Listener Feedback #129

Description: Steve and Leo discuss the week's major security events and discuss questions and comments from listeners of previous episodes. They 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-324.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-324-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode 324, recorded October 26, 2011: Your questions, Steve's answers, #129.

It's time for Security Now!, the show that protects you and your loved ones on the Internet. And that's why I'm dressed like the Chief of the Chicopee Police. Here he is, a real security guru - don't trust me to protect you - Mr. Steve Gibson of GRC.com, the Gibson Research Corporation. Hi, Steve.

Steve Gibson: Hello, Leo. Great to be with you again, as always, for our 324th episode of Security Now!.

Leo: Holy Camoly. Wow.

Steve: Wow, yeah.

Leo: So today is a Q&A, 129th Q&A.

Steve: Yes, it is. And I have to say I tried not to make it all about the need to change passwords because running through the mailbag, 95 percent of our listeners all wanted to weigh in on their feelings and opinions and so forth about changing passwords. So it was difficult not to do an entire Q&A about it. I did...

Leo: Refresh my memory. Why is it that they're - did we talk about that?

Steve: Yeah. I argued that - I was grumbling about how dumb it was that corporations often required their employees to periodically change their passwords for no apparent good reason, and about how - I related, as I had a couple times before, overhearing some executives in a coffee shop one morning grumbling about this and how their system wouldn't let them use any of the last four they had previously used. So whenever this was required, they would successively change their password four times in a row...

Leo: Oh, I remember that, yeah.

Steve: ...to get back to the original one. So anyway, as always, our listeners are on the ball and brought up some interesting points. One went through what some of the legal requirements are exactly, which I think is interesting. And a couple other people did bring up some interesting points. So there was that, and a couple more questions about battery care, which has been - when I mentioned it a few weeks ago...

Leo: Oh, I loved that, yeah.

Steve: Yeah, it's been a huge focus of interest. And actually when Tom and I were doing the Q&A two weeks ago, he skipped a question which I had that I thought, ooh, we'll take care of that in two weeks. And so we've got one which references a site, BatteryUniversity.com, that I sort of wanted to leave everyone with because it's the ultimate reference for this kind of battery treatment stuff. So anyway, we've got not too much news actually this week, but the Q&A episode that I think everyone is going to find interesting.

Leo: As always, plenty to talk about. All right, Steve. Let's get the news out of the way, and I've got your questions. I'm staring at them.

Steve: Cool. Well, as I said, not too much happened in the week that our listeners have been on their own, away from the podcast. Brian Krebs, our intrepid security researcher who blogs often and is really focused on security stuff, did post a very interesting list for the first time ever under the topic "Who Else Was Hit by the RSA Attackers?" We'll remember that RSA, of course, was famously breached in what they called an - the acronym just dropped, I've lost the acronym - an Advanced Persistent Threat, APT, Advanced Persistent Threat...

Leo: Oh, I don't know how you could have forgotten that.

Steve: ...where they discovered that, over the course of some length of time, bad guys were in and operating within their network. Now...

Leo: Whose network? RSA's network?

Steve: Within RSA's, yes, inside of RSA's private network.

Leo: This gets worse and worse.

Steve: Well, in following down, doing all the forensic analysis, a large network of command-and-control servers were located, more than 300 of them, the majority being in the neighborhood of Beijing, China.

Leo: Uh-huh. Is this like botnets or...

Steve: Well, these are, I mean, this attack was very sophisticated, so that some - so these are command-and-control servers. These are not attacking systems. These were systems present to interact with the malware which had infiltrated or been infiltrated into RSA.

Leo: They had 300 of them.

Steve: Yes.

Leo: That gives you some idea of the scale of this.

Steve: Yes. So once they had those, they began looking at the traffic that was heading toward them. And here's the punch line: More than 760 other organizations have or have had networks that were phoning home to the same set of command-and-control servers.

Leo: Wow.

Steve: Brian posts the list. There are 20 percent of the U.S. Fortune 100 companies on the list.

Leo: Oh, boy.

Steve: And he says, "Among the more interesting names on the list are Abbott Labs, the Alabama Supercomputer Network, Charles Schwab & Co., Cisco Systems, eBay, the European Space Agency, Facebook, Freddie Mac, Google, the General Services Administration, the Inter-American Development Bank, IBM, Intel Corp., the Internal Revenue Service (IRS), the Massachusetts Institute of Technology, Motorola Inc., Northrop Grumman, Novell, Perot Systems, PricewaterhouseCoopers LLP..."

Leo: Oh, my goodness.

Steve: "...Research in Motion (RIM) Ltd., Seagate Technology, Thomson Financial, Unisys Corp., USAA, VeriSign, VMware, Wachovia Corp., and Wells Fargo & Co."

Leo: Well, I mean, forget the rest. I mean, that's enough. They probably could put together a pretty good attack. Holy cow.

Steve: Yup. Now, there were some AV companies there, but it was presumed that they had deliberately set up boxes which were infected.

Leo: Honeypots.

Steve: Honeypots, exactly. So they were watching this. And it was by analyzing the operation of this malware that they were able to develop this list of command-and-control servers. And a distressing number and demographic of companies are connected to this, or have been. Traffic has been seen going to the vicinity largely of Beijing, China, from these companies.

Leo: Does that mean that it was a Chinese attack? Or could it be, I mean, how conclusive is it?

Steve: It really doesn't.

Leo: It doesn't.

Steve: That really doesn't. Our listeners know I'm very careful about attribution. And attribution, of course, is the big problem with Internet-based attacks of various sorts. You just don't know who's behind these things. And so it could easily be that, for whatever reason, these group of Chinese machines were infiltrated, and command-and-control servers were set up in them, much as any bot network is essentially a set of machines controlled by a third party.

Leo: In fact, that would be a prudent way to do it, and that's how traditionally hackers do do it.

Steve: Correct.

Leo: But I bet it's China.

Steve: Ah, well, yeah. And there's lots of mumbling about

cybersecurity and state-based attacks. We now believe that the U.S., for example, was involved in the development of Stuxnet, which was responsible for slowing down the nuclear enrichment program in Iran. So it does seem that the Internet is becoming a true attack platform.

Which brings me really to my second topic, which is I'm seeing increasing discussion - and this is not like a news bullet point. But just sort of I wanted to share with our listeners that I'm seeing dialogue about the notion of two Internets. It's been around for a while, the idea that we somehow can't secure the Internet that we have. And so there's a desire on various parties' parts to somehow come up with a second one.

An [FBI] spokesman was talking recently to a group about the concept of leaving our existing Internet alone, but then creating a deliberately non-anonymous Internet, which would be one of the things they were promoting, where it wouldn't be like the one we have now, where it's just this global open free network that anybody can connect to and get on and do things with anonymously. It's more or less anonymous. We talk a lot about tracking and passwords, of course, and all that, which are deliberately identifying activities. But they would be talking about, or the FBI is beginning to talk about the fact that we just can't make what we have secure. And this is becoming...

Leo: Isn't that kind of throwing the baby out with the bath? "Oh, we can't make it secure, let's just start over."

Steve: I know. A Microsoft spokesman at a recent conference was saying the same thing. He was talking about a Red and a Green Internet. He said the Red Internet would be exactly what we have now; but the Green Internet would be much more restrictive, difficult to break into, and fundamentally have technology that made it easier to track down miscreants who were doing things. So he wasn't making the mistake of saying it would be impervious because we know that's not possible.

But as I have spoken about, the underlying technology of the 'Net is such that it was never designed with security in mind. When we were starting our series that we're in now on how the Internet works, the basic underlying nuts and bolts, the idea that you can just drop any packet into the Internet anywhere with an IP address, and it's the goal of the Internet and the Internet's routers to send it to its destination, not caring where it's from, what it contains, anything about it. It just sends it. I mean, that's architecturally beautiful, but it absolutely says nothing about security, and thus we're having problems. So, I mean, it just...

Leo: I think it'd be a fun show at some point to think about how would you design such an Internet? How would you make a Green Internet? What would you do differently?

Steve: Right. Right. And again, it's not like we could do a perfect job. I mean, we can't do even bits of it perfectly. Here's RSA, arguably a super-secure security organization, who is massively penetrated over the course of it's now believed many months.

Leo: Well, see, that's kind of the reaction I have to this idea of let's have a second

Internet, is...

Steve: I know.

Leo: ...why don't we just make the first Internet secure? What is it that we're doing, what is it we would do on a second Internet? We'd do authentication. We'd require packets to be authenticated. We'd use SSL. I mean, can't we do those things on the existing infrastructure? Why do we have to have...

Steve: Well, yes. And another example is - and there is a question about this that we'll be getting to in this podcast. But even SSL, here we have SSL, and it's basically strong.

Leo: It's kind of broken, too.

Steve: Except that then we go, oh, wait a minute, there's a problem with renegotiating sessions that we wish we didn't have anymore.

Leo: And with CAs that have been compromised.

Steve: Precisely. The fundamental idea of having certificates that we trust, well, if we can't trust the people who issue them, then that doesn't work. So, yeah, problems.

Leo: They've been talking about a second Internet for a long time. For a while it was we want a fast Internet just for us. That was one thing.

Steve: Right.

Leo: And now it's a secure Internet. I think the Internet2 Coalition down at Stanford was working on this.

Steve: I was just going to say, yes, educational institutions have talked about wanting their own, like, separate platform.

Leo: There's too many unwashed masses on my Internet. Get off of my Internet. I just, I think, let's fix - I bet we could fix the one we have instead of throwing it out. That just seems to me. But it'd be a fun exercise, wouldn't it, to...

Steve: To fix - okay. If we started, like, where I was just saying, like this problem of autonomous packet routing, if we were to say that, for example, in order to avoid denial of service attacks of the bandwidth flooding kind, they are all about concentrating traffic from many senders to a single recipient. And there are many problems with blocking

that. And that is, for example, way out on the fringes before the traffic starts to concentrate, we have routers which are, without knowing it, they are passing invalid traffic.

So what that says is we have to rethink everything. We can't, like, we can't just put a better SSL, a TLS 1.3 on top of what we already have now because that wouldn't solve any denial of service problems. So the idea would be some sort of authentication mechanism before you were even able to put traffic, I mean, I was going to say connect, but a connection is a higher level abstraction of placing a packet on the Internet. So you would have to have some sort of authentication mechanism that permitted you to inject a packet into some next-generation network. Which means nothing we have done could we keep. I mean, if that was the requirement, it starting from scratch, not a piece of the hardware that we have would work in that kind of scenario.

Leo: Wow.

Steve: So anyway, yes, you're right, Leo. It's been talked about a lot. And I just saw two pieces of news this week, these two different people, one guy with the FBI, one guy with Microsoft, seriously addressing large audiences and saying maybe we need another Internet. Don't worry, we're not taking this one away, but...

Leo: And also maybe I'm paranoid, but I also feel like what they're saying is, maybe we need another Internet that we can control a little bit better.

Steve: Exactly. Well, control would be part of it somehow. I mean, who...

[Talking simultaneously]

Steve: Who provides the authentication that allows us to inject packets on the 'Net? So anyway, we've got a problem. And I just think we'll live with it. I think we're going to limp along. The investment is too great. And Leo, it's not even clear to me that such a restrictive Internet would have ever functioned. Had we started at the beginning, then could it have gone global? One of the reasons it was so successful is that it was open, is that anybody could use it, that people could look at web pages and say, oh, that's how he made that effect work. I mean, it was the organicness of it.

Leo: It's about open. I think open is very key. And isn't a VPN, in a sense, a secure network over the existing infrastructure? I mean, why don't we do that? Why create a second Internet? Even on the face of it, it sounds nutty.

Steve: Yeah, I know.

Leo: What about encryption? What about tunneling?

Steve: Yeah, and all of those work - kinda. A VPN kinda works. SSL kinda works. Tunneling kinda works. I mean, when we're finding little mistakes that we've made in our fundamental protocols, then that sort of argues that we're not capable of building

something this big and secure as we want it to be.

Leo: Remember when Microsoft said we're going to write a completely new stack for Windows? Was it Vista or 7? And you just said, well, that's crazy because we know all the flaws in the old stack. We've patched them. We've worked on them. Inevitably, you write a whole new TCP/IP stack, you're going to introduce a whole new unknown bunch of bugs.

Steve: Well, and it's the classic expression of "Those who forget history are doomed to repeat it." Remember that I made that comment because an old, old bug that had been stomped on back in the UNIX era resurfaced.

Leo: Right, came back, yes.

Steve: It came back because they, it's like, oh, yeah, well, oops, new code, same problems.

Leo: And I think that's the fallacy of, oh, well, let's start over again. We now understand it. We can do it right this time.

Steve: Now we know how to do it.

Leo: Now we got it.

Steve: Speaking of starting over, Symantec and McAfee have both found and been analyzing instances of a Stuxnet variant. I was talking about Stuxnet a few minutes ago relative to the Iranian nuclear enrichment program. This one is called DuQu, so named because it tends to put a DQ on the front of the various file components that it uses. Portions of it are identical to Stuxnet. It seems to be targeting industrial control firms. No one's quite sure what's going on with it yet, where it came from and so forth, and it does remove itself from infected systems after 36 days. So a bunch of people...

Leo: Why would it do that?

Steve: Well, probably because it doesn't want to be found.

Leo: Job done.

Steve: And, yeah, exactly. It's either going to get its job done within that window, or it's...

Leo: Well, that's what Stuxnet, that's right, yeah, yeah, Stuxnet did that, yeah.

Steve: Yeah. So anyway, a bunch of people tweeted. I just wanted to acknowledge that I'd seen it. There really isn't much more news on it yet. Maybe if it's found somewhere really important or doing something really bad or they understand more about it. But also this was expected. The Stuxnet stunned researchers by its level of sophistication and how many different parts there were of it which was necessary to get it to do its job. Which is really the reason that people were feeling, wow, you know, this feels like it's a state-sponsored piece of work. This is not script kiddies. This is not amateurs. This is serious, heavyweight people. So I thought that was interesting.

And then the other, right behind people talking about passwords in the mailbag was people telling me about the Spanning Tree Protocol, which I had implemented once upon a time, so I do know about it. This relates to the question, I think it was two weeks ago, where a listener was experiencing a switch that was being brought down to its knees because essentially a cable was plugged back into the switch, which created a broadcast storm. And I made the comment that I wasn't aware of anyone who specified whether switches did that or not, that is, prevented it.

Well, of course Spanning Tree Protocol is exactly for that purpose, except that what I meant to say was that, as far as I know, none of these little $10 blue box switches, which is the level of technology I was referring to, implement anything as sophisticated as Spanning Tree Protocol. Spanning Tree Protocol, STP, was developed by a well-known researcher at DEC who is now at Sun. And it is specifically to allow complex topology Ethernet networks, which can be interconnected, not just in a simple tree, where by definition of a tree you would never have more than one link between any two nodes, which prevents the kind of network loops and broadcast storms that I was explaining in answering that question. And it uses its own protocol, a very sophisticated network probing protocol, to find best paths through the network and to even support the concept of deliberately establishing multiple links between switches. Or, for example, you might build an Ethernet topology deliberately in a ring so that, if a link went down between a couple switches, there would be deliberately an alternate path the other way around the ring to get to those switches.

Anyway, so that's what Spanning Tree Protocol is. I do understand about it. But that's sort of at a level way beyond the brain-dead switch which I was describing as I was answering the Q&A. But I did want to let our listeners know that I saw their feedback and thank them for it.

Leo: Good.

Steve: And speaking of feedback, I found a - actually this was a little bit older. This was a LockerGnome blog posting from Marc Erickson, who apparently posted to the LockerGnome blog. He said, "Another SpinRite success story." And he said, "I've been using SpinRite for about six months now. The first time I tried it was on a customer's computer that wouldn't boot into Windows XP. I thought, well, I'll try this software first; and, if it works, I'll save the customer the cost of backing up their data, reinstalling Windows and all their other software. And it did.

"Then, earlier this week, it happened to me. I got the BSOD [the famous Blue Screen of Death] inaccessible boot device [ugh, that's one I hate] error."

Leo: That's not good.

Steve: No. And he said, "When I ran the Windows CD to do a repair/install, it did not see any Windows installation present at all. I ran SpinRite in Mode 2 - data recovery - and, after it finished, rebooted. Voila. Windows was back. By the way, Steve Gibson and/or GRC.com haven't paid me anything for this blog."

Leo: I am an actor with a paid testimonial, yeah.

Steve: "I'm just happy to promote a product that has saved me time and money and may help someone else also." So he was just spreading the word. And so, Marc, if by any chance you hear this, thank you for letting people know. I appreciate it.

Leo: Awesome, awesome. Steve, we have - I have - I have in my hands, in the words of Joe McCarthy, 12 great questions or something like that. All right, Steve. I'm set up here. I've got my questions. I'm ready to puzzle you with a few of them. Are you ready to answer them?

Steve: Absolutely. Great feedback from our listeners.

Leo: We love our listeners. I tell you, every - I love coming in to work. We're putting in the new bricks and all the great messages. People come to visit from all over. We've got visitors from Oklahoma City in the studio today. It's just really - it's like you're working with buddies every day. Not just you and all the other hosts, but our audience. Anyway, here we go, 12 questions good and true, as they say.

James Russell starts us off from Australia. He says: How do programs know when the correct password was entered? Well, that's an interesting question.

Steve: Yeah.

Leo: How do utilities like TrueCrypt, Disk Utility, 7-Zip, know when the user has entered the correct password? The decryption algorithm will run regardless of whether the password is correct; won't it? I assume it will just produce pseudorandom noise for incorrect passwords. Or do these programs append some kind of identifying token to the data before encrypting it, check for the identifier after decrypting it? Or would that weaken security somehow? Thanks, Steve. Love the show. James, this is a fundamental question. How do it work, Steve?

Steve: Yeah, it is, it's a good question because, I mean, if the security program were to perform any kind of a test against a password, like is it right or not, checking it against the right password, well, I mean, that would be crazy because it would be super simple to simply reverse-engineer the operation of the code, step through where it's checking the password to see if what you entered matches what it's expecting, and then proceed if it were to match? You couldn't work that way.

Leo: Right.

Steve: So it is as James suggests. The system runs the decryption algorithm under - and that is to say, using the key that the user has provided. And generally what will happen is there'll be a header that the system has put onto the front of a file which is of a known format. It'll be a known layout. They may have, like, a version number of the protocol, or the encryption system may have some identifying information about the program and so forth. So that stuff it knows the proper format for. And so only the use of the proper key would decrypt the encrypted header into the proper format that the program would then look at.

So it looking to see whether the decryption was performed correctly in no way weakens the system because it doesn't matter if someone were to reverse-engineer that and see that, oh, look, it's checking to see whether this random noise equals the name of the security program, for example. I mean, it's like, okay, that still tells the attacker nothing about what key needs to be used in order to decrypt what's encrypted into something that it can see the program is looking for. So, yes, essentially it just uses whatever it's given. And then it's not going to give you the file and say, here you go. It's going to tell you that, whoops, that's not the right key. But the way it tells you is by checking something whose format it knows, which was part of that encryption, and verify that that part was properly decrypted. So, great question.

Leo: Yeah. I mean, yeah.

Steve: That's the way you'd do it.

Leo: Yeah. Francois Lagrange in Brussels, Belgium - Bruxelles - offers some reasons to force periodic password changes. Oh, here we go.

Steve: Here we go.

Leo: Here we go. Get ready. Steve, I am a security architect in a large bank, and I work daily with people managing security devices. So this is the guy who makes you change it. While mandating to change user passwords is a needless annoyance indeed, there definitely are reasons to do so in some cases. Passwords for security equipment must be changed regularly, say every six months, in the enterprise. Think about it: engineers sharing passwords, writing them in notepad files, on Post-its. I've seen passwords of equipment written on a piece of paper - I have, too - next to the equipment itself. Of course. That's the Post-it note; right? Shared to colleagues by telling them almost out loud? Right. Or even think about the many engineers that occasionally have to look up passwords while the project manager is standing right behind them.

I know, I know, there are ways around all this. Still, changing passwords from time to time just ensures the list of those knowing them is just reset from time to time to those that really need to know. It makes sense. People leave, stuff like that. Thank you for everything.

Steve: Yes, I completely agree. So that's different than the model that I was discussing, where you have a one-to-one relationship with an entity where you're not disclosing your password for any other purpose, and it's within your self-interest to protect the password yourself. What Francois suggests is certainly a valid and different scenario. But I like what he said at the end there best. He said, because changing the passwords from time to time resets the list of people who know it.

Leo: Right.

Steve: And so this is definitely a valid use case, where you sort of have more of a community of people who collectively know, who share the knowledge of a password. And it's like, if they're shouting it from office to office or writing it down and handing it to a friend because, oh, he needs to be able to have access to it, that's a different model. And so there I absolutely agree that it would clearly make sense to change the password just to - I guess I would call it "diffusion." There would just be sort of diffusion of the knowledge of the password over time. It would just sort of diffuse into the environment. And so definitely change it. And exactly as the way he phrases it, it resets the list of those who really do need to know it. And so, yes, I thought that was a really good point.

Leo: Fair enough. Question 3, Zach in Madison, Wisconsin. He shares his discovery of a slick Kindle utility: Steve, not related to the show, but I know you own several Kindles, so I thought I'd pass...

Steve: As does Leo.

Leo: As do I. I think I own them all. I thought I'd pass long this tool I just found that has greatly improved my PDF reading experience. Well, I'm reading a PDF right now. Let me - help me. Help me. Normally I shy away from reading PDFs on the Kindle since the text is so small and zooming in or rotating the screen is just annoying to me. So there's this program, K2pdfopt. It's a small EXE from Willus.com. You simply drag a PDF onto it, and it will optimize for Kindle reading. Oh.

Steve: That's nice.

Leo: Yeah. Though you can tweak many of the settings, I find the defaults work great. Take a look; share it with your listeners if you like. It is, and I like this, although he did say it's an EXE, it is Windows, Mac, and Linux, or versions exist for all of them.

Steve: Yes. He did talk about an EXE. I went to the site and looked at it. And the problem with the PDF that we've talked about in the past - well, there have been many problems from a security standpoint. But, from a reading standpoint, PDFs are inherently page-based. You're formatting a page to be a certain size. And if you put it on your Kindle, well, it's going to squeeze it down to be the size of the screen. What this utility does is it breaks that. In your benefit, that is. It says, wait a minute.

Leo: Hold on. Hold on.

Steve: Let's reformat this. We're going to make this content readable, figure out about columns, figure out about sizes. And just, I mean, it completely re-renders the PDF in a fashion that allows you to get the content out of it at the expense of the page layout. But there are many instances where that's a good thing. That's something that you want. And I did verify that he's got it for Windows, Mac, and Linux.

Leo: I'm downloading the Mac version right now.

Steve: Yeah. And you can see on that website - it's www.willus.com. And even on that home page he does refer to this K2pdfopt program. And you can see some before and after pictures of here's what the PDF page looked like first; here's what I did to it. And it clearly makes it much more legible for small screen readers. It even shows it for a smart phone, so like a really small screen.

Leo: Really cool. What a great idea, yeah.

Steve: Yeah. So thank you, Zach.

Leo: Yeah. Thank you, Zach. Moving along - and I'm downloading it right now. And then what you do is you just email it to your Kindle. You have the PDF, and then you email it. And I think they charge you, what, a dime for that. But that's the problem, is that. Or I guess you could hook it up via USB. I have never done that with any Kindle.

Steve: I have, actually. It works great.

Leo: Yeah, and saves you money.

Steve: There's a folder called "Documents," and you just drop all of your files there. And then when you disconnect the Kindle it does an inventory and discovers new things there and adds them to its home page.

Leo: I shouldn't be so lazy. I just mail it. An anonymous listener wanted to weigh in on lithium-ion battery care. You talked about that a couple of weeks ago. He says: Steve, you mentioned that lithium-ion batteries want to stay charged to get the maximum lifetime. So you see, when I get to work I plug in my phone.

Steve: Yay.

Leo: So that it's, even though I may not need it, it's always charging; right? According to BatteryUniversity.com, which is cited on the Wikipedia page for lithium-ion, that's not completely accurate. According to Battery University, if you continuously keep lithium-ion batteries fully charged at 100 percent permanent capacity, the loss is greater than if you keep the lithium-ion battery at half charge. Oh, yeah. But how are you going to do that?

Take a look at "Table 3: Permanent capacity loss of lithium-ion as a function of temperature and charge level." If you keep the battery 100 percent charged all the time at 25 degrees Celsius, you'll lose 20 percent of the capacity in about a year. However, if you let the battery discharge and leave it at about 40 percent most of the time, you should only lose 4 percent capacity. It is correct that you shouldn't run lithium-ion batteries down to zero; but also you shouldn't repeatedly keep charging it, which would keep the battery near 100 percent and then of course reduce the lifespan of your battery.

Steve: Well, our listener is close to correct. And this is sort of the last bit of lithium-ion battery-handling care that I wanted to share with our listeners. Where he's wrong is that this is specifically and only true for storage. So, Leo, for example, all of those old Palm Pilots...

Leo: You should discharge them to half and then put it in the fridge.

Steve: Yes, exactly.

Leo: I get it.

Steve: So it is storage, it's long-term storage of lithium-ion batteries that you do not want to store them fully charged. They don't like that, either. But in your normal daily cycling, as long as you're taking them down and bringing them up, taking them down and bringing them up, like we use phones and other devices and laptops, you do want to, as I originally said at the start of this whole dialogue, plugging it in and keeping it topped off is the right thing to do.

I was just at Starbucks yesterday morning for a couple hours, reading. And there was somebody there with his laptop, sitting there working. And after some length of time he suddenly, in a little bit of a hurry, went rummaging around in his backpack to grab his charger because clearly he'd received a warning on his screen, saying, oh, your battery is almost depleted. Turn me off or plug me in. Well, my point was that he should have always been plugged in. If you're in a situation where you can be plugged in, then you absolutely always should be.

Leo: If you can plug in, do plug in.

Steve: Right, exactly.

Leo: That's the takeaway.

Steve: Exactly. And Leo, we just had a power failure here.

Leo: I heard the [mimicking sound].

Steve: And you and I cruised right through it thanks to my UPSes. Nothing went down.

Leo: Sweet. You're kidding. So it was just a brief little flick.

Steve: Yep. It was off for about 15 seconds while I was talking about batteries.

Leo: I heard a little buzz.

Steve: I looked around. Lights went dim and screens are off. But the critical systems, which is to say me talking to you, just went through without batting an eye. That's routers and switches and all kinds of stuff here, just working...

Leo: What are you running? You're running obviously a pretty hefty UPS.

Steve: Yeah. I've got a couple big old APC monsters, those things that take several people to move, yeah.

Leo: Yeah, love those.

Steve: They're old 19-inch rack-mount deals.

Leo: And you have enough for all the wattage that you use?

Steve: Yup.

Leo: For how long?

Steve: Well, for 15 seconds.

Leo: Uh-oh, wait a minute. Wait a minute. The blinking lights have gone off. Your PDP-8s shut down.

Steve: Oh, yup, they just did.

Leo: You don't keep them on a UPS.

Steve: I did deliberately decide what needs to survive and what doesn't, so...

Leo: Right. No point in keeping those running. Well, hey, that was cool. That's really cool. Moving along to Question 5, Tom Minnick of Baltimore, also in Maryland. What? Oh, also MD. I'm not sure what that means. Wonders about lithium - I think, oh, wait a minute, I got it. Baltimore, Maryland also wonders - the words got swapped - about lithium-ion versus lithium-poly and the effects of deep cycling:

Steve, I really love the podcast. In your Q&A you had mentioned it's bad to deep cycle lithium-ion batteries. Are Li-Ion batteries and Lithium-poly batteries the same? The reason I ask, I've been playing around with RC aircraft for a time, and they all now use lithium-polymer batteries, which by the nature of how we run these planes are being deep cycled. They just run it till it crashes, I guess. There are safety mechanisms in the speed controllers to prevent the batteries from being drawn down past a certain threshold, but these devices don't meet your keep-plugged-in-and-topped-off model. I haven't had a chance to look into this. I thought you'd be interested. Regards, Tom.

Steve: Okay. So, yes, lithium-ion, lithium-polymer are identical in the way they're handled. And Tom...

Leo: What is the difference between the two? Is there different materials involved?

Steve: Well, it's the same fundamental chemistry. One of the nicest things about the polymer technology is that they can be made in what's called "prismatic" shapes. And that, for example, is specifically what allows our iPads and thin tablets and so forth, where you're able to make a battery now, no longer does it need to be cylindrical or of limited flexibility. You can really create wild shapes using this technology. But basically it's the same lithium chemistry throughout.

So Tom's situation is a little bit different. There he is really seriously using the power in the battery. He's pulling a lot of amperage out of it over a short period of time. And there he's going to run into cycle life limitations where he's just - as we know, the battery technology is not perfect. When you discharge it and recharge it, not everything comes back exactly the way it was. So it's a little bit like friction inside the battery. There's fundamentally some wear and tear with the technology.

So my comment about keeping batteries topped off was aimed mostly at our consumer devices, like Kindles, and Apple is now going this direction in general, not having user-serviceable batteries. It really wouldn't even come up as an issue if you could easily take the battery out of your Kindle or out of your iPad and put in a new one.

Leo: You can't take it out of the Kindle anymore, can you. That's right. They used to

have them.

Steve: Only the, yes, only the very first generation Kindle you could open the back and exchange batteries.

Leo: In fact, when I bought it, I bought two batteries because I thought, well, maybe I'll need another battery.

Steve: Yup. I'm the same way you are, Leo. But so my point was that, as I'm seeing people, and it was a drained Kindle that really brought this to mind, it's like, ooh, you know, you can't change the battery in that. So, boy, keeping it plugged in is the best solution. So Tom's usage case is very different because he's probably going through batteries relatively quickly, just using them up, essentially burning them out, because they are just - they're giving all of their juice virtually right down near zero, and then he's charging them all the way up again and draining them all back down. Very different than a Kindle, which just sort of sips at the vapors of the battery.

Leo: Interesting. So he'd actually be a good test case for this. If he does go through batteries, that would be why.

Steve: Oh, yeah. And the hobby battery pack technology is very cool. You plug the battery in, and it's got a fancy connector which taps every cell in the battery and individually manages the charge per cell in order to keep them balanced. And, yeah, they've done a lot in order to get maximum amount of juice you can out of these batteries of cells.

Leo: Well, enough about batteries. Let's get back to passwords. Question 6 is Geoff Forsyth in Ipswich, U.K., who admonishes us not to blame the IT department: Steve, don't blame us for forcing password changes. We have an obligation to do it as part of credit card compliance rules. Blame Visa. Every business that takes credit card payments must comply with the security rules enforced by Visa, MasterCard, and Amex. These rules are designed to reduce card fraud, but they do drive all IT departments absolutely nuts.

Visa, MasterCard, and Amex formed the Payment Card Industry (PCI) Security Council a few years back (PCISecurityStandards.org) and will fine an organization up to half a million dollars if it doesn't meet their rules - the rules are called the Data Security Standard - and then subsequently suffer a breach. The whole world seems to be struggling to implement these IT rules. And though in general they are set up with the best intent, they do insist password changes are implemented for all staff every 90 days. So the DSS says change user passwords every 90 days, require a minimum password length of at least seven characters...

Steve: [Buzzer sound]

Leo: How about at least 10? Use passwords containing both numeric and alpha. At least they said "at least," right?

Steve: Yes.

Leo: Not some between seven and 10 or something like that. Use passwords containing both numeric and alphabetic characters. Do not allow an individual to submit a new password that is the same as any of the last - oh, that's where this comes from - as any of the last four passwords he or she has used. Limit repeated access attempts by locking out the userID after not more than six attempts. Set the lockout duration to a minimum of 30 minutes or until administrator enables the userID. I really hate it when they do that, when they make a call.

Steve: But it is safer. I mean, it's...

Leo: Well, it works. If a session has been idle for more than 15 - this is to prevent brute force, obviously. If a session has been idle for more than 15 minutes, require the user to reauthenticate to reactive the terminal or session. So this is the kind of thing that - now, see, as a bank customer, though, I don't have to change my password. So this is for staff, not for customers.

Steve: Correct. And so, for example, my own organization is abiding by these rules just because I originally implemented my one-time password system. So we're never - every password we use is different from the prior password.

Leo: Do you make them change every 90 days, though?

Steve: There's nothing to change. Every time you log in.

Leo: Oh, it's one-time, they're one-time passwords; that's right.

Steve: Yeah, yeah. So we're - and I do have something like a lockout and so forth. I mean, all of that is sort of - it's what somebody who was conscious of security would build from the beginning. And what I do wish, the one thing I wish that was pervasive was locking users out after some number of guesses. It's just so often the case that the companies don't want the hassle of users saying, well, I tried six times because I have 12 passwords I use all the time, and I couldn't remember which one of those 12 it was. And now I'm locked out. They just, from a customer support standpoint, no one wants to get that call. So, but boy, that would just make such an improvement for security. But at the cost of people being hassled, exactly as you said when you read that rule, Leo. I mean, you're right, it does bite you sometimes.

Leo: Oh, it drives me crazy. Keith in St. Louis, Missouri brings up a very good point

about password changing: Steve, listen to you and Leo all the time. Love it. Feedback regarding the password changing policy topic that keeps being brought up - I love it, this guy is saving us time by eliminating unnecessary or redundant words. Sure this has been thought of, but you keep saying you can't find a good reason as to why users need to change their password every X months.

Let's say I attack a user and get the user's password. I've got the clear text password; right? I want to keep my access, but I'm not going to do anything to let the user know I've compromised his or her account. If the user is forced to change his or her password every so number of months, I'm screwed. Of course, if I had the rights I could create another account, put in a backdoor. But if those are found, I could go back to the user's account unless they're forced to change it. So isn't that a good reason why we need to force users to change passwords every so many months?

I think it would make sense. Yahoo mail, AOL mail, where users don't do a good job with their passwords and probably do get compromised, at least if you encourage them to - this is Leo adlibbing here. At least if you encourage them to change it on a regular basis, that'll protect them against having been hacked.

Steve: Yeah. This, to me, I mean, I see Keith's point.

Leo: It's a little paternalistic.

Steve: Well, and to me it feels a little bit like the user ought to have responsibility for recognizing whether their password might have been compromised. For example, if you give it to someone because you need them to log in for you or something, and then, sure, you could say, oh, and please never use it again. But the smart user would say, okay, now I'm going to change it because I was forced to give it away, and I want to restore my privacy.

So, I mean, I certainly understand what Keith is saying, that, yeah, if a bad guy got your password and was loving on a persistent basis having access to your account, then, clearly, changing it periodically would shake those people loose. But to me that feels like a little bit of a stretch, relative to the hassle that users would be put through changing all of their passwords every X months for no good reason. If you've got a really strong password, and you're responsible with it, my sense is, eh, keep it.

Leo: Good, because that's what I plan to do.

Steve: Yeah.

Leo: Chris Strzelczyk in Michigan. Strzelczyk.

Steve: You did a good job with that, Leo.

Leo: That's a good try, anyway. Strzelczyk in Michigan reminds us about - for all we know, he pronounces it Sade, I don't know - reminds us about insecure fingerprints: Steve, I'm a listener in the programming and server administration profession for many years. My brother is somewhat of a clean freak and tends to wipe his iPad screen often. Hmm. After leaving his iPad at my house one day, I observed the fingerprints left on the screen could be mapped to his unlock code. Since they were pretty much the only fingerprints on the screen, it was trivial to figure out the actual numbers. Now I just had to write them down and decode.

Since my brother is not a Security Now! listener, I suspected his passcode would be something he could easily remember, probably some number that binds to some personal information. I was right. A couple of birthday combinations later, I was inside his iPad. Oh, yeah, that's right, because it doesn't tell you what order. It just tells you what he tapped. Now, on the swipe, like on the Android phones they have a swipe, then you could even see the order.

I wanted to bring this to users' attention. When you have a code on your iPad, you should, well, maybe not wipe it as often so that there's lots of fingerprints, or have a code that doesn't bind to any personal information. Thanks for a great show. Yeah, Dvorak, every time he gets my phone, tries to read the fingerprints to figure out what the unlock code is.

Steve: Ah, John.

Leo: He knows.

Steve: So I've noticed the same thing. I can't remember what it was, but when the iPad, when my iPad was new, and probably before I added the antiglare film because that really does cut down on the fingerprinting problem, there was something I was playing, a puzzle I was solving or something, where it was inherently a grid. And I remember looking at the screen after a few days, I mean, and you could really see the grid that I had been touching.

But I liked Chris's point because this has reminded me to say to our listeners something that I have been intending to for weeks. And that is, I've been very happy with changing my iPad to the setting that gives you the full keyboard, not just the number pad. And it's a beautiful change because you can still use just a very few keys. But because you're dealing now with a much larger alphabet, with more than 26 things, including special characters, you can just go tap tap tap and enter. And even in a situation where you weren't clearing your screen, there's, well - let's see. I guess that is not the case. You'd have more locations if you...

Leo: At issue is the order. So the more presses you have, the more difficult it is to figure out what the order is.

Steve: Correct. So you could certainly use - you could use more than the limited four numbers. But my sense is, if you were using the keyboard, those taps would be obscured among all the other uses of the keyboard and other things. It's over much - you're spread out over a much larger surface area, and much less easy to see what it was you

were typing.

So anyway, I think I've mentioned this, or someone mentioned it once before, and I ended up following up on it, that is, switching from the 10-key pad to the keyboard. It is no additional work and substantially greater security because the number of combinations possible on a keyboard, and I don't remember now how many keys there are. It's certainly A through Z, but also a number of special characters. And of course you have Shift, as well, if you wanted to do that. So you could add a couple shifts. Oh, and that would - shifting the numbers would completely obscure the sequence because no one could tell when you had pressed the Shift key or not. So that would be good, too.

But anyway, a simple change to make to your iPhone or your iPad that really does jump the security up. And the thing that I realized was that I have persistent logons and sessions and remember-me settings on my iPad, just for convenience. And it occurred to me, well, what if someone just picked up my iPad, I mean, or if it walked off by itself at Starbucks when I wasn't looking. What would that really mean? I mean, ask yourself the question. If somebody actually had it, what would that mean? And I thought, oh, that would not be good.

So I enabled the option of wiping it if multiple attempts were made that were unsuccessful, wipe the memory, that's now turned on, as is that screen which proposes very little usage overhead and always reminds me that nobody else can get into it. And there was a breach, by the way. I didn't pay attention to it, but it involved a way of bypassing that with the iPad cover. Did you see that, Leo?

Leo: Yeah, it was pretty funny. What it does is it will - it's the Smart Cover. I don't have an iPad in front of me, so I can't show you. But so what you do is you lock your iPad, which has a Smart Cover on it. And you can trick it, by using the Smart Cover, to put it to sleep; and then opening it up again, it will go to whatever program is open. Now, if you're in your desktop, they won't be able to run any program. They'd just be able to see it was the desktop. However, if you were in email, or you were on your browser, they have full access to that one app. So whatever app was running when you locked it, that's what they'd have access to.

Steve: Interesting. And if you try to change apps?

Leo: You can't.

Steve: Oh, okay, interesting.

Leo: You can't. So you only can - so it's kind of like it's still somewhat secure, but not totally secure.

Steve: So obviously Apple will fix it here in the next update and so forth.

Leo: And by the way, you don't have to have a Smart Cover, as Zeph88 is pointing out in our chatroom.

Steve: Use a magnet?

Leo: Any magnet...

Steve: Yeah.

Leo: ...will do it. Or if you're a bad guy, just carry a Smart Cover in your back pocket.

Steve: Speaking of which, I did tweet something this week that'll still be up near the top of my queue for any iPad 1 owners who are missing their multitasking gestures, which were removed even in developer mode when we updated to iOS 5. Many people who have the original iPad 1 did use the developer mode in order to enable multitasking gestures. It's very nice. I like you can squeeze it in order to go back to home. You can lift in order to get to your - which is the equivalent of double-tapping the Home button - and also then just swipe sideways in order to change apps. Someone developed a non-jailbreak way of reenabling multitasking gestures for the iPad 1.

So go to Twitter.com/SGgrc, and right near the top of my feed you will find that. And I tweeted a link, and a whole bunch of people sent back responses saying, yay, this is fantastic, because they were really happy to have it. So you don't have to jailbreak your iPad to do it. It's something that the guy figured out how to do to get those back for original iPad owners.

Leo: Yay.

Steve: Yeah.

Leo: Yay. Fabio...

Steve: Now back to our regularly scheduled program.

Leo: Fabio Esquivel, whose name I adore, in Costa Rica, he's in Cartago, Costa Rica, asks: Should I be scared of TLS 1.0? After listening to your podcast about the SSL v3.0/TLS v1.0 and TLS v1.1 vulnerabilities, I ran to disable Windows support of these protocols. It's in Control Panel > Internet Options > Advanced > Security Items. I only left TLS v1.2 checked and so felt safe. Days later my iTunes 10.5 started to fail. I could no longer log into my iTunes Store using an Apple ID, so I couldn't download updates for my iPhone.

I didn't think disabling TLS 1.0 and 1.1 would affect iTunes, so I ran some diagnostics, and there I could see some connectivity issues. The detailed diagnostics showed that iTunes could no longer establish a secure connection to the App Store. So I guessed I was wrong. I went again to Internet Options, enabled TLS v1.1, restarted iTunes, still no luck; then enabled TLS v1.0, restarted iTunes, and it

worked. As app downloading worked just fine on the iPhone, I guess iOS 5 is using TLS v1.0 internally.

Should I be worried about 1.0 vulnerabilities? Should someone cry at Apple's doors to oblige them to update TLS to at least v1.2? What's going on? Did you look into this? It's interesting.

Steve: Yes. Yeah. So on Windows, which is what he's talking about, there is a scrolling option box. You can scroll all the way down near the bottom, and there it lists SSL v2.0, SSL v3.0, TLS versions 1.0, 1.1, and 1.2. Now, even my brand new Windows 7 has, of those five options, only SSL v3.0 and 1.0 are enabled. So it is supporting the other ones, but they're disabled.

Now, what would be interesting would be to see whether enabling 1.1 and 1.2 would cause problems. That is, in theory, as we've discussed the way SSL functions - and of course that applies to TLS, same thing - the client offers a suite of supported security ciphers, literally a suite, a cipher suite to the server. And the server should have a prioritized list of from, like, the most secure that it knows down to the least secure. So it compares the client's list of options to its list and selects the most secure.

So what would be nice, and what I think should happen, is if you enabled 1.1 and 1.2, then those servers you were connecting to that were aware of 1.1 and 1.2 would preferentially choose those, and you'd have the most secure connection possible. But if you were connecting to a server that didn't know about 1.1 and 1.2, it would choose 1.0 as the best that it knows. Now, that's the way things should work. And as far as I know, they do. When this was all happening, that is, when - this is relative to the BEAST and this vulnerability that was found in the block cipher, the so-called CBC (Cipher Block Chaining) that is used preferentially by SSL and TLS. That was where the problem was.

So, first of all, we should all just step back and take a breath. I'm really glad this research was done. And it's putting pressure on organizations like Apple to increase the level of protocol support at their servers. So I don't think we need to cry to them. I imagine, I mean, you could complain; but I imagine they understand, and they're probably in their own time moving in that direction.

But this is also a theoretical attack, I mean, over the course of some period of time, by breaching the Same Origin Policy in the browser using a Java Virtual Machine hack, which has since been patched. And you would still need to be able to breach Same Origin Policy in order to pull this off. Then it was possible for them to, through a lot of number crunching, to successively decrypt bytes of a cookie, which gave them access to an existing session. So, I mean, this is - it can be done. I imagine maybe we'll see at some point some packaged schemes for doing it. People are scrambling around, trying to lock down browsers and prevent this from happening. But it's the kind of problem that's ultimately going to be fixed.

However, that said, there actually is something that server administrators could do, if they were concerned. And that is, the older, non-cipher block chaining cipher, meaning RC4, is absolutely strong and useful and bulletproof and doesn't succumb to this problem at all. And so it is possible on the server side for administrators to simply disable the Cipher Block Chaining protocol options and fall back to RC4, which everybody supports.

So if this were a much bigger problem, if it were more of a concern, then I would say maybe that ought to be recommended policy until we get more widespread support of

1.2 version that doesn't suffer from this problem because they're not carrying the initialization vector from the end of the prior block into the beginning of the next one. That's the little tiny mistake they made that has been fixed years ago, but no one has gotten around to updating their protocols because it just didn't seem like a big problem.

So anyway, I would say you should not be scared, Fabio. You might turn on 1.1 and 1.2. That way your system is offering the more secure protocols, and it will use them if they're available, and it'll fall back to 1.0 if they're not available. But don't sweat it. My sense is the browsers are going to get this locked down. And the good news is we'll be moving forward with a more secure protocol as a consequence of these researchers who showed everyone, whoops, there is a way to take advantage of this problem that we've known about since 2004.

Leo: All right. Our last - are you ready? - our last question.

Steve: And Netflix reversed themselves, didn't they?

Leo: They were going to separate the DVD business out, yeah. And they decided not to. Qwikster, they were going to call it. I don't, you know, Netflix is struggling a little bit because they lost a bunch of subscribers. Mostly it's a stock thing. The stock market's lost faith in them. I think they're great. I use them all the time. And I will never, you know, maybe you want to quit your DVD subscription. But I like that, too, because I like having a disk. But I just think the streaming is such a good deal. To me, that's the way to do it.

Miamiandy, who lives in Troy, New York - he probably just wishes he were in Miami - wants to comment on iPhone keyboard vibrations: I was attending CCS this past week and heard Georgia Tech people present their paper about the iPhone detecting typing via the vibrations it produced. We talked about this last week. Is it because of an accelerometer in there?

Steve: Yup, yup.

Leo: And you can actually detect the vibrations on the table. During the Q&A that followed it was pointed out, when they were probed for more detail, that this was only under some very specific circumstances. First, the iPhone was always on the left side of the keyboard in the same spot. Second, they made sure to use a wood desk. Those resonate the best, of course. When they tried it with another desk or on a concrete floor, it just didn't work. Third, they tried it in an isolated environment. This was pointed out to be an issue since vibrations from other items like desk fans or maybe even some speakers would ruin the results. It was an interesting paper, but I think the flaws in its proof should be pointed out so people aren't suddenly paranoid about leaving their phones on the desk. Please, don't ever stop the podcasts.

It's not their phones you've got to worry about. But if you see a stray iPhone to the left of your keyboard, and you're on a wooden table, and there's no other vibrations in the room, be afraid. Be very afraid.

Steve: And people are saying, "Shhh, he's typing."

Leo: "He's typing, shhh."

Steve: Yeah. So this was a very good point. I brought it up because I thought it was fun and clever.

Leo: It was cool, yeah.

Steve: And it perfectly demonstrates this whole class of crypto concerns known as side channel attacks where, without intending to, you're leaking information through an entirely unrelated vector that you wouldn't anticipate. For example, we've seen - and we're now aware of this, for example, modern ciphers and hashes are - they get dings against them in their competitions if the amount of power the processor consumes is a function of the keying material. So, or if the length of time it takes - newer hashes and ciphers are time invariant, meaning that it always takes them the same amount of time to do their work, independent of what your key is and what the data is.

So those are examples of, I mean, these are things we've seen attacked before. Timing attacks are real. They have been used to crack ciphers and to perform decryption. So anyway, this was just another wacky cool example of a side channel attack. But I did want, as Miamiandy says, let's not get ourselves carried away. It's not like our phone is spying on us and actually able to determine what we're typing by listening.

Leo: Steve, as always, you debunk; you explain; you teach; we learn. And love doing the show. We do it every Wednesday, 11:00 a.m. Pacific, 2:00 p.m. Eastern, 1800 UTC at TWiT.tv. You can watch live. But if you don't, don't worry. Steve makes copies available on his fine site, GRC.com. Not only 64KB audio, but 16KB audio for the bandwidth impaired. That's the smallest file size anybody offers on any show on this network. He also does transcriptions, which is just great, so you can read along as you listen. If you want video, we have that, as well as all the audio, at TWiT.tv, as well. And next week what are we talking about?

Steve: I'm not sure. We'll either continue with talking about some of the inner workings of TCP the protocol, or if something comes up in the meantime, there are a couple other things I wanted to research. So I may take us off that for a minute, tackle something else interesting, and then come back. I expect that next week I will be announcing the completion of Off The Grid.

Leo: Really.

Steve: I believe it's finished. I came so close yesterday to posting it for the newsgroup folks to pound on. There's a little different Firefox behavior. If anyone was curious, you could - I did post a PNG image of the grid-customizing UI. If you go to GRC.com/miscfiles/gridcustomization.png, you will see just a snapshot I took a couple days ago to show the folks who are in the newsgroup and waiting for me to give them something to pound on. You can change the font, the size, the horizontal and vertical padding for the characters; you can turn off the upper and lowercase randomization change, the run length of that, even the colorization of the various aspects of the grid.

So it's got all kinds of bells and whistles. And it's working. I'm just - I'm very close to having it finished. So I suspect next week - many listeners have been saying when, when, when. So I hope it'll be next week.

Leo: Awesome.

Steve: And we'll certainly have a good podcast, in addition.

Leo: No matter what.

Steve: Yup.

Leo: Guarantee you that. Thank you, Steve. Thanks for joining us, everybody. And we'll see - and for those who are wondering, I just thought I'd wear a security outfit during Security Now!, that's all. They said, "Is he moonlighting?" Yeah. Got to make a little money on the side. Thanks for joining us. We'll see you next time on Security Now!.