Listener Feedback #212

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-507.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-507-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We have questions. We have answers. We'll talk about the latest security news, too. It's all next. Security Now! is now.

Leo Laporte: This is Security Now!, Episode 507, recorded May 12th, 2015: Your questions, Steve's answers, #212.

It's time for Security Now!, the show that protects you and your privacy and security online. And the guy who really makes this show happen, Mr. Steven Gibson, is here from Gibson Research Corporation. Steve was the first guy to find spyware, name it - he takes the credit for naming it.

Steve Gibson: Far as I know.

Leo: Yeah. And also the first to write an antispyware tool.

Steve: Yup, OptOut. Remember that, way back then? Wow. Wow.

Leo: One of my oldest, dearest friends. And we are always glad to get together on a Wednesday and talk about security. Hi, Steve.

Steve: Or even on a Tuesday sometimes.

Leo: Oh. Yeah. You know, I keep showing up on Wednesday, and you're never here. So that explains a lot, yeah.

Steve: Ah, well. We've found each other.

Leo: Used to be Wednesday. For, like, eight years it was Wednesday. So you understand my...

Steve: Yeah.

Leo: Takes me a while to get used to the change.

Steve: That's all right. So, well, and you've got your whole patter, you know. And so it's easy to just slip back into that patter. It was fun, speaking of grooves, listening to John Dvorak talk about his 78 records and...

Leo: Wow, I didn't know that about him. John is a very - is an interesting dude.

Steve: Yeah, he's, you know, you've got to kind of pry it out over time.

Leo: Yeah.

Steve: Otherwise he's just grumbling about something.

Leo: I've known John longer than I've known you, and I had never heard that he was an audiophile or collected records or had a turntable, for that matter.

Steve: Yeah, he's sort of low key that way.

Leo: Yeah.

Steve: So we've got a Q&A today. Neat questions from our listeners, as always, and a bunch of interesting news to talk about, too. I just got a kick out of the fact that our friend James Clapper's lie that he just blatantly spoke to Congress during that committee meeting isn't going away. And of course I did want to mention the Appeals Court ruling from the Second Circuit Court in New York. Also we've got a new proof-of-concept malware that hides up in GPUs so as not to be discoverable or visible to any standard CPU malware detection. News of Europe's Smart Grid crypto being surprisingly dumb. And then an odd report about the short shelf life of SSDs, which I'm skeptical of. But there were some interesting tables in this report that I want to go over. And we've got some miscellaneous stuff, and our Q&A. So I think all kinds of fun for our listeners.

Leo: Yay, Stevie.

Steve: As always.

Leo: Got to remind everybody, Steve does all the work on this show. He puts all this stuff together. I just show up and listen. But like you, it's all about learning, and I sure do learn an awful lot. All right.

Steve: So our picture of the week in the show notes, I just got a kick out of.

Leo: I love this. You know I'm a chess player.

Steve: I know you are. I thought of you immediately. Our friend Simon Zerafa saw it somewhere and forwarded or tweeted it to me. And apparently it's not an actual game. I mean, I notice that a black pawn is missing where the queen is. And I did see in the comments of the person who originally tweeted it, a bunch of people criticizing it, saying, oh, you know, wait, black has two more moves than white has had, or something or other. But I don't think it was ever meant to be serious. It was just sort of - it's a sort of a spoof on the notion of CAPTCHAs. And for those who aren't seeing the show notes, it's a classic chessboard where the game has progressed to a certain stage. And so it says, "Black plays. Checkmate in one move." This is a...

Leo: Now, I'm curious, because I see it, and I immediately see the move. But I'm wondering, if you're not a serious chess player, is that - it might be too hard for a CAPTCHA.

Steve: So we're talking the queen slides down two; right?

Leo: Yeah, yeah.

Steve: And puts the king in mate and is protected by the black bishop.

Leo: Exactly, queen F2 mate.

Steve: Yup. Yeah.

Leo: But any serious chess player would. But it couldn't work as a CAPTCHA because how many people - not everybody knows how to play chess; right?

Steve: Ah. Very good. Very good point, Yeah.

Leo: Yeah.

Steve: Yeah. So I guess, well, we could call it a filter.

Leo: Yeah. And if it were Chess.com that the site was using for the CAPTCHA, then okay.

Steve: Ah, that'd be perfect, yup.

Leo: I love that.

Steve: Okay. So, news. I just - someone sent this to me, and I appreciated it because I don't normally read TheHill.com, which is, like, serious, deep...

Leo: Oh, I love it.

Steve: ...political insider stuff.

Leo: It's for wonks. It's great.

Steve: It really is. Actually, while I was on that page, I looked down the right-hand column of other stories, and I thought, oh, I could get lost in here because...

Leo: Yeah, yeah.

Steve: ...there is other really interesting stuff.

Leo: Oh, it's great, yeah.

Steve: Especially after Sy Hersh's bizarre reporting over the weekend.

Leo: What did you think of that? I just read that.

Steve: I don't know what to make of it. I can't.

Leo: The 10,000-word piece that Hersh published in the, I think, the London Review of Books, in which he claims to have sources that demolish the President and

Security Agency's story of the Osama bin Laden killing.

Steve: Yeah, it's like what we were fed was a fairytale of how this all happened. But, and, now, Seymour Hersh has had some problems in the past, so his reporting record is apparently not flawless.

Leo: But he was the reporter on the Pentagon Papers.

Steve: And he is listed as one of the top 100 factual reporters of all time. So, I mean, and that's one of the reasons. If it were not Sy, people would have just blown it off as a crank, you know, I mean, just nothing. But he says, I know how to report facts. I have a reliable source. I did some confirming. And basically the story that we were all told about how we were tracking Osama bin Laden's courier back and forth and finally found the place - and in fact, after this came out, NBC independently found two sources of confirmation that someone walked in a year before with the location because he wanted to collect the $25 million reward being given for any information leading to Osama bin Laden's capture.

Leo: Basically, the premise is, the story we were fed, particularly in "Zero Dark Thirty," Kathryn Bigelow's movie, but also in "No Easy Day," Matt Bissonnette's story as one of the members of the SEAL team, was kind of fantasy, concocted for a number of reasons, for political reasons, to protect the Pakistanis.

Steve: Provide cover.

Leo: Provide cover, but also to the glory of the Obama administration. And Gates, who was the Secretary of Defense at the time, was livid, apparently, about the leaks that came out of the White House. And eventually the White House just fabricated, apparently, a story which didn't hold water and lots of additional changes had to be made.

Steve: Yeah, I remember there were some sort of things like...

Leo: I find it highly credible. I hate to tell you.

Steve: You know, there was like the fog, they were saying "fog of war."

Leo: Oh, we didn't know, yeah.

Steve: Where the facts weren't quite straight. There was something about a computer, and it turns out there were never any computers.

Leo: They didn't gather crap from there.

Steve: Yeah.

Leo: Well, the real, the bottom line on this, and this is the thing that - and people are saying Seymour Hersh's story is thinly sourced, you can't deny that.

Steve: No.

Leo: And who knows what this guy, this source of his, maybe he had an agenda; right? But the bottom line was that the Pakistanis had imprisoned Bin Laden since 2006, and he was ill.

Steve: And were holding him for negotiation leverage.

Leo: He was frail. He was not running al Qaeda. He wasn't doing anything. He was in prison.

Steve: There were no couriers coming and going.

Leo: And the whole thing was fabricated so that - it was essentially an assassination action, and neither the SEALs nor the administration wanted to acknowledge that, so they fabricated this much more elaborate story around it.

Steve: Right, and something about him not actually having an AK-47 that he defended himself with.

Leo: He didn't shoot at them. They went in there to get him.

Steve: Yeah.

Leo: But who knows. We may never know.

Steve: But how we got off on this...

Leo: Well, you mentioned it. And I actually read that yesterday, and I thought, wow, I wonder what Steve thinks about this.

Steve: Well, and it is a fascinating question. And because I saw it in the right-hand

column of TheHill.com, because of course naturally this is generating a huge amount of upheaval. I saw one of the little blurbs there said that the Navy SEAL who actually shot Osama, he's been brought out of wherever he's been and denying all of this and denying Sy's side of the thing. But of course you'd expect blowback and a defensive reaction. So who knows. But what I love is that our listeners will know how infuriated I became at that video of the testimony where Senator Ron Wyden asked James Clapper, the director of National Intelligence, pointblank, whether any American material was being collected by the NSA. And remember, this was three months before Edward Snowden's first revelations appeared. And so, and in fact, we played the video on the podcast because I said, "Leo, you just have to look at this guy just saying no."

And anyway, so the point is that there was sort of an informal roundtable meeting, I think Friday. And James Clapper's attorney, who was there, I think representing him, I don't think Jim was there, said "Jim Clapper wasn't lying when he wrongly told Congress in 2013 that the government does not wittingly," which was, you know, James Clapper's words...

Leo: There's the fudge word.

Steve: "...wittingly collect information about millions of Americans. He just forgot."

Leo: What?

Steve: Is what the attorney is now saying. Yeah, he forgot.

Leo: Watch that video again. It shouldn't, like, A, no "unwittingly" showing up there. He says, "No, we do not do that." And he didn't seem too forgetful, either, for that matter.

Steve: Right. And then remember we had fun with what he said a little bit later when pressed because later he said, after the fact, he said that it was the least untruthful possible answer - remember "least untruthful"? It's like, oh, okay - given the secrecy of the program at the time. So there his position is that he knew the truth, but he just couldn't say it in an open Senate hearing with cameras rolling. So this article says, "During a panel discussion on Friday..."

Leo: Well, that I believe, by the way.

Steve: I do, too, yes. "During a panel discussion Friday, Robert Litt" - who's the attorney - "said that Clapper just didn't have a chance to prepare an answer for Ron Wyden and forgot about the phone" - forgot about - "the phone records program when asked about it on the spot." Now, when I read that, I remembered that they also gave him all the questions the day before. And then...

Leo: He forgot to read the questions, is what he forgot.

Steve: And then Litt, continuing, "We were notified the day before that Senator Wyden was going to ask this question, and the director of national intelligence did not get a chance to review it," Litt said.

Leo: Yeah, he didn't read it.

Steve: And then he said...

Leo: I didn't do my homework.

Steve: Then he said, "He was hit unaware by the question." Oh, and then here it is: "After the hearing, I went to him and I said, 'Gee, you were wrong on this.' And it was perfectly clear..."

Leo: Oh, please.

Steve: I know. "And it was perfectly clear that he had absolutely forgotten the existence of the 215" - that was that section of the Patriot Act - "program." And then here's the final one. "Litt, he said, also erred" - the attorney is saying - "after the hearing by not sending a letter to the panel to correct the mistake. 'I wish we'd done that at the time.'" Which, I mean, so this thing is just such a nest of trying to work your way out of changing your tune, and he forgot. And, boy, isn't it, you know, I went to him, and we both agreed that he answered that wrong. And, boy, you know, we should have sent a note to the panel correcting the testimony, but I guess we forgot that, too. So anyway.

And speaking of Section 215, as you covered on TWiT in the second half that I missed on Sunday, the Second Circuit Court in New York, the top federal court, struck down the NSA's bulk phone surveillance program as illegal. Now, I didn't have a chance to follow this up. But if this was - and this was an appellate court, which means that somebody sued somebody over something, I mean, like over this.

Leo: Well, there was an earlier court decision which was appealed.

Steve: Right. But so there must have been two parties to a suit over the NSA bulk phone surveillance. And what I didn't track down was what the original complaint was. But the court...

Leo: I'm thinking ACLU, but I'll have to check.

Steve: I would imagine that's true. And the court held that a provision in the USA Patriot Act known as Section 215 cannot be legitimately interpreted to allow the systematic bulk collection of domestic calling records. On the other hand, that's all the court did. Now, that's really all an appellate court is supposed to do. But the article said that there was no injunction ordering the NSA to stop, no slap on the wrist, nothing else. All they said was the program was illegal. But that's probably appropriate at the appellate level. I

would imagine now maybe the original litigants will, now that the appellate court has its decision, will go to some next step.

Leo: Well, they do have that option. But I think what's going to happen is the government is going to wait for Congress to see, A, if they reauthorize the Patriot Act; and, B, if they write some additional language. There was an interesting article today that said this could have much - this was in the Washington Post - much wider impact because apparently laws are often written using cut-and-paste.

Steve: Of course they are. How else do you get a thousand pages that nobody reads?

Leo: So the offending line in Section 215, which was the justification, was very vague justification for bulk collection, is apparently used widely, including in laws regarding pen register warrants.

Steve: Wow. Just sort of drop it in.

Leo: And so the impact of this could affect much more. All of it we'd be happy about because it's all about privacy. It's justification for doing kind of widespread fixing expedition stuff like that, without probable cause. And good. Good.

Steve: Yeah. So next month Congress is expected to take up the question of our 14-year-old Patriot Act and decide what to do with it, to give it explicit legal rights and protections, maybe change it in some way. Most people are expecting they're just going to kind of leave it alone, just reauthorize it and just...

Leo: They have, every few years, done this for a long time. By the way, it was ACLU v. Clapper.

Steve: Oh, nice.

Leo: Et al., yeah.

Steve: Nice. Good. Well, you know, that slippery memory. Even when you're notified the day before...

Leo: I forgot. I forgot.

Steve: I just forgot.

Leo: The dog ate my homework.

Steve: Needs to scratch his head a little harder.

Leo: I brought it home, and I just - I started watching "House of Cards," and I forgot.

Steve: Yeah. It was boring. Boring politics.

Leo: Boring to read that, yeah.

Steve: So anyway, I'll just finish by saying we live in interesting times.

Leo: Okay.

Steve: And it's just fun to see how this is all coming down. Speaking of which, there is nothing to panic over yet, but this is just proof-of-concept code. It's living on GitHub right now under the name Jellyfish. It is a proof-of-concept of GPU infection malware. It functions as a rootkit and a keystroke logger. And again, the keystroke logger was just sort of - so it had something to do after they got it loaded up into the GPU.

Now, what's tricky about this is that the graphics processing unit is a master on the bus. So it has DMA access to the system's physical memory. Normal applications explicitly don't. What makes you know, we talk about so-called "userland," or application space versus the kernel. And its operating systems deliberately work to protect the kernel because down in the kernel you're god of the machine. You have access to all the peripherals, all the memory, everything else going on. So there's this deliberate - that's what these rings are, you know, Ring 3, Ring 0. They're isolation of privileges of what instructions and memory applications have access to.

Well, what's a little chilling about, well, very chilling about this is that this notion of using the GPU is a full circumvention of all of that protection because a userland program can load code in the GPU, which then from its vantage point has access to the entire system. So to say this is worrisome is probably an understatement. At this point, they've got it running. It survives warm boots. They use the OpenCL API, which in some cases needs to be installed, although the Macs, which are leading graphics rendering and technology and standards, already do have that installed by default. But now the various GPU vendors have come up with a common set of coding conventions that create some uniformity among their hardware that begins to then allow us to have one solution that runs across graphics cards.

So the keystroke logger was an interesting hack, demonstrating really only that, from up in the graphics processing unit, it was possible to look down, essentially snoop on main memory because, when you're entering keys, those keys are going into a buffer somewhere. And the GPU is able to find it and essentially do a keystroke logger. And at the moment no tools find this. Of course, this is the first one we know of. And now you can imagine we'll start having antimalware for our GPU, or maybe get some security locks or hardware signatures, I mean, who knows how we'll move in the future.

But I just sort of liked the coolness of the hack, that a user program could load code in the graphics processing unit, which due to its direct connection to the bus then gives that

code from the program that was restricted absolute global control over the system. Again, no malware exists, just proof of concept. But we know how these things go, also. Now that the hackers are all aware that this is possible, there'll be a race to actually turn our graphics processing units against us. Wonderful.

There was an interesting article picked up in - it was at ThreatPost.com, I saw it in also TheRegister.co.uk - picked up on a report that two university researchers in Germany and Portugal published in a paper exposing essentially the encryption weaknesses in what's known as the Open Smart Grid Protocol, OSGP. Now, any listener of this podcast has heard us say over and over and over, you never roll your own crypto. I mean, it wasn't well understood 20 years ago, or maybe even 10 years ago, that it was just a bad idea to, like, say, oh, I'm going to write an encryption algorithm. First of all, we already have them. I can't think of anything crazier. And it's notoriously difficult to do, such that even world-class cryptographers with decades of experience still have difficulty creating strong crypto, and weaknesses are found that surprise them.

So how can amateurs do it? Well, amateurs can't. And in this case, the group ETSI - I'm sorry, no, ETSI is the standards group. The European Telecommunications Standards Institute has standardized on this. And this was released three years ago. There are now more than four million "smart," in quotes, smart meters now deployed. And then these researchers in university discover that they made up their own message digest. They didn't use, not even SHA-1, or not even MD5, because bad as those are, or bad as MD5 is and worrisome as SHA-1 is, they're at least strong message digests. These guys just came up with their own homegrown message authentication code, a MAC for authenticating messages called OMA Digest. And the researchers looked at this OMA Digest, and it just - it crumbled so quickly under their scrutiny that then it became a game of how many different ways do we have of bypassing this.

So, for example, in one, they came up with one attack where you need to see 13 queries of this protocol to recover its 96-bit secret key. Or, if you only have four queries, then with a 2^25 time complexity, which is not bad, that's 2^25 is like 25 bits, so it'll take a while, but not burdensome. And that's if you only have four queries that you're able to observe. Then you can crack it. Or if you, I mean, so anyway the point is that this thing was just a horribly written standard. And no one can understand really why they just had to come up with something themselves.

The only argument that I could see is that there might be so little processing power in a smart meter. You know, it might just be limited to a tiny little chip that's got to be super low power, low complexity. They may not have had much computation resource to work from. But we still have good technology that can run in small chips. And there's now custom hardware that'll do these things. And there was a few years ago.

So anyway, Matt Green tweeted on the 6th of this month, a few days ago, he said: "Apparently the smart grid crypto protocols are so broken that researchers are having to invent faster attacks just to challenge themselves." And then there was an announcement from these crazies that they were going to reexamine the crypto that was used in the protocol and maybe update it. I don't know how, what they do with the four million installed meters. And again, it's the kind of thing where this is the university researchers leading an understanding of a protocol. But, boy. And this thing was created in 2012, long after it was understood that you just - you never create your own crypto from scratch. You look at, from the vast array of proven technology, crypto technology, you pick the one that best suits your needs which has been proven. And for whatever reason they didn't do that.

Again, on the theme of we're living through interesting times, there was an interesting

story about how, in the wake of this election and change of control in the U.K. last week, that the so-called "Snoopers' Charter" was set to make a return. And David Cameron, of course, famously said that his party was planning to introduce even more wide-ranging powers if he was reelected to government. He said that there should be no form of communication that the government was unable to read.

So it's going to be interesting to see what happens. The argument is that the conservatives now in power in the U.K. are already planning to introduce huge new surveillance powers, which is being known in shorthand as the "Snoopers' Charter," hoping that their increased power and the removal from government of the Liberal Democrats that were previously blocking this will now allow it to go through.

And from a standpoint of technology, which is where we always come back, I don't know what this means because, if the testimony that we talked about last week is any indication, it looks like there is strong opposition in the U.S. for allowing deep snooping of communications. I'm hopeful that that's the case. We haven't, you know, there's no decision yet. But the U.K. appears to be going in a different direction. Yet we all live on the same planet, and we all kind of have the same technology that we're sharing, you know, the Internet works for all of us. Yet what happens if the U.K. says, well, that's not going to work for us. We need to be able to decrypt all communications. Well, a lot of U.S. communications goes over to the U.K. and vice versa. So anyway, definitely interesting times.

Okay. So many people picked up on this next story and tweeted me. And I don't quite know what to make of it. This was a posting on the KoreLogic.com blog, K-O-R-E-L-O-G-I-C, which the blog posting was talking about the shelf life of data in SSDs. And it was written from the standpoint of informing law enforcement or anyone doing forensic evidence gathering, that they need to not let SSDs sit on some shelf somewhere because the actual specifications for the endurance of disconnected SSDs is startlingly short in terms of time. So again, a lot of people tweeted it to me. I dug into it and went back to the Jedec.org original document to go back to the source and see what this was about.

The only thing I can make of it is that these are worst, like very worst-case, sort of theoretical worst-case limits which any device that wants certification needs to surpass. And in the show notes I have the first table from which this blog post pulled its facts, which notes that there are two classes of SSDs, the so-called "client" classes and "enterprise" classes, or client and server classes. And they characterize a client class as operating, actively operating at about 40 degrees C for eight hours a day. So that's inside of the standard user's machine or laptop, where they're running it for eight hours, they turn it off, and turn it back on the next day. The enterprise model is hotter, at 55 degrees C, and never being shut down, 24 hours a day. So, now, here's the kick, is that the spec is saying that these SSDs in the client model, if stored at 30 degrees, will retain their data for one year.

Leo: Thirty degrees Centigrade is what?

Steve: It's not that hot, maybe, well, 80, 85 something? Because, like, 25 is typical, right, 25 is room temperature?

Leo: Okay, yeah.

Steve: So it's warmer than room temperature. But they're also saying, of the enterprise devices that have been written a lot harder at 24 hours a day, but also stored hotter, they're assuming a storage temperature of 40 degrees C, that they will only retain their data for three months. So, okay.

So now, stepping back from that a bit, what we know of this technology is that these cells are to some degree leaky, that is, they're capacitors that have electrons stranded out on a small piece of them, and that the field being generated, the electrostatic field being generated by the electrons is what can be sensed in order to read these. And we certainly know it's reasonable that, as you increase the temperature, the storage temperature, that you would increase the leakage of these charges from the individual bit cells in the SSD.

So all that makes sense. But, I mean, it completely goes against our own experience, that if you don't use an SSD for some period of time, it's just going to be dead. I mean, or maybe a client system has never been in a position where it's been off for a whole year and unused. And there's also this assumption that somehow its power-on versus power-off had an effect on this. And it wasn't clear what that was, unless it was just the ambient temperature. When it was power-on, it was at a higher temperature because it was in an operating computer.

The other thing, though, is that in the next page of the notes I show the table, which is really interesting, which shows the number of weeks of retention as a function of both the active and the power-off temperatures. And so the table has green cells lit up for their two typical cases. But what's interesting is that it seems to say that you get much better retention if you run them hotter, but then store them colder, which I thought was really interesting. So if you run it at 55 degrees and store it at 25 degrees, then this table says, over there on the far right, you get 404 weeks of retention. Whereas, if you stored it at 55 degrees, that's eight weeks of retention.

So for what it's worth, I mean, and this is material apparently submitted by Intel, and this was a slide presentation given from the official site in an official presentation. So the absolute bottom line is refrigerate them. They'll be joining - my stored SSDs will be joining my Palm Pilots in the refrigerator. I'm just teasing, of course. But seriously, if you're in a situation where you're wanting to store data archivally, I would say, first of all, a hard drive looks like a much safer place to put it for archival storage than basically what is a brick full of little capacitors that are bleeding their charge out. And whereas a hard drive is not temperature sensitive for its storage, I really do believe that there's one takeaway is that SSDs absolutely would be.

And so think about temperature. If anything you're doing is storing SSDs offline for the future, we need to think of them not as super reliable data storage, but as very temperature-sensitive data storage. Which I think was a useful takeaway from that.

Leo: Most of us keep our hard - don't, like, well, I mean, I don't have any SSDs that are in the closet. I mean, they're all in use.

Steve: Yeah.

Leo: They're all fine.

Steve: No one wants to not use them.

Leo: Yeah, you know, you're not going to use them for archival storage until they get cheaper per gigabyte, I guess.

Steve: Right, right.

Leo: And if they're in use, even if the computer is powered down, that's okay. Aren't they being trickled a little bit?

Steve: The only way I could make any sense of it mattering whether they were in use or not is if they were powered up, and the SSD itself did a periodic sweep. You know, essentially sort of did its own SpinRite.

Leo: Ah, they must be, yeah.

Steve: Well, except evidence is they don't because SpinRite fixes them.

Leo: Right.

Steve: So if they were doing their own SpinRite, then SpinRite wouldn't have anything additional to offer. And in fact several of the tweets said, "Oh, Steve, SpinRite's got a great future here." Because in fact what will happen is you'll start getting ECC errors, correctable errors, which is what SpinRite specializes in fixing by rewriting the data before it has a chance to become so bad that it is uncorrectable. So anyway, so just it was sort of interesting. But, I mean, the only way this made sense to me, like why would it care if it was unpowered versus powered, is if something about it being powered prevented the leakage from the cells. But that isn't the case. I mean, so the only...

Leo: There's some kind of maintenance routine. You know, I've got to ask Allyn Malventano, our SSD guru. He's, I'm sure, digging deep on this.

Steve: Cool, yeah.

Leo: I'll get back to you.

Steve: Okay, good. Last week I mentioned my favorite video editing tool, Video ReDo. And I found the next day an email had been sent to me from Dan Rosen, who's the founder and CTO of Video ReDo. He said, "I found your private email..."

Leo: Oh, nice.

Steve: Yeah. Well, wait till you hear because he's a fan, Leo. "I found your private email address in our database, so I hope you don't mind me using it. Along with the rest of the Video ReDo team, I wanted to say thanks for this week's shout-out. I've been a longtime fan of your show, Leo, and the TWiT network in general. It was really a thrill to hear us mentioned in such glowing terms." And then he says, "If there are any Video ReDo features or enhancements you'd like to see or even just want to discuss video editing technologies, please let me know." And for what it's worth, that thing is feature complete. There's nothing that I need that it doesn't do. And he says, "Although I haven't needed it in quite a while, SpinRite has 'saved my bacon' [he has in quotes] a number of times. So thanks for that, as well."

And then following on that, I ran across this next one in the mailbag, and I didn't want to tie up one of our Q&A questions. But a Daniel Burstiner in Uniontown, Ohio wonders about downloading from a TiVo. And he said, "Okay, Steve, you mentioned Video ReDo in context with downloading from TiVo. How do you download recorded programs from your TiVo? I've been a TiVo user for many years, and I still have not been able to do it reliably. I'm also a SpinRite owner from long ago. In fact, I just went through my collection of 3.5" floppy disks and tossed all of them except my original SpinRite disk, for sentimental reasons." So for what it's worth, to Daniel and anybody else who's interested, and even you, Leo, since I believe you're a TiVo user, as I am again...

Leo: I am, yeah.

Steve: And I have been since the Series I, the silver boxes. There is a cool open source project on SourceForge called KMTTG. I have no idea why that's what it's called, KMTTG. That is a very slick - it is Java based, but that gives it platform independence, so that was a tradeoff they made. But it's a very slick tool for downloading and processing the special TiVo format files into MPEG-4. It's got FFmpeg built in, so it can transcode it into other formats. And you're able to just, like, tag a bunch of things and say get them for me, and it does. It will not download anything tagged with, like, an HBO protected tag.

Leo: Oh, okay.

Steve: So it won't do that. For that you need to use an HDMI capture card because those have HDCP support, and so you're able to...

Leo: Yeah, that's not going to work either because of HDCP.

Steve: No, it does, I do it all the time.

Leo: Really.

Steve: Yeah.

Leo: What do you capture it to? Because usually HDCP is not enabled on anything

like a computer.

Steve: No, it's - I don't have it here in front of me. But several of the little standard cheesy capture cards from China, they all have HDCP on them.

Leo: Of course they do.

Steve: And it sucks it right down.

Leo: You got it from China. You didn't mention that, right.

Steve: Yeah. So...

[Crosstalk]

Leo: [Indiscernible] is using TiVoToGo, which is a TiVo-approved protocol, this KMTTG.

Steve: Yes, yeah.

Leo: And so that's why it won't work on proprietary content.

Steve: Correct. Correct. And I would argue that we want to play by the rules. But for anyone who wants to, like, there's something on broadcast television that you want to share, this makes it very, very easy to suck the file out and convert into standard MP4 video files.

Leo: Chatroom says, [Eric Dickman] says Kevin Moye, who's the creator, is KM, and TTG of course is TiVoToGo.

Steve: Ah, very nice.

Leo: That's how you remember it.

Steve: So, okay. We've joked and talked about Coin, the multifaceted single-use credit card replacement. And I have to say that, after looking at the video of the guy who is behind it, where he started off at the beginning of the project learning how to solder, I was originally a little concerned. And then he discovered that, when you wrapped wire around a piece of metal and energized it, a magnetic field was generated. I'm like, okay.

Leo: You're kidding.

Steve: We have a ways to go.

Leo: Amazing.

Steve: So anyway, who knows where Coin is. But there is another suitor coming along that, oh. It's called Plastc, you just leave out the "I," so P-L-A-S-T-C. And this sucker looks, once again I'm excited, looks fabulous. It's got eInk on the face of the card, and a touch surface, great slick-looking video, and it's available for preorder. So of course I have mine on the way, or preordered, and I'll let everybody know...

Leo: Did you ever get your Coin?

Steve: No, I don't think Coin actually shipped. I think they're now learning about glue at this point. So they still have...

Leo: Once they get glue down. So apparently Ron Richards from All About Android did get his last month, which is news to me, so I'll have to ask Ron if it works.

Steve: Yeah. I did see something about like a limited number of them coming out, but not in general yet. And so this...

Leo: It's risky. For instance, essentially the technology Coin uses is now owned by Samsung. It was called - what's it called? Something loop. LoopPay. Samsung bought them and put it into its Galaxy S6. It's the same thing, generates a magnetic field which the stripe reader can see, and you store it. The nice thing about Samsung is they have NOCs, and they have a secure store, and you can put your credit cards in there. And so it's just like Apple Pay or Touch to Pay.

Steve: And they really understand about glue already.

Leo: And they know about things like inductance.

Steve: Yeah.

Leo: So they have a shot. But that's the problem when you invest in something that is going to be a year off.

Steve: Yeah, good luck. We still don't have our Temperfect Mugs, by the way.

Leo: I know.

Steve: They just got four from the factory.

Leo: Whoa. Well, that's [indiscernible].

[Crosstalk]

Steve: Three of them had lost vacuum, so that didn't work. And the fourth one, that still had vacuum, was missing the special thermal block internal that transfers the heat to it. So these guys are just having real learning curve troubles. I mean, this is what happens when you just tackle something, a big manufacturing...

Leo: Manufacturing's hard. That's why Apple has this huge advantage, because they've done so much of it now, they know exactly what they need to do. Samsung, I just saw, sent something like 55,000 aluminum milling machines to China for making its Galaxy S6.

Steve: Wow.

Leo: I mean, this is not an easy thing to do.

Steve: No.

Leo: Just walk in off the street, say I want to make something.

Steve: No. You'll like this next link. You ought to bring it up: "If programming languages were vehicles." We've been talking about programming languages a lot.

Leo: Okay, yeah.

Steve: I retweeted this. This came from a good friend of mine. And I got a lot of positive feedback from my tweeting of it. So I thought I would share it with you just because it's fun. It's all the various languages. And again, if they were vehicles, what vehicle would they be?

Leo: C would be an Army Jeep, reliable in situations where your life depends upon it. C++, obviously a Hummer. Whoa, C# is C++ painted red. Java, a pickup truck, kind of sort of gets the job done, but slower, bulkier and polluting. Python is a Honda minivan. Easy to drive, versatile, not fast or sexy, but neither are your errands. Perl is a Volkswagen bus painted in psychedelic colors, the same purpose as Python, but

only bearded ex-hippies use it. LISP, I like it, is a programming language stripped to the bare essence, around forever. Using it makes you stronger, but only an athlete or a maniac can make a living. And it's one of those old penny farthing bicycles. A unicycle is Haskell, the hipster version of LISP. And PHP is a deathtrap. Use it only because you're stuck with it. This is great. Go?

Steve: Isn't that fun?

Leo: Yeah, Go's an electric car, shiny and new. COBOL, seemed like a good idea at the time, that's a steam engine, I believe.

Steve: Steam-powered contraption.

Leo: Space shuttle, MATLAB. Weather balloon, R, when they can't afford MATLAB. This is great. And very insider. I mean...

Steve: Yeah. And there's our conclusion is...

Leo: JavaScript.

Steve: ...JavaScript.

Leo: If you put big wheels and a racing stripe on a golf cart, it's still an effing golf cart. Thank you. That's great. Love it.

Steve: I thought you'd like that, yeah.

Leo: Love that kind of stuff.

Steve: So in the mailbag I also found a subject that jumped out at me, not surprisingly. The subject was "SpinRite does it again." Henry Cocozzoli in Livonia, Michigan, he said: "Hey, Steve and Leo, I had a user come to me Monday. He had been copying his data from his old machine to his new machine, when the old machine's hard drive failed. He thought he had a CrashPlan backup. He had never enabled it. First step was to enable CrashPlan on his new system."

Leo: Yes.

Steve: Yeah. "Next, I booted up SpinRite and let it run on Level 2 on the old system. When it was done, lots of green R's," which of course means it recovered sectors which were initially unreadable, "and a few red U's," which means there may have been a few

bits that it was not able to recover. But it probably got most of the rest of the 4,096 bits, which is oftentimes, like if those are in a directory, then you can access the rest of your drive, even if you've got some sector damage that we were unable to completely recover.

Anyway, he says: "Next, I booted off a thumb drive, so I wouldn't stress the disk any more. I copied the data to an external drive. Now I could work with the copy. Because the drive was encrypted with DDPE (Dell Data Protection & Encryption), I had to decrypt the data to make it readable. Thank you for the best drive recovery software in the universe. Let me know if you need testers for 6.1. Screen Savers, Security Now!, and SpinRite fan. Keep the Netcasts coming."

Leo: You got it.

Steve: So thanks, Henry.

Leo: Thank you. Isn't that nice. Hey, we've got questions and answers for Steverino. You were showing us the bitcoin - where are we today on bitcoin?

Steve: Yeah, I've just been meaning for several weeks to - we haven't revisited bitcoin for a long time.

Leo: Yikes.

Steve: But here's the last year's history in the bitcoin saga. It's just sort of sad.

Leo: I knew I should have sold my coin. But, you know, 250 bucks is still high. I mean, it's still good.

Steve: Well, compared to zero, yeah.

Leo: Well, the guy, remember the guy who paid, like, two bitcoins for a pizza? It's, you know...

Steve: The only good news is those annoying twins...

Leo: The Winklevi.

Steve: The Winklevi.

Leo: Yeah. They are big bitcoin investors.

Steve: Yeah, and they rode it right down off the cliff, or down the hill.

Leo: They did what you don't want to do as an investor. They bought at the peak.

Steve: Yeah.

Leo: Yeah.

Steve: Not the way to make money.

Leo: Unh-unh. I still have - you still have the one you generated?

Steve: I still have my 50. Yeah, it was...

Leo: You have 50.

Steve: Yeah. It generated - back then, if you scored the hash, you got 50 bitcoins.

Leo: That ain't bad.

Steve: Yeah, and it would have - cashing in when it was 1,250 bucks, that would have been good. But no.

Leo: Well, Steve. If my calculations are correct, you still have thousands of dollars. Fifty, what is that? That's $12,500. I'd cash in now, if I were you. You got that because you solved - your computer by chance solved one of the block chains, did one of the...

Steve: Actually, I got it thanks to the podcast.

Leo: Early on, yeah.

Steve: Because in order to do the bitcoin podcast, where I figured out how the crypto works, and we did an episode on bitcoin, I thought, well, what the hell, I've got a computer over here, I'll just leave it on overnight.

Leo: What the hell.

Steve: And I came out the next morning, and there was 50 bitcoins. Like, hey, what do

you know? And of course those days are long gone when some guy with, I think it was an i3, an Intel x86 i3 is like, okay, well, I'm not minting any, I'm not going to score any bitcoins anymore. And now you don't get 50. I think you get, I think it's either 25 or 12.5. It keeps dividing down how many you get for...

Leo: All carefully planned by Satoshi Nakamoto.

Steve: Yeah. I don't think that curve that we just showed was as carefully planned.

Leo: I don't think Satoshi - well, maybe, who knows.

Steve: I don't think he does.

Leo: I don't think he cared. That wasn't, you know, some have said, well, this was a Ponzi scheme, and the early bitcoin people like Satoshi were the ones who got rich. I bet you that was not it.

Steve: I just think it's wonderful. I think it's fun. It's great podcast fodder. It's interesting crypto. And, you know, just this is the time we're living through, which I just - I think it's fascinating.

Leo: Marvelous. All right. I have questions; you have answers. I believe that is the...

Steve: The way it works.

Leo: You certainly don't want me to answer them, let's put it that way. This is question one from Levi McCormick. He asked, via Twitter, @levi_mccormick, how does cert pinning prevent SSL proxy interception? Considering enabling it for my government-accessed sites. They all have SSL proxies. Well, you're going to have to explain this one.

Steve: Okay. So they have SSL proxies because the government sites need to look inside of SSL in order to check them for viruses and malware, presumably, maybe in order to do, like, content restriction stuff. And those government sites can mandate that all of the government employees have a certificate from the proxy that gives the proxy the ability to decrypt their communications. Cert pinning - and this is the direct answer to Levi's question - does not and cannot prevent SSL proxy interception, but it can detect it.

So what cert pinning is, just generically, is all certificates have a hash which is typically their - it's called their "fingerprint." And due to the way certs are generated, these fingerprints are unique. And so, for example, say that you had the original Amazon.com cert, and it'll have a fingerprint. If there's an SSL proxy between you and Amazon, then it will be synthesizing a fake Amazon.com certificate and giving that to your browser in order to make your browser happy. Basically, it's pretending to be the Amazon.com server. So your SSL connection goes to the proxy, where it's decrypted, and then the

proxy turns around and asks the real Amazon.com.

The point is, the certificate you receive is from the proxy, rather than from Amazon. And the fingerprints will be different. They cannot be the same. There's just - to make duplicate fingerprints, you would have to crack the hashing problem, where you could have arbitrary certificate contents somehow jimmied a little bit in order to generate a given digest, a given hash. And that's specifically what hashes are designed by super smart people to make computationally infeasible.

So pinning is telling your browser explicitly what the fingerprint should be. And this, of course, is the way Chrome detects illegitimate Google certs. The Chrome browser, because it's from Google, knows in advance the fingerprints of the real Google certificates. And so if a proxy attempts to spoof Chrome, all kinds of alarm bells go off. And in this case Chrome is able to talk to Google and say, hey, someone just fed me a fake Google cert. What should we do about this?

So what cert pinning is, it's a different way of trusting a certificate. Last week I was on TWiET with Father Robert, and we were talking about using DANE, which is a DNS-based alternative to hierarchical, the whole certificate chain approach. In the normal certificate chain approach, you trust the root, and then this chain of trust extends to the server that you're visiting. DANE says, if you have secure DNS, like DNSSEC, which we're sort of still waiting for full saturation deployment of, but if you can trust DNS, which DNSSEC allows, then you could ask Amazon.com for the fingerprint of its certificate, and it could provide that to you. Or the whole cert, depending. But the idea is there are other ways of obtaining the information.

So one way of trusting the cert you receive is if you trust the signers. The other is if you trust the fingerprint. So pinning is just sort of - it's an alternative way to bypass the whole CA hierarchy and say, I know that the legitimate certificate has this fingerprint. And if that's the fingerprint you receive, then you're good to go. And in fact that's the whole concept behind GRC's SSL Fingerprints page. If anyone's curious, I explain all of this there. If you just google "SSL fingerprints," I think ours is the first link that comes up because it's been around long enough, Google has found it. And it shows you legitimate fingerprints which the GRC server sees, and you can compare it to the ones you see. If they're different, then it may well be that something in your connection has changed them because I'm sure there's nothing in my connection. That is, the GRC server connection. We're right sitting in a Level 3 datacenter, directly on the Internet.

Leo: Okay. I'm glad you...

Steve: More than you ever wanted to know about certificate pinning.

Leo: No, actually, I'm glad you explained that. Makes sense. Makes perfect sense. Matt in Surrey, United Kingdom wonders how we know that the TrueCrypt binaries are trustworthy: Hi, Steve and Leo. Longtime listener and SpinRite devotee. How do we know the TrueCrypt 7.1a binaries from the TrueCrypt.org website were created exactly from the open source code that was recently audited? Couldn't the executables have been created from a different code base that could contain hidden vulnerabilities? I'm sure most users don't download the source code and create their own .exe file.

Steve: So that's your standard open source question, and it's a good one. One of the problems that TrueCrypt had, due to its lineage and the fact that it had been around for a decade, is that - and this was sort of a controversial aspect of it - is that it was difficult to build the binary. For example, in one case you needed to have something like Visual Studio 98 or something to, like, to build from the C code one of the components that the whole TrueCrypt binary required. However, a number of people did go about taking that challenge and, from the source, recreated binary identical, or bit identical binaries.

So we do know that the source that was there does generate exactly the binary that has been audited. And of course that was - you would expect that to be one of the prerequisites of the auditing project in general is we need to know that the source we're auditing is what generated the binaries that are being downloaded. And that was determined. But again, Matt, great question, and the standard question with open source because most people just grab the binary because it's a lot of work.

Leo: Well, let me tell you what you should do any time you're grabbing binaries. And in fact here's a TrueCrypt mirror of 7.1a. You see, and who knows, I don't know, this guy who did this, Ken White, I don't know if he's real or not. But here's the binary. It's a DMG, so it's preconfigured for Macintosh. But here's the sig. And what you're going to do, and people do this all the time when they download binaries from open source sites, is the signature is a hash of this binary, and you then compare it to the hash that's offered on the site to make sure that you're getting what you think you're getting, as opposed to something else. You see all these - these are all the SHA-256 hashes for various versions. So you could do an MD5. And all the open source tools that allow you to install remotely, like apt-get, will have a command that will let you do this, verify that the binary you have...

Steve: Yeah. The only problem is...

Leo: Well, you have to trust Ken White.

Steve: Well, yeah. The only problem is you're - well, you also have to trust his site because you're getting the hash from the same place as you got the binary.

Leo: Right.

Steve: So if the binary was corrupted or made malicious, and the bad guys were smart, they'd have put the hash of the malicious one so that they balanced. What I did with my TrueCrypt repository was I had somebody else who runs a security site, our Defuse.ca guy, Taylor, I had him host all of the hashes so that...

Leo: There you go.

Steve: So that I'm not offering them on the same site because actually it makes little sense to have the hash posted on the same site as where you get the binary. You want them coming from different places. And by the way, Google has decided that for some reason mine is no good at the moment. Tweets began coming in yesterday saying that

Google and Mozilla Firefox are both flagging it as malware, even though they've been there for a year, not a byte has changed, they have been run - just a few hours ago our friend Simon Zerafa ran it through his virus total, and it's zero out of 57 virus scanners found a problem with it. It hasn't changed. The hash still is the same. Google just got a burr up its wherever, up its server, and decided that it didn't like GRC.com offering the TrueCrypt binary.

Leo: Where do you - where on this? Is it on the main site?

Steve: If you just, yeah, if you just go, like, "TrueCrypt repository," I'm sure it's on the main menu there. Probably under maybe "Other"?

Leo: Yeah, there it is.

Steve: Bingo. Yup, and I put up a notice to note that Google is complaining. But look at the - we've got three quarters of a million views and lots of people downloading.

Leo: So they complain on the download? Is that it?

Steve: Yeah. When you, in Chrome or in Firefox, when you download it using one of those links, they say, oh, warning, this could be malicious.

Leo: Oh.

Steve: IE doesn't care, and as far as I know Safari doesn't care.

Leo: Oh, that's interesting. Huh. This file is malicious...

Steve: There it is, yup.

Leo: So you don't have a choice of saying okay. You just can say never mind.

Steve: Yeah. In Firefox there's a configuration. And in Chrome there's something about, you know, "hurt me badly" or something. You press a button that says, "Yeah, give it to me." Like, you know, with a negative connotation.

Leo: Wow. Oh, that's really weird.

Steve: I imagine they'll clear it up. Well, it's probably a false positive that came from - or maybe somebody reported it being mischievous. Who knows.

Leo: Huh. You can, yeah, you have to go into the settings, deeply into their settings, to turn that off.

Steve: Yeah. And people have, and then they've got it, and then they ran it against VirusTotal. They've also compared it against the version they got from me nine months ago. There's no difference. Nothing changed.

Leo: Huh.

Steve: Yeah. I know, you know, it's like, okay, fine. I'm sure it'll get fixed.

Leo: Huh. Moving right along, Question #3. Doug in Pennsylvania notes that whitelists do not work either: Steve, I'm behind on podcasts, but on Episode 495 you got a chuckle out of the picture of "why blacklisting doesn't work," with a sign banning skateboards, bicycles, rollerblades, roller skates, and a guy on a unicycle going by. But imagine a reverse sign which approves methods of transportation. What would it say? Well, it'd be impossibly long to read, and there still could be exceptions. What about people in motorized wheelchairs? What about seeing-eye dogs? What about piggy-backing or rickshaws, probably not intentionally allowable, but might get through on a general rule?

The meme probably hasn't been around long enough for someone to develop a retort picture, but I can picture a crowd of people all pointing at the signs. None of them would be allowed to pass, but they'd all be legitimate forms of travel. Of course, it's all in good humor because the sign only stops the honest people in the absence of any viable enforcement. But what is the solution? That's a good point. A whitelist might have to be extremely long to encompass all of the possible legitimate ways.

Steve: Yup. And of course, I think that the joke really doesn't reverse. So the fact that we're saying - the joke's point was why blacklists fail. And a list of clearly banned wheeled vehicles, and then the guy rolls by on his unicycle because, oh, it's got one wheel, as opposed to a bicycle that was listed as having two. But I liked it, I liked this from a philosophical standpoint because this is the trouble with whitelists. If your system is configured only to allow known software to run, then we're sort of back in that first generation of UAE, remember, with Windows, where it was just too much in your face. It was constantly coming up, and the screen going black. And quickly people are just trained to just click past it. Yeah, yeah, fine, whatever, I just want to get back to what I was doing, very much like my own employee Sue, who was given the dire warnings of pending disk failure, but it said, "Press Escape to continue." And so, oh, look, it works. And she did that until it finally did fail.

So the problem is, it's like this whole notion of blacklisting and whitelisting feels like the wrong solution to the problem. The right solution to the problem is to have systems for which there is no malware. That is to say, that are malware-proof. And I think that's probably impossible because we know that there are almost sort of like two classes of mischief that bad guys can get up to. One is an exploit in Flash that - oh, and by the way, it was a Hugo Boss ad that was spreading the CryptoLocker code.

Leo: Wow, what, really? Was it a legit Hugo Boss ad?

Steve: A legit Hugo Boss ad.

Leo: So somebody must have gotten into their stuff.

Steve: Right, and somehow managed to inject that, yeah.

Leo: Wow.

Steve: So there's that, which is not the user's fault. Well, okay, except that they had Flash that was running. But still a lot of people just use the system the way it comes. The alternative, though, even if computers were absolutely perfect, if there were no bugs and no defects, and nothing wrong could happen, you still have the user as your weakest link in the chain. And you can't cut the user out. We need users. That's what they're for. And you're going to send them email, and there's going to be a link that they can click.

I mean, and it may not take advantage of an exploit, but it could take them to a different website where, again, it says, oh, we're PayPal, and we're doing an audit of your account. Please log in so we can tell you how you're doing. And the unwitting user does. Again, we don't even need there to be software problems for there still to be computer crime. My conclusion here is, yes, whitelists don't work, or they're a pain. Blacklists are probably, obviously, a bigger problem. But even if we had perfect computers, we still have people.

Leo: True, true.

Steve: So we have plenty of podcasts ahead of us.

Leo: True, true, true. Thank goodness.

Steve: Yeah.

Leo: Yeah. I like doing the show. Here we go. Question 4, Larry Littlehale, Concord, North Carolina with a little anecdote: Recently a site I visit for alternative news stories has started doing something really annoying. I hate it when this happens. After the page opens, there is no ad in the text. If I scroll down the page, an ad opens inline with the text. It's a graphic. It spans the entire width of the text column, and it's a video, and it begins playing. It only turns on sound on mouseover, but it's so annoying to have this huge video launch. So I went into page source, found the link it was pulling from, and added it to my Restricted Security zone. This was, in this case, as.ebz.io. I can deal with banner ads by studiously ignoring them and even allow them to flash. But when the ad starts modifying the display of the

story, that's an ad too far for me. So he's doing manual adblocking. That's called a "takeover," and that is the most annoying thing. And I don't blame him. Everybody does it now because nobody sees the banners otherwise; right?

Steve: Right. And similarly, I really liked the discussion we had where we reminded everyone that it's the pulling of the ads into the browser that generates revenue for the site. And I have no problem with that at all except I guess they're getting less traction or they just - everyone wants more. And so they know that, if you put something in the middle of the screen that the person has to click away, I guess they'll get a larger, I mean, it'll come to more people's attention. And so essentially we're the victims at this end of this kind of malarkey.

As a consequence, you can imagine, Leo, of the dialogue we've been having about this, there's been a whole bunch of stuff in the mailbag that I just - I haven't, I mean, it's stuff that everyone's just sort of chiming in with their two cents. And lots of good thoughts. Some people have said, hey, why not let the browser pull the ads, but block showing them, so that we're not annoyed, yet the site still gets credit for them. And of course, if that happened extensively, then that would be a problem, too.

Leo: Yeah, yeah.

Steve: So, I mean, and the whole goal is to offer people things that they're interested in. What I have found is that, since I'm normally logged into Google when I'm doing different things, when I'm, like, on some random site somewhere, I am seeing relevant ads. I'm seeing ads about PCB fabrication. And I think, wow, that's a strange ad to have here. And then I go, oh, yeah, well, it's because Google knows it's me. So again, I do think that what we need to hope for is that the ads won't become really crazy. But if they do, I like Larry's approach, is this thing, it's bothered him over and over and over. And he just finally said, okay, no more. And so he just basically blacklisted that one site so that those ads can't come up. And it's their loss because they pushed him too far.

Leo: Yeah. And, you know, that's why I'm very sympathetic. Because if ads are really annoying, then that's not going to work for anybody.

Steve: Right.

Leo: And I think people start using adblockers when ads start jumping and dancing and singing and making noise. And I understand that. I don't blame them. I really don't. Question 5 comes from Brian Scallan of Hertfordshire in the U.K., an ARM/Intel question: In 505, the news item about the PDP-8 emulator running on the Raspberry Pi was great to hear. I hope everybody ordered theirs. If you'd consider presenting a fairly detailed compare-and-contrast between Intel and ARM RISC instruction sets and architectures, I'm sure many of us would be fascinated by your insights on this topic. Thank you, Brian. Really?

Steve: So two things.

Leo: You want to do that? Yeah?

Steve: So my mention of the PDP-8 emulator was responsible for about tripling the number of orders.

Leo: Nice.

Steve: So I've been in communication with the guy who's doing this. He's delighted with the response. And of course I'm delighted that several hundred of our listeners will be able to get one of these beautifully built machines for a couple hundred dollars. That's just - that's the bargain of the century. And to Brian and others, one of the things that I've always been trying to do is to create an archive of instructional and interesting content. Back starting in 2010, through January, February, and March, and finishing up in June because there were some interruptions, I did a series called "Let's Design a Computer," starting with Episode 233, which was "Let's Design a Computer Part 1." We were interleaving them with Q&As. And so 235 explained about "Machine Language"; 237, "Indirection: The Power of Pointers." 239 was "Stacks, Registers & Recursion"; 241, "Hardware Interrupts"; 247, what I called "The Multiverse" - multitasking, multithreading, multiprogramming, multiprocessing. And 242 was "RISCy Business." And that was the episode building on all the previous ones, where I talked about the origin of the ARM and why it's ended up doing as well as it has in the industry, why its power consumption is so low, and the nature of the architecture relative to the Intel.

So I know that there are many times listeners will say, I've been listening to you for the last three years. It's like, okay. But these were five years ago. So you probably didn't hear these. But they're still there. They're there and available. And it was a really fun series. We did also one earlier than that, "How the Internet Works." So I'll just note that, you know, these podcasts are still around. And for people who have joined the podcast since then, who might be interested in some additional content, it's all there.

Leo: Good. It's all there. RISC v. CISC and all of that.

Steve: Yeah. And we covered it in detail and in depth with, I mean, we sort of nailed it. So I won't be doing it again because it's done.

Leo: You don't need to. It'd be repetitious. And nothing's changed.

Steve: No, in fact, even the news will sound new again because nothing's changed there, either. Still having the same kind of problems.

Leo: Yup, yup. Rik in San Jose wonders about Bluetooth proximity authentication: Steve, on this week's Security Now! you and Leo were discussing the benefits of unlocking mobile devices with your thumbprint, in the context of a recent vulnerability on the Samsung platform. That's the Galaxy S5, they had a problem. I don't use thumbprint to unlock my phone. I have a Nexus 5. It doesn't support that.

However, I did recently upgrade my timepiece to a Pebble Steel, a smartwatch similar to Google Wear and the iWatch.

One nice benefit of switching to this smartwatch is that now I can have my phone automatically unlock if it detects my watch is in range. This is really useful. It has all the benefits of using the thumbprint from a convenience point of view. However, it has an advantage. I can more easily disassociate myself from my watch than I could from my thumb. Good point.

Steve: Yeah.

Leo: Solution is based on Bluetooth. The Android 5 platform apparently allows me to auto-unlock the device when the operating system detects any devices on my list of Bluetooth devices. You suggested the security of using your thumbprint is less than perfect, but it is convenient. From a security point of view, I wonder how this Bluetooth-based solution compares. I know Bluetooth devices are paired with a key and are somehow authenticated. But do you think the Bluetooth authentication is sufficiently secure? How easy would it be for a malicious device to impersonate my Bluetooth watch? I don't recall, but did you ever cover Bluetooth on Security Now!? Thank you. Rik.

Steve: Ah, Rik.

Leo: Maybe a little bit.

Steve: Episode 280 and 283. That was toward the end of 2010, that same year. We did an episode on "Bluetooth" on December 23rd, that was Episode 280. And "Bluetooth Hacking" was a couple weeks later, in the middle of January 2011, Episode 283. So once again, the content is there. I'll give you the short version, which is what we determined is, okay, there are sort of two pairing methodologies. One is for a device that cannot show you a pseudorandom string, which you then enter into the other device. Or the other device doesn't have a facility to accept a pseudorandom string.

The point is, if you have a display, and some means to enter it on the other end, you have an out-of-band information. You have information visually that is not going over the radio. So that automatically means that no one eavesdropping on the radio signals can know the information that was conveyed out of that channel, non-radio, that is, through the display. So you could securely pair devices where there was a randomly generated token that was transferred out of band in the presence of an eavesdropper of any level of power. But many Bluetooth devices, just because they're small, they're keyboardless, or they're displayless, they just don't have the ability to support any kind of an out-of-band authentication.

So there, what you are depending upon is that no one is listening at the moment that you do the pairing. And we joked about it in that podcast, back in 280, about the one vulnerability in the Bluetooth protocol, there is one, is at the instant of pairing. Once they're paired, then they have shared a secret that is never divulged, and that keeps them paired without any security risk. And so we joked about going out into the middle of an empty parking lot, where you could see all the way around you, and there's nothing

near you, and that's when you do your pairing, and then from then on you're safe. You probably thought maybe that's overkill. But for what it's worth, the bottom line is the Bluetooth protocol was well designed, and the tradeoffs are understandable and make sense. And if you want much more about it, Episode 280 will give you all the information.

Leo: Of course, that was probably Bluetooth 2 or 3 we were covering. I mean, we're up to 4 now.

Steve: Yeah. There haven't been any changes in the...

Leo: Oh, okay, the authentication's the same.

Steve: Yeah.

Leo: Oh, okay. That's interesting.

Steve: Yeah.

Leo: Yeah, I think you're also in a Faraday cage in your car. So if that's where you're doing the pairing, you're probably almost as good as being in the middle of an empty parking lot.

Steve: I would say, except for the fact that your cell phone works in your car means that you're not in a Faraday cage.

Leo: No, that's true, you're not, are you. Goes through the windows, I guess, yeah.

Steve: Yeah. Right.

Leo: Yeah, the way that this works on Lollipop on Android 5.0 is you do then have to say, okay, yes, that is a secure device. And so when I see it, it's okay to unlock.

Steve: Right.

Leo: And I think, as I remember, I still have to unlock, but then it stays unlocked. So you have to unlock the first time.

Steve: Ah, that's nice. I think that's the right...

Leo: That would be the way to do it; right?

Steve: That's the right tradeoff. Yeah, because it's like - I was going to say it's like the way I've designed SQRL, where the first time you authenticate, you have to give it your whole passphrase. From then on, until something happens, like the screen blanks, you're able just to give it a little - just the first "n" characters.

Leo: Right.

Steve: Because you don't want it to be in your way, but you still want to make sure that it's you.

Leo: Right. And I can do it by - not just by Bluetooth, but by geographic location, too. So my phone, if I unlock it at home or at work, will stay unlocked.

Steve: Ah, nice. Nice.

Leo: I think that's relatively safe.

Steve: Yeah. Again, we understand that with these conveniences are tradeoffs. So you can imagine, if it were, like, really important to spoof your location to your phone, that could be arranged. But, you know, how likely is it versus the convenience?

Leo: Right. Chris in Denver wants help understanding VPNs, HTTPS, and man-in-the-middle snooping: Steve, it's not that I didn't believe you, but I had to see for myself. So I was just as appalled and later frightened when I read about AT&T's move to charge customers $29 less for those users who would allow their surfing to be watched by AT&T. I recalled your comment about, "Heaven help us if our ISPs force us to install their certificates" so that they can be a man in the middle and decrypt all of our HTTPS traffic.

It got me to thinking, what about those of us that use a VPN service? Is implementation different enough that, even if we have to use the certificate of the ISP to have web access, a VPN would still be secure? If so, wouldn't we be safe by first establishing the VPN, which I'm theorizing bypasses the ISP's man-in-the-middle certificate, and then open up the browser and allow certificates to negotiate the way they normally would, without being intercepted? Your insight might help alleviate some of the panic behind this clearly disturbing practice, if companies think they can get away with it. Thanks in advance for the explanation. Chris in Denver, and so far safe from the AT&T ISP.

Steve: So Chris's understanding is right. And a number of people have asked about, like, how much can they get from their VPN because it does represent a means of bypassing the ISP. The way to think about it is that your ISP is the first point of contact as your traffic leaves you to get out to the Internet. The ISP is between you and the greater

Internet. And that's because it is your connection to that greater Internet. So without a VPN, your traffic must pass through the ISP. And I just - I have a feeling one of these days we're going to - something's going to change where ISPs are going to, you know, part of getting Internet service will be installing certificates from our ISPs on our machines, which is a day I dread.

But in that circumstance, or in any proxying circumstance, because that's what the ISP is doing is proxying our SSL connections, establishing a VPN, if the ISP allows it, and that's the only gotcha, is the VPN creates an impenetrable tunnel through the ISP out to the Internet, out to a point where the ISP is no longer in control. Then our traffic emerges unmolested from the termination point of the VPN. And I do think that ISPs will probably be forced to allow VPNs because a lot of corporate users have to use VPNs in order to connect to their corporate networks. It's becoming increasingly widespread that that's the way out-of-office travelers and telecommuters are operating is they VPN into their corporate network, and then use their machine that way. So an ISP that blocked VPNs just wholesale would be in trouble, I think. So the good news is the rest of us can use it just to maintain privacy.

Leo: Do VPNs use 443? Do they use the...

Steve: Technically, they can use anything they want. So, and in fact mine, the CryptoLink that I put on hold because of the concern that the government might outlaw them, was going to use everything. And, I mean, like 443, email, SMTP, ICMP, UDP. Because what I realized was you could just do everything at once, just send out a blast of packets. And then the other end would see what got through and then let you know. And you'd go, okay, I can maintain a connection this way. So there were a lot of things I was going to do. And maybe someday, after SpinRite is put to bed, and if we resolve the nightmare with encryption, I'll get back to it.

Leo: Good. Eric Sarratt in Sylva, North Carolina wonders how you handle embedded hyperlinks: Steve and Leo, longtime listener, blah, blah, blah. Love the show. I just ran into a situation which I am sure you've run into, and I was wondering how you'd solve it. I've been sent an insanely long hyperlink for a document which I must download, a parking pass that's not available anywhere else, for a lecture I'm attending. Now, given the normal, sane security advice of not ever clicking hyperlinks in emails, how would you acquire the parking pass? My thoughts are to copy the hyperlink into Notepad, then paste it into a browser. Would that remove any spoofing problems? I've become extremely wary of all email hyperlinks and attachments after I started getting emails from my "father" with spoofed hyperlinks. Hackers had stolen his email address book. So if I can't trust email from my own kin, why would I ever trust email from ANY organization? Place Leo's dramatic effects here. Da da da da. Thanks for the show. Eric.

Steve: So, okay. This brought me in mind of a couple things. We've been talking about this for a while. And one distinction that's worth making is that, in the instance that Eric cites of an insanely long hyperlink for a document which he must download because it's a parking pass for a lecture he's attending, this was an expected email. And it's one thing to receive unsolicited emails, and another thing to receive something that you're expecting. So you go to the website to register for attending this lecture, and they say we'll send you an email with a link to your parking pass. Well, you know, the chances are very good that, when that email comes in four seconds later, that that's what they did,

and that that is actually a link to the parking pass, rather than something that is somehow malicious. Again, obviously it's not impossible that it wouldn't be. But I think there is a tradeoff.

I will tell you, because Eric was asking me how I handle it, that's the way I think. I treat something that I am expecting - I just recently re-upped my subscription to MSDN because I'm actually excited about Windows 10, and I had let it lapse because I didn't need any operating systems for a year. But I rejoined. And in comes email from Microsoft, which I was expecting. So I clicked the links to activate my account and so forth.

Now, that's very different than what looks like something from somewhere, PayPal suddenly deciding that they're going to want to check my account and have me log in, something that I didn't initiate. This is very much like the great wisdom that I think originated from Brian Krebs, which was don't ever download anything you didn't go looking for. When the site says, oh, you need a new version of Flash, click here to update, no. You didn't go looking for it. It was being offered to you. Just say no to that.

And I will just finish by saying there are times when I do not touch email on my PC, but I will open it on my iPad, because I recognize the difference in security and that there's nothing on my iPad to be hurt. And if there's email where I'm curious, I'll just hold off, or I'll actually go in the next room and bring it up on my iPad. I'll open it there because that's a much safer environment than this crazy, attack-prone, target-rich environment of a Windows PC.

Leo: That's actually a good idea, yeah. Alex G. in Montreal, Quebec, Canada wins this week's "Thinking Outside the Box" award with "Why not sign HTTP?" Steve and Leo, lately you were talking a lot about HTTPS making it impossible for ISP to cache content. Instead of encrypting content that's not private, like TWiT.tv, why don't we get the server to sign the page before sending it? It would allow the browser to authenticate the data and prevent malicious modification, and it would permit caching because there's no encryption. Longtime listener. Keep up the good work. Can't wait for SQRL. You could do that with email. You could sign it with your PGP key, your public key, or you can encrypt it. But you don't have to do both.

Steve: Right. And in fact, the original protocol, well, even today's protocol does allow for this. It's called using a null cipher. That is, you know how a cipher suite is the key agreement, the cipher, and the message authentication portions, and they have them in all kinds of different mixtures. So you have, you know, AES and SHA-256 and Diffie-Hellman for key agreement where you mix and match. Well, there are cipher suites, low numbered ones, which deliberately have a null cipher. So they are signed for authentication, but not encrypted. So it's still plaintext.

Unfortunately, there have been, naturally, attacks where, if a client offered that, and the server - oh, and a man in the middle removed all the other ones, and a server also offered that, then you would downgrade the agreed-upon encryption from something strong to something with no encryption, null encryption, literally. So that's all been removed at both ends because that's not what anyone thinks they're getting. But I just - I got a kick out of Alex's note or thought, that, hey, why not not encrypt, just authenticate? And it turns out that used to be done. But, unfortunately, it also got abused, and so we had to remove that.

Leo: Hmm. We've somehow figured out the caching thing. I don't know how it works. But we will have caching on the website.

Steve: Great.

Leo: But still HTTPS. We'll still have SSL.

Steve: I actually saw a note from DigiCert saying that they've been contacted by you.

Leo: Yeah, that was nice. So I bought a wildcard cert from them a while ago for our VPN. And then I just recently bought a wildcard cert for TWiT.tv. And one of their service guys said, oh, I saw you try to sneak by and pay for a cert. So here it is for free. And I said, oh, well, thank you.

Steve: Very nice.

Leo: Because I bought it for three years. It was not an insignificant - so I consider it a donation to TWiT. It was, I don't know, a few thousand bucks, I think.

Steve: Well, yeah. They really are good people.

Leo: So they're doing the work right now, and the new site will be SSL through and through, downloads and everything.

Steve: Nice.

Leo: Yeah.

Steve: And it just feels good to be there.

Leo: Why not?

Steve: You'll have security 24/7. Nobody can intercept your communications. Our listeners will feel better about poking around. So, yeah.

Leo: I mean, the reason why not is because certificates are expensive, and I'm also having to pay the programmers to do this. But that's why not. But...

Steve: And you're a perfect case in point. Look at just the extraness that you're having

to go through to do caching in an HTTPS environment. It's not impossible, but it's also not the default. The default is, oh, just let it - who cares, you know. Just have caching.

Leo: Well, also the path for our site is frighteningly complicated because the...

Steve: You mean the way the data winds around?

Leo: Yeah, there's an API. So Drupal's running on servers at Acquia. That's the Drupal API. But that's kind of just an API, which is then called by Node.js, which is running on a completely different company's servers, Heroku. And then the caching for that is provided by a third company, Redis. And of course if you download the show from that site, you're getting it from a fourth company, Cachefly. All of which have to do HTTPS. And three out of the four, Cachefly doesn't have to, but three out of the four have to be *.TWiT.tv. They have to be within our domain. So, but we've - believe me, don't ask me how it works. But...

Steve: Well, and the fact that it can is a good sign because I really do think we're seeing a move. We're seeing the industry saying okay, we're going to encrypt all the time, everywhere.

Leo: Right. Actually, we should do SSL in the chatroom, too. You are - you could be. I'll have to figure out how to do that. Bear can do that for us because irc.twit.tv is part of the wildcard space.

Steve: Nice.

Leo: I don't know what happens if you get SSL in IRC. I don't know what that - I don't know if that's good or bad.

Steve: There might be a reserved port for secure IRC.

Leo: I'm sure there's some way to do it.

Steve: I wouldn't be surprised, yeah.

Leo: Gary Beals, San Jose, California wants to understand car keyless entry hacks. And we did a whole show on that: I was going to send you a note tonight asking for your take on the keyless entry hacks. Then I started listening to Episode 505 and saw that you plan to tackle the topic soon. Many of my neighbors are reporting on NextDoor.com that they locked their car doors, came out the next morning to ransacked glove boxes. I of course thought of you and knew you could explain how could my neighbors' car security systems be bypassed. So I was pleased to hear you're way ahead of me and are planning an episode on the topic.

In your discussion, could you please address mitigation strategies? A recent "CBS This Morning" news segment suggested storing car fobs in a shielded box, like your refrigerator. And they'll last longer, too, folks. The batteries will just - could you also indicate whether the same hack works on garage door openers? Should we store our garage door openers in the fridge when they're not in use? Thanks to you, Leo, and the gang for your great show.

Steve: So this, I just couldn't resist ending the podcast with this tease of next week's topic.

Leo: Oh, good.

Steve: Because some security researchers have tackled the question of keyless entry hacking and how bad guys are walking up to cars and opening their doors. And once you hear it, you will never be able to unhear it. It is like, I mean, it's just like, oh, my god. It is so cool and, in retrospect, so obvious, and unfortunately rather chilling. I think there's a growth market, at least until car manufacturers fix this, in little Ziploc Faraday bags because the problem is real. And it's - for as little as $17 you can buy something on eBay that makes this work.

Leo: Yeah, Nick Bilton wrote about this in The New York Times, how it's some sort of amplification thing that people do, yeah.

Steve: Yeah.

Leo: Okay. Well, I can't wait to find out about it.

Steve: It is, we're going to do full coverage on it next week, car keyless entry hacks.

Leo: Will you cover garage doors, as well?

Steve: Turns out garage doors, just to answer Gary's question, are not the same problem because they require an action from the user. The secret here is that, if you have the car fob in your pocket, and you just approach your car, it knows it's you because this thing's in your pocket, and it allows you, you know, it opens the door for you.

Leo: Oh, right. Right, I don't have to press a button, I just open the door.

Steve: So that's the weakness in the system is they took convenience too far. And in doing so, there is a glaring security hole that it turns out is trivial to exploit. Going to be a great episode.

Leo: And incidentally, not only do you open the door, but you get in the car, you push a button, and the car starts, and you drive off.

Steve: Uh-huh.

Leo: Well, of course then they would drive out of the signal, so that wouldn't help. Steve, you're great. Anybody tell you that lately? You're great.

Steve: This is what we do. This is a great network, Leo.

Leo: It's good. It's good stuff. Good stuff.

Steve: The network, yeah. And for anybody who missed Episode 2 of The Screen Savers on Saturday, it was - Kevin was back and in great form. And you guys had a great time.

Leo: It's going to be fun. Saturday, "The Quad Father." Father Robert Ballecer co-hosts The Screen Savers with me.

Steve: Yeah, are you - oh, co-hosting with you.

Leo: Yeah, I'm always going to be on that show.

Steve: Okay. I thought maybe you were out of town, so he was filling in for you.

Leo: No, no, he's co-hosting with me.

Steve: Great.

Leo: I will be - I will miss one when we go to Europe in July. I'll miss one episode. The Fourth of July episode we're not going to do for the Fourth of July. And then, but the 11th, July 11th I think we have a - I'm not sure who's going to host that for me. But no, no, no. And I'm not giving this show up ever. This is...

Steve: No. So it'll be you and Padre on Saturday.

Leo: I made that mistake once, I gave up The Screen Savers once before. And that was a mistake. Never again, my friend. Never again. Yeah. And Robert's great. We'll probably do some maker stuff because Maker Faire is in town. It'll be a lot of fun. I can't wait.

Steve: Cool.

Leo: Yeah. And a blast from the past, somebody you know well.

Steve: Oh, yeah?

Leo: Well, of course. The fun thing is getting together with all my old buddies.

Steve: Yeah.

Leo: And they're your old buddies, too, in most cases. Louderback was on Triangulation yesterday because, you know, yesterday was the 17th anniversary of the founding of TechTV, the beginning, the first episodes.

Steve: Seventeen years.

Leo: Seventeen years ago.

Steve: Wow.

Leo: And I don't look a day older, do I.

Steve: We were all kids.

Leo: Thank you, Steve. Actually, I should give you some plugs here. Wait a minute. Hold on. Slow down, Leo. GRC.com, the three-letter domain. They don't make them anymore.

Steve: Nope.

Leo: Steve's got it, stands for Gibson Research Corporation. It's where SpinRite lives, the world's best hard drive maintenance and recovery utility.

Steve: With a long life ahead of it thanks to SSD early fade.

Leo: That's right.

Steve: Yeah.

Leo: That's good news. For you. We also can find 16 Kb audio there from this show, transcriptions, handmade transcriptions by an actual human being, Elaine Farris. GRC.com. That's also where you leave questions for future Q&A episodes, GRC.com/feedback. Lots of other good stuff there.

Steve: Yup. The home of SQRL, which has had my attention now for about a year and a half. And just in the final - we're getting down to it. We made some changes recently about where some UI stuff goes and decided to move it over into the client so that it's the same, your identity management is uniform across the entire Internet, rather than leaving it up to websites because, you know, no two websites do anything the same. And so if the identity management was left to the website, you'd have to learn each website separately. So we're just, you know, the protocol looks like it's finished. And I'm, well, a couple weeks away, probably, from being able to have something to have everyone play with. I'm very excited.

Leo: Well, woohoo.

Steve: Yay.

Leo: I'm glad.

Steve: You're glad.

Leo: I'm glad. I'll bet you're glad, too.

Steve: I'm glad. Whew. I miss SpinRite.

Leo: Full quality MP3 audio plus high-def and standard-def versions of the show are also available at our site. So you don't have to watch it live. We do it every Tuesday at 1:30 p.m. Pacific, what is it, 2030 UTC, 4:30 Eastern time on Tuesdays. But if you can't watch live, on-demand versions available at TWiT.tv/sn for Security Now!, and wherever you get your shows. This show's old enough now that it's on every darn platform. No missing that. Including our fine TWiT apps, written by a variety of developers, all of whom are TWiT fans. Thank you.

Steve: That's what I use on the iPad is TWiTpad.

Leo: It's great.

Steve: It's great.

Leo: Yeah.

Steve: Yeah.

Leo: Let's see. I guess that's about it. Thank you, Steve. See you next week.

Steve: See you next - right-o, my friend.