hand in the air.

Leo: From Google I/O last week. Steve, been years since I've written to you, but I still listen to or watch Security Now! when I find the time. Just thought I'd mention something in case you don't know: LeapMotion.com. Three years ago they made an IR-based human interface radio, similar to the radio-based Google one you talked about in the episode. Keep up the great work and the show.

Steve: So I do know about Leap Motion. If you look, if you go to the site and look at the video, it looks very much like what Google has with Project Soli. Theirs...

Leo: Except it doesn't work as well. Because I bought one.

Steve: Well, yeah, several things. It's IR vision. So he says "IR-based human interface." It's IR. And what I didn't say, I should have articulated it when I was - actually one of the main reasons for my excitement is that Project Soli can inherently cost nothing. I mean, it is subject to complete solid-state integration. It's some traces on a circuit board that are the 60GHz antennas and a chip, which we know cost nothing. The Leap Motion solution, being IR vision, is always going to be expensive. And so, again, I didn't say it. I should have.

The thing that just winds me up about the Soli project is it can be everywhere for free because the technology can end up being reduced to nothing in terms of cost. And of course, ultimately, that sets the price of consumer products and determines, like, are we going to have it in our watch? Are we going to have it in our pad and behind our keyboard and so forth? So that's something that I hadn't mentioned, that his mention of Leap Motion reminded me of, is that, yeah, there are other vision-based systems. But this thing, because of the way it works, that it is doppler, essentially high-frequency doppler radar, it can cost nothing. And that really - that, like, makes or breaks whether someone can afford to add it to their product.

Leo: Actually, his mentioning Leap Motion reminds me of how disappointing Leap Motion was. And I hope Project Soli is better than that. Looks like it is. But we'll see.

Steve: Yeah, the nature of doppler radar is such that you get very different information. For example, the idea of very fine motor movements being discernible by the technology, you'd have a very difficult time doing that with vision; where with radar I think you've got a much better, sort of fundamental sense input to then be processed by their image processing pipeline.

Leo: Question 7 comes to us from Paul Toal in Yorkshire, England. He wanted to follow up on credit card eCommerce probing: Steve and Leo, I love SN, and I've been an avid listener for a couple of years now. Great podcast to listen to whilst running. In your most recent Q&A, a listener was asking about how you could stop the online credit card spamming payments, remember, the ones for a dollar, et cetera, just to test the credit card.

I work for a large enterprise software firm. We have a lot of different software solutions, including many middleware offerings. One of our products offers real-time fraud detection and prevention and sits within our Identity and Access Management portfolio. A few years ago we had a customer who had exactly the same problem as your listener described. They sold flowers online and found that many credit card scammers were using their site to test stolen credit cards. It was also costing them money processing these fraudulent payments.

Steve: Right.

Leo: We provided them with our real-time fraud detection and prevention engine, which looks at transactions in real time and identifies fraud based on a whole range of factors, including user, device, location, behavior, and predictive analytics. Sounds like profiling to me. You use the policy engine to configure the rules and policies for what you need. Doing this the customer is able to build policies like looking for multiple small transactions from the same device, or multiple small transactions from the same location in a short period of time. They can then use the engine to take action like blocking or referring the transaction. This can happen on the online eCommerce site before referring the end-user to the credit card payment processor and therefore avoiding any charges. This customer has seen a huge drop in their credit card charges, as they have managed to spot many of these bogus transactions. Keep up the good work, and thanks for all your podcasts.

Steve: So I appreciated that because I realized that of course he's right. There are such services. And when I responded, well, there's really no way, I was thinking in terms of a single transaction, like blocking a single transaction. But certainly, if you look at transactions in the aggregate, if some scammer somewhere is picking on your site, exactly as is said here, there will be, like, they're coming from the same IP, the same single IP, or a device that's got all of its headers, you know, we've talked about header fingerprinting where you hash all of the stuff, all of the request headers coming from the device doing the requesting, and that generates a fingerprint which is more often than not unique. There are clearly things that a system could lock onto if it wanted to proactively prevent this from happening.

So while it's not possible to block a single payment, I appreciated Paul's mentioning that there are systems that do more of a heuristic analysis. And while maybe the first three would get through, the system could say, wait a minute, three $1 transactions from the same place, that's not characteristic of the purchase patterns on our site, and then just start returning that the cards had been declined. Pretty soon the bad guys are going to say, wait a minute, why are all of our cards suddenly declined? Then they'll go test their cards on somebody else's website.

Leo: These people are no fun.

Steve: Yeah, we're not using them anymore.

Leo: Tom Walker in Littleton, Colorado, offers an HTTPS mystery: Dear Leonator and Steverino. Honestly, he wrote that.

Steve: He did.

Leo: One of my clients called with a problem where the computer at her Bingo Hall, which was a replacement she just brought from home, was giving her an error when she went to her website. The error was "Certificate has expired or is invalid." I couldn't figure out the problem over the phone, so I made the one-hour drive to the Bingo Hall. The error occurred at any - I know what's wrong. But anyway, I'll go on - at any HTTPS site, not just her website, but only from that computer. She said the

computer worked fine from her house, and it wasn't until she took it to the hall and booted it up that the problem happened.

So I rebooted the machine, and the BIOS said the date/time was not set and to use the operating system to set it, and to replace the battery if the problem persists. So I pressed F1 to continue the boot, verified the certificate problem was still present. I then set the date and time in Windows - it was set to 1/1/1980 - and rebooted. And bingo, and the certificate errors were gone. And I mean bingo.

Steve: Yeah, I was going to say.

Leo: I get this question fairly frequently on the radio show. It's always this.

Steve: Yup. And we've never talked about it on the podcast.

Leo: Yeah. And, you know, the first sentence I knew what was wrong. I mean, I've seen this so many times before. I do get some tough questions, though, Steve. One of these days I'm going to have to rope you in on the radio. We had a weird one this week. But anyway.

So the trouble was apparently related to the date/time being set to 1980. Likely the BIOS battery was dead. It was an old refurbished computer, and when she drove from her house it lost its date and time. But 1980 does not make the certificate expired because it had an expiration date of 2016. So what's the deal? Does it have some prejudice against certs that are good for 36 years? Or does it just go back in time, putting itself in 1980, and say "What the heck is TLS?"

Back in my day, we didn't care about security. We gave out our SSNs to anyone, and we had our identities stolen all the time, and we liked it. We liked it just fine. Tom Walker, Littleton, Colorado.

Steve: So, Tom, the mystery here is that certificates not only have an expiration date, they have a start date.

Leo: Yeah.

Steve: A not-valid-before and a not-valid-after. So certainly back on January 1st, 1980 was before the certificate had been issued. And so the issue date of a certificate is typically what's used for the not-valid-before date. So thus it was invalid, technically not because it was expired, but because it didn't think it had been born yet.

Leo: Simple enough. One last question for you, Stevie.

Steve: Yup.

Leo: And this comes to us from Robert S. in Florida with the Clever Idea of the Week Award for his solution to PKES vehicle hacking: What if manufacturers were to build in some sort of gyroscope or accelerometer into the key fob? This could detect when the key fob is being moved, shaken or jostled. And then they'd just prevent the key fob from responding to a vehicle ping unless it's been activated by movement.

Steve: I thought that was just so clever.

Leo: There's like a million ways they could do this. It's just annoying they didn't.

Steve: Yeah. If it's sitting in the bowl, the key bowl next to the front door, or hanging on a key hook, or not being carried by you as you're walking toward the door, the key fob could know that. It could realize that it's being moved around. It's like, oh, okay, no, I should start responding to pings. Now, this isn't a complete solution because it doesn't solve the problem of the car owner walking away from the car and, while still walking, being hit with the amplified ping. The car responds and opens up its door. The bad guy gets in, does it again, starts the engine and drives away. So it's not as secure as, like, really solving the problem. But as you said, Leo, it's just one other thing that could have been done, if anyone was worried about this.

Leo: That's the problem. Nobody thought about it. Or if they did, they decided to ignore it.

Steve: Yes, it's difficult to understand.

Leo: I'll let you know if this...

Steve: And they thought, oh, well.

Leo: I'll let you know if this thing works.

Steve: Yeah, if that little flashing light...

Leo: That would be good. But you'd have to remember to do it each time, which I...

Steve: Well, and my guess is that, since you don't have a Toyota Highlander, nor any brand of Toyota, it's probably just flashing to say, oh, yeah, the battery's still good.

Leo: Hello. Hi, Leo.

Steve: I think that, you know, my guess is that every make and model is going to do

something different. I just wanted to say to our listeners, hey.

Leo: Read the manual.

Steve: People have been discovering that there are disable systems built in that they just didn't think they needed. But now, yeah, probably.

Leo: We didn't mention that the reason that the computer went back to 1/1/1980 is that's all BIOS-based PCs go back to 1/1/1980 because the PC era began in 1981.

Steve: Yup.

Leo: So the counters start - I don't know why they picked 1980 but probably that's when they were designing the BIOS was 'round about then.

Steve: Yeah. And they said, well, we need, you know, we want it to run forward as long as we can. There is a clock chip in those that was the original basis. And then I guess probably the BIOS reads it. I don't remember. It's a funky chip, though. It's like a calendar chip where the hardware itself maintains days, months, and years. So it may just be a feature that they've never changed.

Leo: Yeah. Somebody in the chatroom says their BIOS goes back to 2000. Macintoshes go back, I want to say 1904. But that's because they're UNIX-based. And UNIX - actually, no, wait a minute, 1970. That's when UNIX begins, 1970.

Steve: Yes, yes. And of course '36, '36 is going to be a new year of doom.

Leo: A problem, yeah, because we only have 32 bits.

Steve: Yup, and we're burning them up, just like our IPv4 address space. We're fixing to run out.

Leo: A second, a bit a second, every second. Steve Gibson is at GRC.com. That's where you'll find SpinRite, the world's best hard drive maintenance and recovery utility. That's how he makes his bread and butter. But there's so much free stuff there. He's so generous. It's also where you should go if you have a question for Steve, GRC.com/feedback; or you can tweet him, @SGgrc, that's his Twitter handle. You can also get 16Kb and 64Kb audio versions of the show there. You can get transcripts, fully human written, fun, wonderful transcripts at GRC.com. We have, at TWiT.tv/sn, audio and video, as well. And you can also get it wherever you get your podcasts, like iTunes and so forth. You can watch live - 1:30 p.m. Pacific, 4:30 p.m. Eastern time, 2030 UTC - every Tuesday, right after MacBreak Weekly. And I think that's everything I have to say.

Steve: Mentioning the transcripts, I'm reminded that I got a panicked, horrified email from Elaine, our wonderful transcriptionist, saying that she was driving somewhere and happened to look up at a billboard or something, and she gasped and realized she had spelled Segway incorrectly.

Leo: Oh, no. She spelled it G-U-E.

Steve: For the last three podcasts. So what I received was corrected transcripts for every use of the term "Segway," with an apology. I said, "Elaine," you know, and she said, "I'm so, so, so, so, so sorry." I said, "Hey, who could ask for more than you recognizing that there was a problem, and you fixed it." I said, "When I'm posting this week's transcript, I will update all the other ones and correct the little typo."

Leo: Who would have thought we'd say "Segway" in three different episodes? Yeah, Segway is a brand. It's spelled W-A-Y. And I think they want you to say, "What's a Segway?" And you can say, "Well, they're about 60 pounds."

Steve: Okay. That's a bad one.

Leo: Yeah, that's the old Henway joke. Hey, thank you, Steve. We'll see you next week, right here.

Steve: Is this your last week with us next week? Or do we have you for longer? I keep seeing you doing the extra recording of the Tech Guy show.

Leo: Yeah, we started a little early for that. Let's see. I'm leaving - the first one you'll miss me will be July, like, 1st, I think.

Steve: Oh, we've got you for a long time.

Leo: We've got plenty of time. I'm leaving June 27th, back July 14th.

Steve: Great.

Leo: Thank you, Steve.

Steve: See you next week. Thanks, buddy.

Leo: Bye-bye.