SECURITY NOW!

Transcript of Episode #328

Listener Feedback #131

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-328.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-328-lq.mp3

Leo Laporte: It's time for Security Now!, the show that talks about security online, privacy online, and a few other side issues like Vitamin D, eReaders, and science fiction. And that's all because our guy, Steve Gibson - the man, the myth, the legend - besides being a security expert, is a man of the world and has many interests. Good cabernet, we should include. We've yet to do a show on cabernet, however.

Steve Gibson: Oh, we'll let John handle that. Dvorak.

Leo: Hey, Steve. How are you? Yeah, Dvorak is the wine wiz.

Steve: We're sneaking this in under the wire of the Thanksgiving holiday. Elaine, in fact, shot me a note, since she does our transcription, saying, uh, it might be a day later than usual for the transcript. And I said, well, yeah, of course. You need turkey along with the rest of your family.

Leo: Oh, yeah, a holiday, that's fine. I do hope everybody's planning - everybody in the U.S., anyway, is planning a good Thanksgiving. You have to say that because in Canada they had Thanksgiving last month. And the rest...

Steve: What?

Leo: Yes, they do it in October. And the rest of the world's going, Thanks-who-ing?

Steve: Yeah. I also got some reminder of what an international audience we have when I guess I tweeted, yeah, it was, I tweeted a reminder about daylight savings time on Saturday afternoon.

Leo: I saw that, yeah.

Steve: And I got a lot of people saying, eh, we did that last week. It was like, what, uh, what? Where?

Leo: Yeah, the U.S. changed it...

Steve: I know.

Leo: ...a couple of moons ago.

Steve: That was an annoying year.

Leo: Broke a lot of things, yeah.

Steve: Lot of machines didn't know yet, so.

Leo: That's right, that's right.

Steve: In fact, I have a clock that says, "New daylight savings time or old daylight savings time?" It was built at a point when it was before the switch, but it knew it was coming. So I thought, well, that's pretty cool.

Leo: I think there are two interesting movements afoot that have no hope, but I would - actually three. I'm going to give you three movements that have no hope because we're so entrenched in our way of life that I think just anybody with common sense supports. One is getting rid of daylight savings time.

Steve: I'm there. That's my choice, Choice A.

Leo: Eliminate it. Two is getting rid of the penny, which nobody uses, and that copper is expensive.

Steve: More than a penny.

Leo: More than a penny, right. But still, no point. And three, get rid of the electoral college because clearly that does not work.

Steve: Yup.

Leo: And all three, any thinking person - actually, though, of the three, daylight savings time might be the one that somebody could dispute. Any thinking person I think would agree on all three. And there's not a chance in hell that any of them will happen.

Steve: No. I would also argue that, well, there's many problems with our political system. But it is a problem that the Senate has as much power as it does because you get overrepresentation of very low...

Leo: Low population, yeah.

Steve: ...low population states.

Leo: And that's exactly why, to really go off track, you'll never get rid of the electoral college because it gives these states like Wyoming more power than a state like California because they're a small population, but you've got your two senators, so you've got your two electoral votes guaranteed. And they're never going to go for it. So anyway...

Steve: I promised Eileen that we would remember to tell our listeners about the TWiT plan for the holidays as it affects Security Now!.

Leo: Oh, good.

Steve: And that is, apparently we're not going to, I mean, we couldn't run the Portable Dog Killer episode again anyway because we got away with it once, and I got a lot of complaints saying, well, Steve, you broke your "we've never missed a week" commitment. It's like, well, okay. But so tell us all what it is that you guys are going to do.

Leo: Do I know?

Steve: Yeah. It's like a "best of."

Leo: Oh, yeah [laughing]. Oh, that. I'm looking, "Eileen? Eileen?" Yeah, we're doing - so this is actually - it's good. I'm glad you mentioned it because I do want to send people to the "best of" page so they can help us because - I would say cast your vote, but it's more than that. Help us by picking your favorite moments from the past year.

Steve: But it's moments. It's favorite moments.

Leo: We don't want to do a whole show like we did last year. We want to get bits. Now, this is a little to tougher on this show because this show is really very fact-based, and there's not a lot of wacky, Steve dresses in a kilt moments. So...

Steve: Well, and, yes.

Leo: We may do the Portable Dog Killer if we don't get enough votes. Let's put it that way.

Steve: Someone commented that when you, a couple weeks ago, told me that it wasn't just iOS that was being sandboxed, that the announcement was affecting the App Store for the main OS X, apparently my look on the camera as I just stood there with my mouth open, looking like a moron...

Leo: That's a moment.

Steve: He said, oh, now, that's one we've got - the problem is it doesn't translate very well into audio. All you get is silence from me, so.

Leo: It may be - we may have to punt on this one. But if we can come up with a half an hour to an hour worth of great Steve moments from 2011, we'll do it. TWiT.tv/bestof. And it's not just this show. We're hoping to do every show a "best of" because we like to take the week after Christmas off. I'm going to go back East and visit family. And so we want to give all our hosts the time off, as well. So...

Steve: Well, we certainly - I was going to say, what we certainly can do, although this doesn't help us this year, is everyone be cognizant of this approach for 2012 and make notes of things that you think would fit.

Leo: We did it last year, but nobody remembered. It's always a last-minute thing. At least we're planning this one in November instead of the last week of December. So TWiT.tv/bestof. Your help is much appreciated. Favorite moments. And it could be - in this show it wouldn't be, like, wacky moments. It would be important security news. Your Bitcoin piece, for instance, I thought was very interesting. I wasn't here for that. Should be - I think parts of that should be repeated. I think we should probably repeat your discussion of Stuxnet. There are certain things that were newsworthy. So it doesn't have to be goofy. It really could just be the big...

Steve: Things that bear repeating.

Leo: The big stories of 2011 I think would be perfect. So keep that in mind. And of course...

[Talking simultaneously]

Leo: Yes, I'm talking to you, folks. And if nothing happens, Portable Dog Killer. The problem is, repeats of a podcast are kind of silly because you can download that episode and listen to it anytime you want. That's why we prefer the "best of," because that's something that we put some work into. Then one episode has just the highlights of the year, and I think that's a really great thing to do.

Steve: Let's hope we can have it.

Leo: Yes. The passwords would be good. I mean, I can think of a lot. Actually, I'm going to probably sit down and go through and say this one, this one, this one, this one. Because I think there's some really important newsworthy things that we covered this year. But...

Steve: In the meantime.

Leo: In the meantime, all that aside, I think you and I have some stuff to talk about from this week.

Steve: We do indeed. We're going to follow up on - I have some statistics from what happened to the SOPA, the Stop Online Privacy Act event which you and I covered as it was happening, on the day of it, last Wednesday, when I was up in-studio with you. And some interesting tidbits of news. Some feedback. Also we'll talk about my experience with the Kindle Fires that arrived. I had ordered two of them. And so forth. So I think we've got another great podcast for everybody.

Leo: You know what arrived today? A Nook tablet. So we might compare the Kindle tablet to the Nook tablet. Very similar, actually, in hardware. This one's a little faster because I think the extra RAM has made a difference.

Steve: It might, although it has the same dual-core OMAP processor.

Leo: Processor's the same. But I noticed the page turns are much more smooth.

Steve: And I notice that it's being advertised by your friend from Glee.

Leo: Yes, it is, Jane Lynch. Did a great job. No matter what she does. So let's talk. What's new in security news? I guess we'll start with SOPA.

Steve: Well, yeah. This was the reaction to this legislation, which was being viewed by many very popular websites as the most onerous and worrisome, sort of over-the-top government privacy-violating legislation yet - more so than the Protect IP legislation, which is probably, hopefully, thankfully stalled in the Senate.

Leo: That's the Senate version. Ron Wyden, all praise to Ron Wyden, who said "I'll filibuster it. I'll stop it." So we know that's not going to get through. Although the thing is, these keep coming up. I mean, maybe there'll be another one.

Steve: Yes. I mean, one of my favorite slogans, unfortunately - I'm very active in following U.S. politics. And my favorite phrase is "The best government money can buy."

Leo: Yeah. Bought and sold. But the thing that - I think this is a very good lesson. The thing they do with that money is essentially get votes; right? So ultimately we have more control as a group because our vote is the final arbiter of whether somebody gets in office.

Steve: That is true. And so of course, well, we don't really want to devolve into a political discussion.

Leo: No, please.

Steve: Special interests end up with disproportionate strength.

Leo: But we are special interests when we act in concert. And that's what the Internet has done, and that's what's so exciting.

Steve: Well, the bad news is we're fighting the MPAA and the RIAA and these large organizations that keep putting pressure to make these things happen. And in fact...

Leo: They want to break the Internet, frankly.

Steve: Lamar Smith, who is the Texas Republican representative who's one of the sponsors of this bill, the SOPA, Stop Online Privacy Act bill, he said, "Well, you know, I'm not technical." Well, okay. And this is the problem, is that one of the many things this does is it breaks DNSSEC. That is, DNSSEC is all about preventing DNS spoofing, which is essentially what this is, is legislated, government-backed DNS spoofing. And so many of the people have been concerned because essentially it means we can't have DNS security if we're going to have a mandated, legislated, deliberate breakage of DNS.

Leo: Because Lamar Smith says it's one quarter of all the Internet traffic is offensive infringement, and we're going to stop it.

Steve: Yeah. He said one quarter of Internet traffic...

Leo: One quarter.

Steve: ...is infringing.

Leo: And you know he's being spoon-fed that from the RIAA.

Steve: Precisely. Now, the good news is there was a serious groundswell that resulted, peaking last Wednesday during this day of everyone being called up and out to act. One million emails were generated to U.S. representatives by people around the country that cared about this. I don't know how they have this number, but 87,834 telephone calls.

Leo: That was from Tumblr alone. Tumblr...

Steve: No kidding?

Leo: Yes, that's the Tumblr number. Tumblr.com was very aggressive because they would be one of the companies that would really be screwed by this. They'd suddenly be responsible for every bit of content on their - and they're one of the biggest web hosts in the world. And they didn't want - so they were very aggressive. And that count comes from Tumblr.com alone. Talk about a great response.

Steve: Wow. So the average length of those phone calls was 53 seconds. The longest one was 31 minutes. Got to feel sorry for the representative who got on the other side of - hello? And a total of 1,293 hours of phone calls were spent talking to representatives.

Now, I wanted to give our listeners who care about this, as you and I do, a URL, because this is being organized around AmericanCensorship.org is the website. And at the top of their website they mention, essentially, distill it down to three bullet points that I thought were worth sharing. So under "Website Blocking" they explain that the government can order service providers to block websites for infringing links posted by any users. And they said "Risk of Jail for Ordinary Users." They explain, "It becomes a felony with a potential five-year sentence to stream a copyrighted work, even if you are a totally noncommercial user, for example, singing a pop song on Facebook."

Leo: Well, and we do this all the time, and it's our position that it's fair use as a news organization. It's protected. But that doesn't mean that they don't take our shows down all the time. And we would be faced with prosecution. And we might be able to defend ourselves; we might not. But it would certainly cost us a lot to do so.

Steve: And remember, it's certainly possible for people to take the position, oh, well, yeah, but that would never happen to a regular user. But let's remind everyone of the case of the innocent mom of, like, I think her kids were, like, three and four years old, who was attacked by and sued by the MPAA for movies that were found on her machine which were loaded and being redistributed by malware that she had no idea was there. And some huge, tens of thousands, for some reason the number $64,000, I mean, literally, the courts were coming down with cash judgments against her, requiring her to pay this money. So this kind of thing does, I mean, and can happen, and apparently will. Anyway, I don't mean to...

Leo: I think the most important thing...

Steve: ...get worked up.

Leo: ...for people who listen to this show is the message that it would break DNS, that it isn't a good solution. It would - and I didn't know, but that it would impinge on the ability to do DNSSEC is huge. I mean, we have a fight on our hands to get everybody to implement DNSSEC as it is. But it's clearly...

Steve: And we need it to prevent spoofing. And this is mandated, legislated spoofing.

Leo: It's incredible. It's incredible.

Steve: Yup. And finally, the last bullet point on AmericanCensorship.org explains, under "Chaos for the Internet," they said, "Thousands of sites that are legal under the DMCA" - which I already have big problems with because it prevents, for example, researchers from being able to reverse-engineer crypto technology in order to research it - "would face new legal threats. People trying to keep the Internet more secure wouldn't be able to rely on the integrity of the DNS system." So it's just - it's bad. And I think the point you make, Leo, is I don't know if we're going to win this, ultimately. There is such continuous pressure from the powers that be, that do not want the Internet to be free and open, that want control over it, I don't know. I mean, I'm glad everyone's putting up a fight.

Leo: I think we can win this. I think we won this round because I think enough members of Congress got the message. When you get 87,000 phone calls, that's a significant number. Now, somebody in the chatroom said, well, it's not the people who listen to this show, it's the dumb people we have to convince. That's not true. There are 70,000 people listening right now to this show. There are plenty of technically sophisticated people. There are more than enough. Because remember most people never call their member of Congress, never have anything to do with them. So each call counts a lot. So we do have the power. We can fight this. And I think we will win in the long run.

Steve: And the other thing all of our listeners are, are opinion leaders. I have, in a Q&A that we'll get to later, a point is made - and I chose this because I wanted to make a point to our listeners to tell their less security-aware friends, to remind them of the importance of something. So that really does need to happen.

Leo: As Puppy says, though, really the problem is complacency more than anything else. Let's not be complacent. We've got to continue this fight, continually vigilant, because it's our Internet. And, yeah, most people may not realize the threat. We do. So we're the ones who have to fight.

Steve: Yes. Yes. Now, a ton of news was made in the last week, and my Twitter feed was full of people bringing this to my attention, about something that may or may not have happened. And that was the news that an Illinois based water district had its SCADA system, the industrial control system for running it, hacked. And the consequence of that was that a critical water pump was burned out.

Now, the question is whether this was true or not because now the Department of Homeland Security, who has a division called ICS-CERT, which is the industrial control systems cyber emergency response team, emphatically says that there is no evidence whatsoever of any external intrusion from Russia or anywhere else. And what's odd is that the original blog posting that got picked up by the news organizations, and even an executive at that water district, who was then on, I don't know if it was radio or TV, but on some public live media, contained all kinds of information about log entries that were found, and IP addresses belonging to Russians. And that for two or three months there was, like, odd behavior being observed by this system, and that apparently it was that the SCADA system was being, like, shutting down and then coming back up. It was the fact that it was offline for a while that caused this pump to burn out because it wasn't - it lost its supervisory control technology that it was relying on and overheated.

So I don't know what to think. You could see some pressure on the part of the Department of Homeland Security because this news got so much press and so much attention that they could be wanting to tamp down on anyone worrying that, okay, well, a water pump today, a nuclear reactor tomorrow. Which of course...

Leo: Right. In fact, break a water pump...

Steve: ...is the concern.

Leo: Right, break a water pump in a nuclear reactor, you could have a meltdown. So this is not insignificant.

Steve: Yeah. So I don't understand.

Leo: The DHS and the FBI say that we found no evidence of a public water utility hack. We also read that their password was three letters.

Steve: Actually, it's not their password. There was some other confusion there.

Leo: Oh, okay.

Steve: It's that there are SCADA passwords in use actually in other places, as I understand it, that are three letters. And in some cases they ship with three-letter passwords, and no one changes them.

Leo: Oh. Oh.

Steve: Yeah.

Leo: ABC.

Steve: Actually I think the one I saw was "100" was the - it was one zero zero was a sample three-letter password that was in use somewhere.

Leo: Somebody needs to remix the Michael Jackson song. "Breaking into SCADA's easy as 1, 2, 3, A, B, C." Oh, boy. Oh.

Steve: Yeah. Yeah. So we don't know. There was some good news, and that is that Google is moving forward with some efforts to increase the security of connections to them when using either their browser or Firefox today. And apparently there's support for this in - and I think it was one of the later versions of IE under Windows 7. And that is, remember how we've talked about the way SSL works, where you have a suite of available cipher systems which a web browser offers to the server. And we were talking in the context of CBC, the one particular protocol used by block ciphers, had a problem under SSL prior to version - or actually TLS v1.1 and 1.2. So TLS v1.1, which is SSL 3, had this problem, and that it was possible to simply fall back and not use the CBC technology, but use RC4 as your block cipher, which then would keep you from having this problem.

Well, in a different sort of tangent off of that, the guys at Google have implemented something known as "Ephemeral Diffie-Hellman" encryption. And actually that's a key agreement protocol which is very efficient. The efficiency of that allows them to change keys often. And changing keys often, that is, like for every secure session that you set up, is very good because that creates something known as "perfect forward secrecy." And several articles that I saw talking about this were written by non-crypto-savvy people who said that forward secrecy was a protocol, and it's actually not. It's a feature of - it's something that you get which is a good thing when you change your key often because the point of it is that it keeps you from ever going back in time. If someone were to hack a key that is now in use, they wouldn't be able to go backwards and hack stored encrypted sessions because those would have been encrypted under a different key.

So what Google has done is they've augmented the normal security suites that are available with some new ones which inherently use ephemeral keys which are easy to compute and won't load down servers. They have built this into some updates to the OpenSSL suite and made that publicly available. So what'll happen is this will filter back into OpenSSL with a future version of it. It'll be available. It'll then get pushed out and built into next-generation fundamental SSL suites that are available in the UNIX flavor OSes. And apparently Microsoft is already in the process of adopting this. And then, as our browsers are made aware of this, and currently Chrome and Firefox both are, we'll just all start using this, and that'll be a good thing.

So this is just a really - it's a perfect example of when a protocol was thought out well, and it was inherently designed to be upward compatible, how you can slipstream good evolution into that protocol, never breaking anything, and just automatically taking advantage of innovation in the crypto that just sort of filters out into everything.

Leo: Maybe Vint Cerf was right. He said it's the self-healing Internet.

Steve: Don't throw it away, yes. And I did want to make a little TSA announcement. I saw for holiday travel that children under 12 no longer have to take their shoes off.

Leo: Yeah. I know that because I traveled with a child under 12 about a month ago.

Steve: And has that always been the case?

Leo: Nope, just happened. Literally about a month ago, yeah.

Steve: That's what I thought, yeah. So although people...

Leo: So I guess they figure there'll be no shoe bombers under 12.

Steve: Well, the bad news is I think today, as we're recording this, people are probably already in line, so they may not be hearing this announcement.

Leo: Oh, they put - the way I found out, they put big signs up at the airport.

Steve: Oh, good, good.

Leo: Yeah, yeah. So but you still have to do all the other stuff. In fact, those security lines have gotten crazier and crazier and crazier. I got - usually I get scanned. SFO uses millimeter wave scanning, which is not an X-ray and supposedly not dangerous. But I got a backscatter scan in Vegas, the last time I left Vegas.

Steve: Actually I got a backside scan.

Leo: I don't want to know.

Steve: I'm not kidding. I guess - it was one of the trips I had taken recently. And I thought I did everything right. I raised my arms above my head and stood there. And then the person said, "Do you have something lumpy in your right back pocket?" And I had some, just some bills folded in half. And it was sensitive enough that I took my wallet out of my left back pocket where I normally keep it, but I didn't realize that just folded paper would upset it. Then I had to go through the whole pat-down routine because now suddenly that sets off their alarms, and they're like, okay, well. So I guess that was a backside scan that I had.

Leo: That happened to me, and I turned out - I think it was just my shirt was bunched up when it was tucked in. I had nothing in my pockets. I didn't have a belt on. It was all fabric. And they still said, we've got to scan this back here. And I think it was just probably my shirt was bunched up. I mean, this is ridiculous.

Steve: Well, and Leo, I don't want to get off the track, but I haven't ever said this before except to friends of mine. All of these problems that we had were from foreign flights coming into the country. The fluid mixing thing was - I think that was - was that out of London that was coming into the U.S.?

Leo: Yeah, I think it was.

Steve: And the shoe bomber, the underwear bomber, all these variations of bombers, they were not when people were flying from Orange County, California, 500 miles north to Northern California. I mean, think about it.

Leo: It's security theater. I mean, that's what Bruce Schneier calls it. He is, of course, a great security expert. And he is very vocal on what we could do to have effective security as opposed to what we are doing, which is essentially theater.

Steve: Oh, and look at the cost to us, to us citizens. Anyway. So last Wednesday you were showing off your Fire, your new Kindle Fire, and I was drooling over it and telling you that mine were waiting for me. I was so excited about it, back when it was introduced, that I bought two of them under the theory that if one is good, two would be better. They've both been returned to Amazon.

Leo: Yeah. What happened?

Steve: Well, in fact I tweeted. I said a couple days after that I saw a Kindle Fire teardown where its cost was estimated, and it was estimated at cost, that the $199 Kindle costs Amazon $201.70. And so I tweeted that. I said the Kindle Fire teardown shows it costs $201.70 to make. Of course that doesn't factor in the cost of return shipping.

Leo: Oh, interesting. You had to pay for return shipping?

Steve: Oh, no, no. They did. But I'm saying that that was also their cost, which wasn't factored into the Kindle...

Leo: Right. They lost more money on you than anybody else.

Steve: Well, I mean, and in fairness - okay. So I plugged it in, charged it up overnight, and went to Starbucks bright and early on Thursday, the next morning, to have a nice sit-down with it. And I started not being that impressed with it. The power button sticks out, so it will turn itself on or off if you rest it on its lower edge, which is why many people have power buttons that slide, because that's not something - that's not as natural an action. But and many of the Kindles, the early Kindles slid. The newer Kindles at least don't stick out, except the Touch's sticks out; whereas the regular, what they call the Kindle now, just the Kindle Kindle, its button does not stick out. But then little things.

Leo: It's a little easy to hit that button, I agree with you, though. I think that that's not a good place for it.

Steve: Yeah. And unfortunately that's maybe the only thing which doesn't fall into the category of they can fix it, because the beauty of any of these...

Leo: It's all software.

Steve: Yes. Any of these state-of-the-art devices is that it is software. And so everything I'm going to complain about, I understand can, hopefully will, get fixed. But, for example, the bright - and these are, as I was playing with it, I was thinking, Steve Jobs would have never shipped this. And so, like, it wouldn't have gotten past him. For example, the lower 25 percent of the brightness control is defective.

Leo: Is black.

Steve: It does nothing.

Leo: Does nothing.

Steve: The lower 25 percent, you slide it back and forth, it has no effect whatsoever. It just bottoms out. And so they need to rescale that.

Leo: But that's, as you say, easily fixed with a firmware upgrade, yeah.

Steve: Easily fixed. Also on the title page of a book that I was experimenting with, as I dragged it back and forth with my screen, I was able to get it to leave debris behind. So it wasn't properly refreshing the screen as I moved it. And it was jerky and not very smooth.

Leo: I think you got a - more bad ones. I think that's another bad - I haven't seen any of the graphics issues that you just described. And while the screen is not quite as fluid as an iPad, admittedly, I haven't - it's pretty fluid. I haven't seen any issues with it at all.

Steve: Well, I wanted...

Leo: I think a lot of people are having trouble.

Steve: I wanted - oh, they are. I wanted to see whether the cover flow was better in landscape orientation than it was in portrait because it is unusable in portrait. It is just - it's awful. Now...

Leo: Oh, you know what, Steve, you have a bad machine. No, I'm not kidding. I wish I had mine here that I could show you. The cover flow works fine in portrait, in landscape.

Steve: I've had other people complain. I've seen other people complain about the cover flow. It just, the way it works - now, maybe it's that I have 247 things in the archive.

Leo: Well, that could be it.

Steve: Except that ought to make it just deeper. But, I mean, it doesn't - it's difficult for me to, like, bring something to front. It was snapping off to the left prematurely. And again, fixable by software. Font selection...

Leo: I think fixable by getting one that works. I think you've got - I'm not kidding. Let me - here. I have one here. Let me - I have to set up the screen so you can see it. Let me just see if I'm getting the same effect on this Fire. Because it's completely smooth on both portrait and landscape mode. Let me pull up a shot of it. I don't have an over-the-shoulder shot. I wasn't planning this. Here, all right. Here is - this is not my Kindle Fire. This is Liz's. I mean, it's a little slow updating the image the first time through, but that's completely fluid to me. Are you seeing it?

Steve: I actually can't. My video from you froze quite some time ago.

Leo: Oh. These things do happen. So, I mean, I feel like that they may have a lot of hardware defects, and that you may have a bad graphics card in there. Interesting, yeah.

Steve: Well, I think it's design. I think it's a very bad UI. I was unimpressed with its navigation, which seems inconsistent and often unclear.

Leo: Really.

Steve: Like how to move around. I mean, I'm asking a lot from it. I will say, yes, it's $200, and that's an amazing price for a tablet that has this much potential. But at this point I'm very unimpressed. It needs major revision. For some reason it's already at Rev. 6 when it comes. And it may be that both of mine - because both went back. Both of them hung completely. And I did order instantly, so maybe they were literally the first ones off the assembly line.

Leo: That's what I think. I think there are probably several sources for these. And I'm wondering if some of the sources are just not making good ones.

Steve: Yeah, well, as I mentioned to you before we began recording, when I Googled "Kindle Fire frozen," I immediately found other people having the problem, and I made three postings in an online Amazon-based thread that drew attention. I tweeted about the problems. And I just considered it, unfortunately, a very bad launch failure. But I did hear that you and Paul talked about it that same day, on Thursday, and really liked the Kindle Fire.

Leo: Well, what's interesting to me is I'm hearing very different experiences from people. To me, I mean, I was just showing the page turn and the cover flow, and it's snappy. There's a little tiny bit of hesitation, tiny. I mean, you'd have to be a little picky in the page turn. Certainly nothing like the actual physical hesitation in a page turn or like a Kindle. I think for 200 bucks this is, well, I think for any price this is an amazing product. It's certainly the best Android tablet I've ever used. I agree there are some flaws. I think the on/off switch is a little easy to hit, although I don't hold it that way, so I never hit it. I really like it.

Steve: So you would call it "the best Android tablet to date."

Leo: Oh, easily. But that doesn't say much because most of them are pretty horrible. I mean, that's really literally not saying a whole lot. But I just feel like...

Steve: I think I'll probably wait three or four months for them to settle down, for whatever problems they're having with production to, I mean, I wouldn't want to get one and then have them improve the hardware design. And they do tend to do that.

Leo: I'm wondering, I'm thinking that there's a variety of hardware out there, or manufacturing problems with some of them. And might have something to do with that. I mean, I found this to be - and Kevin Rose was on TWiT saying the same thing, "This thing is horrible, I can't...." And people like Paul are saying things like, well, compared with the iPad, of course it's not as good, but for the price it is very good. But I actually, even, I mean, at 200 bucks I think it's amazing. And I've been recommending it to people. So I just have a very - I have a very different experience.

Steve: Well, I would not recommend it as a book reader.

Leo: Well, I like the eInk. And I do say that. I mean, I do point out that, if you're reading in daylight, or you want really crisp text, then this is probably not a good choice. There is also an issue with this, it signs you into your Amazon account. So I had a caller on the radio show, said "I want to get a tablet for my girls." And while I would recommend this for kids because the price is right, and I don't think they'll have the same issues that we have, their problem is you can't lock down the purchases. So you're giving a kid basically a device to buy anything they want at any time. And I don't know if that's such a good idea.

Steve: Yeah, you're giving them your credit card with no controls over it.

Leo: Right, right.

Steve: Yeah. Also the Touch I have a problem with, which is that - and again, fixable by software. But first of all, I'm not a big fan, I think, of touching the screen in order to change pages. I really like having a physical button. And my favorite Kindle of all time actually turns out to be the DX. I've been reading, I mean, a lot recently because I'm just so in love with these Honor Harrington books. And I've come back to the DX, just because of its large screen, which is so comfortable for me. And in fact I have two of them. I'm probably going to bring one up to Northern California and see if my mom wants to upgrade hers from the one that I got her a couple years ago, which was probably the Kindle 3, to the DX, just because she's in her 80s, and...

Leo: It's big, yeah.

Steve: ...I think she would probably appreciate the larger screen.

Leo: My wife has inherited the $70 or the $80 Kindle, the basic Kindle because I've got the Fire, and she loves it. I think the Kindle is a very accessible product.

Steve: Yeah, now, my problem with that one is I think it's almost too small. My favorite practical Kindle, I recognize the DX is not for everyone because it's $379, and that's really pricey. But my favorite one is what they now call the Kindle Keyboard, which is what they used to call the Kindle 3, because it's got that paddle at the bottom to hold onto. For me it's just easier. I like having switches on both sides, which of course the $80 Kindle also does. But it's almost - there's almost nowhere to hold that little Kindle, the newest one, because it's just like they've removed the margins, and there's no more keyboard at the bottom. And so it's sort of a little difficult to, like, get a grip on it. But again, it is super small, and they really have improved the page turn, too. They no longer do the big whole screen inversion paint. They only do that every six page turns. So five out of the six page turns just change the text. And it's very pleasant.

Leo: Oh, I didn't notice that. I didn't notice that.

Steve: I first saw that on the earlier Kindles in the Table of Contents. I noted - or maybe it was in magazine reading. It was something - or news, news subscriptions. There were several places where they weren't doing the whole big black inversion. And I thought, wait, a minute, how are they getting away with that here? And what they've done is they've extended, well, they've extended it so that it - I think what happens is there's, over time, there's some buildup, sort of like some drift. And so they said, okay, well, we'll let you do sort of the easy-on-the-eye page turn five times. But when you do it a sixth time, we're going to sort of like, like the Etch-A-Sketch, erase the whole screen and then redraw it in order to clean up anything that might accumulate.

Leo: Now, I'm curious. I just got, to compete with the Kindle Fire, I just got the Barnes & Noble Nook tablet. I thought, if I'm going to review the Fire, I should review the Nook. And some of the things that you'll like immediately, first of all, it feels thinner and lighter. It's got beveled edges instead of square edges. It's got an on-off switch here, where you're less likely to hit it. It actually reminded me a lot of the Kindle Fire, with one kind of small exception. It's a little faster, and I think that's because it has more RAM. It has the same processor, but it has more RAM in it.

Steve: Yeah, I think the Fire has 8GB of RAM?

Leo: No, RAM is 512 on the Fire. RAM.

Steve: Oh, 512. That's right.

Leo: Yeah. And this is a gigabyte of RAM.

Steve: Okay.

Leo: Yeah. But it's very similar in a lot of ways. It's funny, and I'm not sure why this is, its refresh rate is not the nominal 60Hz that most screens are. You notice we didn't get any - you can't see it, but we didn't get any flicker on the Kindle Fire. We're getting, on all our cameras, we get a lot of flicker on the tablets. So they've got an odd refresh rate going on here. I'm not sure why. But I do think it's a very similar product, maybe a little faster. So if you could live with the fact that it's not Amazon...

Steve: And the price?

Leo: 50 bucks more. Which is not, in my opinion, a good idea because people are going to pick the bigger brand for 50 bucks less. I don't think very many people say, oh, it's got double the RAM. I don't think that comes up at all. Anyway, enough about eBook readers.

Steve: Okay.

Leo: I think you still got a hardware malfunction in yours. Your new one.

Steve: I don't think so. I think I'm just picky.

Leo: Okay. When you come up here next time we'll have a head-to-head faceoff.

Steve: Well, I'll have my own. I mean, I will, I ought to have one. I have multiples of all the other ones.

Leo: Might as well; right?

Steve: Ultimately I think I should have one. I would like to have an Android tablet. I don't have an Android tablet yet, so...

Leo: It's, in my opinion, the best Android tablet. It's limited in some ways. It's not updatable. Amazon has to update it. And you don't have access to the full marketplace. You have to get stuff from the Amazon store. But it's still an Android tablet.

Steve: Yeah. And again, I'm also in a position, as are you, of being able to buy individual devices for individual purposes. So I have a DX because I like reading on that large eInk screen. And I have the smallest Kindle because it's nice to have one that goes in my pocket. And I've got my iPad for everything else. But if someone had to just choose one device, and budgetary concerns were forefront, then I think this does it all for 200 bucks. You get a tablet half the price of the iPad or less than that, and also a useful reader.

So some tweets. David Wright, who tweeted - he said he's an Englishman in Germany. He sent a mention to @SGgrc: "SpinRite saves the day again. 2004 laptop back up and running again, ready for another couple years of faithful service." And David Ward, tweeting as @DaveQB11 from Sydney, Australia, said @SGgrc: "Left Firefox 4 open overnight at work. And today we have it consuming 4.6GB of memory." 4.6GB. So, David, you get the record. He says, "Only about 85 tabs."

Leo: Well, look on the bright side. At least it's a 64-bit app.

Steve: Exactly. I will mention something that I caught it doing, and that is, I watched it as I was in Task Manager. I had Task Manager open, watching the memory just, like, kicking up over time in Firefox, just going, like, every time Task Manager would refresh, like very few seconds, it would be larger. So Firefox was just growing continuously. And I thought, you know, I wonder. And I closed an open PDF, and it stopped.

Leo: Adobe! Blame Adobe. Oh, that's interesting.

Steve: Yeah. So, and I wouldn't be at all surprised if among those 85 open tabs that Dave Ward had, some of them were PDFs that were being viewed. Often the case for me. So that may be part of the problem. And then Simon Zerafa, who is a frequent Twitter and contributor, he send a fun quote that I liked, attributed to Samuel T. Redwine, Jr. The quote was: "Software and cathedrals are much the same - first we build them, then we pray." So...

Leo: Love it. So true.

Steve: And also a new URL for Chrome. Eric Vollbrecht in Minnesota, he said: "@SGgrc Memory stats for all running browsers go to Chrome." So you open Chrome. And it's funny because when he said all running browsers, I didn't realize he was serious, but he was. So Chrome://memory-redirect/. You put that URL into Chrome. And the page is titled "About Memory: Measuring memory usage in a multi-process browser." And what surprised me, because I didn't read his tweet closely enough, was it showed not only a breakdown of all the processes that Chrome had spawned, but also Firefox. And then I thought, wait, wait about IE? So I fired off a copy of IE and refreshed the page, and sure enough, IE was there, too. So I don't know quite why they're doing that, but it's sort of interesting to see how it all breaks down.

Leo: That's so you can make the comparison.

Steve: I guess that's the case. Although it actually wasn't very flattering in Chrome's case. But I have many more tabs open in Firefox than I do in Chrome because I'm still predominantly over in Firefox.

Leo: It does say there's a bug that they seriously over-count their own memory usage. I love Google. Again, engineers, they're going to tell the truth. I'm going to launch Firefox, and I'm going to launch Safari, and I'm going to see what happens. How interesting, that they would build that into Chrome.

Steve: And I also had a tweet from @blindbites, who said also, he said, regarding SpinRite, he said, "Thanks for SpinRite. Saved my PC once again. No exciting story. PC wouldn't boot, and lockup during system recovery. Ran SR, one hour, all okay." And I didn't realize until just now, but the story that I had, a little bit longer, to share with our listeners about SpinRite, was from David Ward in Sydney, New South Wales.

So it's probably the same David Ward who tweeted about Firefox and all of his tabs because he said: "Hi, Steve. I've been hearing the testimonials and mentions of SpinRite for a few years now on Security Now!. I thought it sounded nice for those without any options, but I can do a bit of data recovery if I need it. Plus I have backups of all important data. So it's not a problem for me. Plus I thought, how can a reformat/re-zero not solve any problem anyway?

"Well, I recently had a disk in our MythTV media machine that started acting up. It was a disk in the recording pool of drives, so some of our recordings" - he said, i.e., girlfriends - "were on it. Mounting the disk separately was showing it was empty. Not the end of the world. Well, maybe for me, but not for anyone else.

"So I did a manufacturer's check on the drive, and it came up fine. I did some other little things, and the disk continued to check out fine. Still no data upon mounting it. Skeptical, I talked a friend out of loaning me his copy of SpinRite, with the promise to him and myself that, if it did anything, I would buy it. But I was also pretty sure it probably wouldn't.

"Running at Level 4, I saw it seemingly stuck at 41 percent on a spare test Pentium 4. So I changed out the motherboard/CPU, et cetera" - wow, okay - "and resumed from there." I'm not sure why that was necessary, but anyway. He says: "A day later, we're done, and we have all our data accessible. So there, Steve. Have my $89. I just purchased it. Well worth it."

Leo: Yabba-dabba-do.

Steve: "Well worth it for SpinRite and all the other information you provide me on the podcast. Keep up the great work.

Leo: That's a neat story. And we would tell people, if you use SpinRite, and you see it pause at a point, that's the problem sector. And it's reading and reading and reading, and it won't give up until it gets that data off of there. That's all that was going on.

Steve: Sometimes it can take a while, but it will ultimately move forward and probably have done that drive a world of good, in this case.

Leo: A world of good. All right, Steve. I've got some questions here. Are you ready? You feel good? You ready to go?

Steve: Ah, yes.

Leo: I've got all these browsers open. Let me close a few so I can see the questions. Terri Bell starts us off with a reason why HTML5 is not a Flash killer. We were talking about the fact that Adobe is abandoning Flash on the mobile platform and going to support HTML5. Flash applications are compiled, and compilation produces binaries that take effort to reverse engineer. A Flash application may also be obfuscated to afford "protection" against Flash decompilers. In contrast, the source code for HTML5 applications is there for anyone to grab. It can then be modified and similar competing applications or services set up with a fraction of the original development cost. Although obfuscation and moving application logic to server side will help HTML5 developers protect their work, some developers will not see that protection as adequate and choose Flash instead. Thus HTML5 will replace Flash in many instances, but not all.

Steve: Well, so what do you think? I was trying to think, okay, what's that valuable that people are coding in Flash? And I guess games is - you wouldn't want Flash games to be reverse engineered...

Leo: He has a point. I'm sure that there will be - there are ways to obfuscate JavaScript, but they're pretty easily cracked. I wouldn't be surprised if there were ways to p-code it or byte-code it so that it isn't legible. That's an interesting question.

Steve: Well, actually, Flash is compiled into a p-code. The actual runtime is an interpreter that interprets Flash p-code.

Leo: No, I understand, yeah. But that's what I'm saying, is that JavaScript would need something similar to make it - I don't know. I just think Flash is dead, no matter how you feel about it, or how proprietary you'd like to make your code, if no mobile device supports it. They are going to continue, they're going to make one more version for Android that'll be the final version. But say two years from now, you're losing out on the mobile market, which will be, by then, the No. 1 way to surf. Are you going to use Flash? I don't know. I mean, you might use it, I guess, if you're Yahoo! Games.

Steve: Yeah, I don't think anybody today even, given the anti-Flash position that Apple's mobile products have, if they didn't have to, would use Flash. I think you'd just say okay, HTML5 and JavaScript. We can do what we need to that way. So.

Leo: In the chatroom, Web755 is pointing out there is a Flash decompiler.

Steve: Actually I've used some. When I was putting videos on my site, I wanted to understand, I guess I wanted to tweak something that was - it was free, but it wasn't available in source form. And so I used a decompiler in order to get it back so that I could tweak it and then recompile it. And that approach worked just fine.

Leo: So maybe it isn't so secure.

Steve: Yeah, it's not.

Leo: It's not any different than obfuscated JavaScript.

Steve: Well, yeah. And remember we have our standard - our standard refrain is, if it's going to run on a user platform, then you cannot protect it.

Leo: Right, because it has to be decompiled by the platform.

Steve: Yes. And so there's lots of clever people with lots of time on their hands, apparently, who say, oh, let's write a decompiler. That'll be fun. It's a senior project for computer science.

Leo: Rosen Penev in California has our next question: Steve, I've been a listener since the Bitcoin episode, and your Password Haystacks page is fascinating. Your explanation of it was awesome. But a question recently popped up. You mentioned how the bad guys try to use dictionary attacks and then progressively trying out more sophisticated patterns like numbers and symbol combinations. My question is, how likely is it for a bad guy to use a different alphabet to try to crack a password? My guess is highly unlikely since most people don't use alphabets other than the Latin alphabet. But I'd appreciate it if you could comment on this since this could potentially mean that you can have passwords that are short and secure - I guess by using Cyrillic or something. Great podcast, as always.

Steve: Well, so, okay. The problem is one of compatibility. We run across it with sites that won't allow special characters, for example, even sort of a little bit off the map, but not even very far off the map characters, or even Unicode in some cases. What you'd really like is a site that will take whatever it is you give it and hash it and store the hash. In which case it's only incumbent upon you to be able to recreate the same thing again. Now, even that can be a problem because not all keyboards are equally capable. It's possible, for example, with a traditional PC keyboard to hold down the Alt key and enter the decimal code into the number pad and have that work. The problem would be, what if you wanted to log in on your iPad to the same site? You wouldn't be able to recreate that same process.

So unfortunately we're reduced at this point to a world of least common denominators in order to get cross-site and cross-application and cross-platform compatibility. So while it's tempting to imagine putting some funky characters in from some different language, you need to make sure that you're able to do that wherever you want to be, and of course that the site honors them correctly. For example, if I did that, say that I used a "c" with an umlaut over it or something, or maybe an "o," you'd want to make sure that just typing in a regular "o" didn't also log you in, in which case you wouldn't have achieved anything. It would be reducing the strength of your password without your knowing it. So, yes, the idea is absolutely a good one, that is, of using strange characters and different alphabets. The problem is one of getting sufficient compatibility, which I think would probably be a showstopper.

Leo: Patrick Moran, London, Jolly Olde England, writes about an article about triple-DES. He says: Is nothing safe anymore? 3DES, the triple use of the Data Encryption Standard - because DES was cracked, so you do it three times, now it's safe? No, it's been cracked, according to Electronics Weekly. Patrick Moran, a very happy SpinRite owner. I take it you probably looked at this article.

Steve: I looked at the article. There was a reason I chose it, because it had the best analogy I've ever heard of a side channel attack. And I didn't want to forget to mention it to our listeners because it was just so great. A little background: DES was the Data Encryption Standard - thus the acronym DES - which if memory serves was a 56-bit key. And that's not enough bits anymore. And DES itself had some structural problems. So what was recommended was to do it three times. And that might sound like, well, how can you take something that's not secure and just do it more and you get security? But that's exactly what our block ciphers do. Remember that AES runs multiple iterations - they call them "rounds," multiple rounds - using different pieces of key material. No one round is at all secure. And in fact reduced round versions of AES have been cracked because you rely on the successive rounds in order to get the strength. And when you get enough of them, you just can't penetrate it.

So, similarly, taking a block cipher like DES and doing it three times - not using the same key three times but essentially three different keys. So you take 56-bit key times three would be your new key length. And then you use a different 56 bits out of that key for each of the three times. And you end up with something which is still very quick because DES had the benefit of being a pretty quick block cipher, yet you get really strong security. Oh, and in this case it's being used in some electronics-based cards in Europe. And it was one of these cards which was cracked using a side channel attack. And we talked about that recently. In fact, they noted that power consumption being used by the card when it was in use leaked enough information for them to obtain the key.

Leo: Wow.

Steve: So, and we've talked about how - yes, isn't that cool? - that just, if you're sharp enough, and of course we're talking about sharp people, just noting variations in the amount of power the card is consuming while it's processing the crypto, if you understand the way the algorithm interacts with its power consumption, that can be enough in order to give it away.

So here's the analogy. Just such a perfect analogy of a side channel attack on old-school protection. And that is a safecracker listening to the tumblers drop. Because when you think about it, I mean, in the same way - and we've talked about this before, Leo - the hokey way that long combinations are broken in movies, where they stick this cracking device up against something, and all the digits begin spinning, and one by one they lock in, that's exactly the way a safe is cracked because you are - I don't know if anyone is too familiar with the insides of these things. But you have essentially a stack of disks which are notched, and the goal is to line up all of the notches, and then the disks also interact with each other, pushing each other. And so it takes - and that's where you get this turn it right five times to sort of get all the disks lined up, and then continue to this number. Then go back to that number, the other direction to this number, back to that number and so forth.

Well, each of those processes leaves a successive disk lined up correctly until, when they're all lined up, you're able to have the bit essentially fall into all of the lined-up disks. Well, safecrackers learned that, if you stuck a stethoscope on the safe or something else, you could audibly hear that pre-alignment of the disks before they all got aligned, and crack the safe. And so that's a perfect example. The designers of the safe never wanted you to get any information at all as you were turning the knob back and forth. And so it is the case that by using this audible side channel attack on a safe, or at least really old-school safes, that safecrackers could detect when the last disk was aligned, and then the second to the last, and the third to the last, and the fourth to the last and so forth, until they got the safe open. So they were able to, just like in the movies, get individual digits at a time, the equivalent of that, one at a time using sound as the side channel attack on the mechanism, which I just thought was a cool analogy.

Leo: Question 4 from Joe Campana in Ontario, California. He grumbles [grumbling] about Adobe Flash updates: Steve, I don't know about you, but in talking with my non-techie friends, I found they're very frustrated, as I am, when updating Adobe's Flash Player. Primarily, is there any logical reason why we are faced with their user agreement every time we update this? Adobe can't actually be changing the terms for every update, can they? I got into a white heat yesterday when faced with this again, for the hundredth time. It seems to me there could be an agreement that would be affirmed when it's installed initially and could be reaffirmed occasionally. Maybe we can make this annual. More importantly, when faced with the Update Flash window, a majority of my friends just click Cancel and move on, not knowing that an update is a good thing in terms of safety.

I believe that if there's any software that should update quietly, in the background, this is it. Of course there should also be the ability to roll back to an earlier version should an update break something. But for the general public, I overwhelmingly suggest that Adobe consider this approach. Sadly, Adobe seems to be following in Microsoft's footsteps in that they don't seem to use the software they're writing. Have they ever tried to walk their mother through an update over the phone? Thanks for everything you do. Signed, Joe in Ontario.

Steve: So that caught my eye. This was the question that I mentioned I was going to get to relative to our listeners, who I'm sure are updating Flash, probably in something of a sweat or a panic when they realize that Adobe has just fixed 12 things which are all being actively exploited to install Duqu or the malware du jour on their system. But the idea and, well, the idea that Joe mentioned, that he believes he's got friends who are not doing so just because they're annoyed that they're having to do it all the time...

Leo: Or they don't get the message. They may say, oh, it's working fine for me. I don't have any bugs. I don't need to update.

Steve: Right. It's not about Flash not working.

Leo: Right. It works fine for me. Why do I - I'm not going to update it.

Steve: And at the same time there are all these anecdotal reports of everyone apparently on the planet getting infested with malware all the time. Well, this may very well be one of the ways it's happening. So I just sort of wanted to take a moment, for the holidays, to remind our users to make sure that their less security-savvy friends, everyone they know who isn't listening to this podcast, does take software updating seriously and follows through when Flash is saying it needs to be updated. Don't read the license agreement. No one does.

Leo: Just say okay.

Steve: Just say okay. Yes. I would not be upset by the license agreement. No one in the history of man has ever read that once. Not even the first time.

Leo: I think we should make a new geek thing. When you go home for the holidays, you fix everybody's, you update everybody's systems. You lock it down. In fact, I'm going to do this. I'm going to go to Thanksgiving tomorrow at my dad's. I'll probably go over to his system, make sure he's updated recently, that all the software is up to date. This would be a good thing. Geeks Home for the Holidays. We'll have to come up with a catchy phrase for this.

Steve: And in fact it may not even be required that our listeners remember to do so because probably the moment they walk in the door, their aunt or their uncle or mom or dad is like, oh, John, printing stopped.

Leo: It's not working, Stevie.

Steve: Exactly.

Leo: Well, it's true. Every time I go back East to visit Mom, we set aside at least a day for - she does it, for tech support. It's like, okay, remember, one day we're going to get everything working that's stopped working over the last year. Geeks Home for the Holidays. We need an acronym.

Steve: That's spring cleaning. Winter cleaning.

Leo: Geek Update for the Holidays. Something. We'll come up with something. Question 5 comes to us from Philip Smith in Lafayette, California, just around the corner apiece. He says he's not happy with iOS security: It's not as safe as you imply, the approval process that's part of the app publishing requirement. I know you understand that, but I think it's time to remind our community, just because it's a closed system does not imply innate security. He refers us to a post about this on ThreatPost.com. Love the show. Cheers, Phil. Let's pull up that post here. But I think that I've said this many times, that there is the presumption, well, Apple's checking every app on the App Store, so they must all be safe. And of course there's no way you can know with 100 percent certainty, you're not even getting the source, that this app is doing everything you think it's doing.

Steve: Well, and it's more than that, Leo. And this is why I wanted to entertain this question from Philip. It is impossible. And I don't mean in the sense of Apple couldn't be doing a better job, I mean in the sense of we want an impossible thing. We want our computers to do what we mean, not what we say. It is absolutely the case.

Look at, for example, the clipboard. The clipboard is a massively convenient feature because it was originally maybe designed within an application, you would mark a clause or a paragraph, for example, and then cut or copy it, and then put your cursor somewhere else in the same app, and then paste it. So it made it very convenient to move things around within an app. But then it was realized, wait a minute, let's make that a global resource, that is, the clipboard, so that we can do inter-app cut, copy, and paste. Whoa, and that's really handy. I mean, who isn't copying URLs out of one place and dropping it somewhere else, putting it in notes, putting it in a browser and so forth. And now that we've got super-complex passwords, we're likely using the clipboard in the same way.

The problem is that it's a global resource, and malware has access to the clipboard just as our apps do. So there's a perfect example of something which is a feature which, because it's a global feature of the system, it becomes useful, that is, it's much more useful to us if we can use it for inter-app movement of data. But if we leave sensitive data on the clipboard, and this happens all the time, and it has been widely exploited by malware, then the malware is able to access the clipboard and get whatever we may have happened to leave there. Now, whose fault is that?

Leo: It's the clipboard's fault. Clippie. Clippie, protect me.

Steve: I knew Clippie was going to come into this. So it is, I mean, that's just - I'm sure there are plenty of other examples. But that's a clean one of where something that is a feature that we all want and use and would be very annoyed if we didn't have, has the potential for abuse. Not because there's anything wrong with it, but because it's a feature that can be misused. Now, if we look at iOS, here's a system which is highly locked down. And, for example, there is deliberately less inter-app flow.

Leo: They do a lot of sandboxing.

Steve: There's a lot of sandboxing. And people chafe at the sandboxing, that it's not easy to move something from one place to another. But that's also protecting us in the same way that not having a global clipboard protects you, and here's the danger of having a global clipboard we were just talking about. So, I mean, I don't know that I've ever implied that Apple's iOS security is ultimate protection. I certainly know that it's not. And in fact I think we have a question a little bit later about how Charlie Miller got himself recently blacklisted for a year because he was wanting to show Apple that they had a problem that they weren't acknowledging. But I guess my point is that we want power at no cost. And unfortunately, even if everything is working right, I think power comes at a price. And iOS being more of a consumer platform, you're getting functionality and you're getting more security than if you had more flexibility. But, yeah, I don't think Apple can do a perfect job. It's impossible for them to know what every app is going to do all the way down. We talked about the sandboxing and how useful it is for you to declare to Apple the things you want your app to do, and we'll be talking about that a little bit shortly.

Leo: I look forward to talking about Charlie Miller because I think that this is a ridiculous response to a security researcher coming up with a flaw.

Steve: Oh, it is a totally wrong response.

Leo: Yeah. It's, oh, we don't like it. Well, we'll talk about it in a second. But first Jason Pritchard in Las Vegas because he's wanting to know about monitoring the Internet. He says: Steve and Leo, you've talked in the past about organizations like AV companies, research facilities and such, monitoring Internet traffic seemingly for statistical analysis, maybe to see how a virus moves. Knowing the routers send packets to the specified destination, and switches only send packets to the port containing the device to which the packet is addressed - that's the difference, I guess, between routers and switches - how would one monitor anything except the traffic destined for the monitoring device? I know that some broadcast traffic would be detected, but even that should be limited to the network segment in which the device resides. How do research organizations gather information about network traffic? Are they doing it from an Internet backbone? If so, why would the owners of those connections let anyone near them? Thanks for all the great information throughout the years. Jason Pritchard.

Steve: I thought that was a great question. And we've never, in all of our years together, addressed it directly. Okay. So there's two ways. First of all, an organization like Symantec is massive. I mean, they're doing - they've got their security research guys. But there's all this other stuff they're doing, and operations, and payables, and receivables, and email, and everybody's on the 'Net, and they're surfing and Googling and browsing. So a large organization only has to monitor itself in order to have a huge beautiful cross-section of spam coming in, and threatening email [indiscernible] and what people are - where people are going and what's happening on the Internet. So just an organization looking - the security side of an organization looking at its own, considering its own navel, if you will, pondering its own navel, gives it enough. I mean, a huge cross-section of information.

Then the other thing that these organizations do - and it's actually something that I have done in the past. I don't monitor it all the time. But, for example, I myself have a block of 64 IP addresses that are completely unused, unallocated. They exist out there in the 4 billion IPv4 space, yet they have never been associated with anything. And that's just a big honeypot. There is traffic on those 64 IPs for no good reason whatsoever. It's not bound for me. It's not bound for anybody.

Leo: Oh, that's interesting.

Steve: It's just stuff out there. It's weird how much traffic there is on this space that never belonged to anybody. It's just things out there. It's like Code Red and Nimda still existing in a closet somewhere out there. Every so often a Code Red or a Nimda packet comes in.

Leo: Hello. Are you there? May I infect you, please?

Steve: That's just what it sounds like, Leo.

Leo: I'm lonely. Haven't been able to infect a PC in years. Still hoping. Are you running Windows ME? Are you?

Steve: And of course it has been said, and it is still true, that if you put an unpatched Windows machine on a raw Internet connection, it won't be long until it is found.

Leo: Now, I wonder if that's apocryphal these days. I mean, I wonder if that's still the case? We should try it.

Steve: It's still the case.

Leo: Is it?

Steve: It really is, yeah.

Leo: Wow. And that's those worms that self-propagate, and they're just endemic on the Internet. They're like herpes. They just float around.

Steve: Never get rid of it.

Leo: Never get rid of it. Terry Zinger in Dover, Ohio writes: I'm unsure whether you can help. Here goes. I recently noticed in my Norton AV 11.x for the Mac some interesting entries: ARP Cache Poisoning - incoming. It listed a MAC address I didn't recognize. Further investigation indicated the MAC address was for my smartphone. I have some tech background with the U.S. Navy, 30 years; another 20 years in tech with the educators. I retired as the Tech Director for a small school district in Ohio. I'd never heard of ARP poisoning until Norton started to report it.

Question: Can we stop this attack? After lengthy research - my phone's attacking me! After lengthy research I did the following: changed the wireless router at home to reflect the latest security. I realize this is probably closing the barn door after the cows are halfway to Japan. But can I change the MAC address on the phone, then block the old MAC address? Is there a clean solution? I assume this attack did not necessarily take place here at home. I use wireless access on the phone whenever possible to help save battery. I hope you can find a little time to help an old veteran and well-used techie. Thanks, Terry A. Zinger. Hmm.

Steve: Okay. So...

Leo: We've talked about ARP poisoning, I know, many times.

Steve: We have. And my best guess is this is a false positive, which is very possible for the following reason. First of all, a little quick review over on ARP is it's the protocol which is - that is, ARP formatted protocol packets on Ethernet. It's the protocol which is used to bind an Ethernet MAC address, which is the way the packets actually are addressed on the physical Ethernet wire or Ethernet air, to an IP address, which is an entirely unrelated addressing scheme, the way packets are addressed out on the Internet, because the Ethernet and the Internet are completely separate things.

So, for example, Internet IP addresses, as we know, are 32 bits; whereas Ethernet addresses are 48 bits, 24 being the vendor ID and 24 being a unique number within that vendor ID. So there needs to be some way to give an endpoint on the Ethernet that always will have a unique, globally unique MAC address to give it an IP address. And ARP is the way that's done. You can have static IP assignments where the device knows its IP. And so when somebody else on the Internet - I'm sorry, on the Ethernet, I mean the local Ethernet, says, "Hey, who has this IP address?" the device that does will say, "I do." And it responds with its MAC address so that the requester is able to then send IP-oriented traffic to the proper hardware. So that's the IP-to-MAC mapping.

However, most people do not use static IP addressing. They use dynamic IP addressing, using the protocol we've also talked about, DHCP, Dynamic Host Configuration Protocol. In that case, a device which is being asked to connect to the network, like your smartphone would when you bring it into the house - and as Terry said, he tends to use his WiFi wherever possible. So he's got it set up so that his smartphone will get on his local network. With DHCP you are asking the DHCP server, which is almost always someone's router, which also has a DHCP server function built in, it is saying, "Hi there. I'm being asked to use DHCP. I need an IP address." And so the DHCP server looks at its table of available IPs and assigns one which is currently available. And technically they have a lease time, that is, the lease expires, and then it needs to be refreshed.

So now we have the case of Norton AV 11.x running on the Mac. It apparently has a feature where it's going to warn users of ARP cache poisoning. To do that, it would have to be monitoring all the traffic on the network all the time, and essentially see and monitor, build its own table of IP-to-MAC address associations or mappings. And what probably happened, what almost certainly happened, is for some reason that table got out of sync. And it's not hard to imagine that it might have. Maybe the machine was turned off and then back on. Or it was briefly unplugged when some ARP traffic changed the network's awareness of these mappings, but this one Mac machine didn't see that. Anything could happen that could cause this to be desynchronized because there is no good way, there's no perfect way to prevent that desynchronization. Essentially, this is not a feature I am a big fan of, for exactly this reason. You don't want to upset people with false positives of something like this when it's so possible that they would occur.

So one way or another, the Mac ARP table, the Norton AV's ARP table, was no longer synchronized with the DHCP table that is residing in the router. So the smartphone walked in the front door in somebody's pocket, asked for an IP address, and the DHCP server gave it one, which Norton AV had as allocated to somebody else. And that is exactly what ARP poisoning would look like, was if there was a conflict between known and agreed-upon IP addresses and MAC addresses. But with DHCP, you're giving out new IPs all the time. So in a statically assigned environment, where everything has a fixed IP, you wouldn't ever expect there to be a false positive.

But when IPs are floating around, especially with a smartphone coming and going, something else might have received an IP. Maybe the smartphone came back in and tried to use the IP because its lease had not expired, tried to reuse the IP that something else had been given in the meantime. I mean, there's all kinds of scenarios where you could see a collision that would generate a false positive. And I'll bet that's what's happened. So Terry, take a big deep breath and relax a lot. I mean, sure, ARP poisoning does exist. Maybe your phone got some malware installed. But given the way the world is, I'd bet that wasn't the case.

Leo: I want a new bumper sticker: ARP Happens.

Steve: It does.

Leo: Yes, it does. Keegan Ead in Tempe, Arizona is a bit confused about Apple's OS sandboxing. I hope his mouth is not as open as yours was. Should not it be the operating system's responsibility to protect itself from viruses or malware? This sort of behavior seems to be what Apple is suggesting is appropriate, that the average user should not need to acquire third-party security packages to keep their computers at baseline. I think we're all in agreement on that. So why are the developers involved at all? Thanks, Steve. You put on a great show. I am not security or IT, but you manage to keep the program easily accessible for anyone. Thanks again. Keegan Ead.

Steve: So I thought maybe I should simplify this whole thing for anyone else who is a little confused by this because it is, in detail level, it is/can be confusing. But it's simple. Operating systems are responsible for protecting themselves and applications. Yet we've just been talking...

Leo: They just don't do a very good job.

Steve: Well, exactly as I was saying in the example of the clipboard, the clipboard represents not a defect, but a feature, which can be abused. So a perfect example would be if the application did not need the clipboard. If it didn't use the clipboard ever, then it could enhance the security for everyone by declaring that right off the bat. When it starts up, it says "I do not use the clipboard." Then the OS could remove clipboard access rights from that process. And the beauty would be, then, that if that process ever did misbehave, if it got infected, or it was acting wrongly and tried to use the clipboard or any other feature similar which it had previously declared it had no use for, the operating system would block it, and that's a good thing.

So it makes absolute sense for - I love what Apple's doing, that they have this notion, the notion of entitlements. So the idea would be that clipboard access would be an entitlement defined by Microsoft. The programmer could say, I either need it, I need that entitlement, or I don't. In which case the program would not be entitled to access the clipboard. And if all programs that didn't use things they didn't need declared themselves to be nonusers, security would be a lot better. So I think it's a great thing.

Leo: Bruce Harrison, Auckland, New Zealand, wonders whether Skype breaks TCP/IP: Steve and Leo, whilst - I love it, "whilst." Whilst listening to your description of TCP/IP and how it works, I couldn't help but wonder how Skype and other streaming - I'm not talking like a Kiwi, unfortunately - and other streaming technologies work. Does TCP/IP meticulously ensure that packets are resent and assembled into order whilst Skype disdainfully discards them? This person is a wordsmith. Deeply appreciate your weekly efforts. Bruce.

Steve: Well, not only is he a wordsmith, he's very smart.

Leo: Really.

Steve: Because this is a great question. It's something that we've sort of talked about tangentially, but not, again, never addressed directly. And that is, we've looked now at some detail in three separate podcasts, building upon the prior ones about How the Internet Works, we've looked at TCP. And we recognize that it sequentially numbers its bytes, it buffers them as they're being sent, as packets of bytes are being sent on behalf of the applications, until acknowledgment is received from the other end that everything that it has sent has been received, in which case it's free to let go of them. But if packets are lost along the way, acknowledgments aren't received, it will retransmit lost packets. It does all this work for us in order to give us a so-called perfect, a reliable as opposed to a not reliable connection.

But Skype is just wanting to send audio. And what happens with packets being dropped or reordered or anything? What does Skype do? Well, the beauty of Skype is it doesn't use TCP because that would be a big problem, exactly as Bruce has surmised. Skype uses UDP. You and I, Leo, are talking over UDP protocol rather than TCP for exactly this reason. Skype excels at forgiving us, forgiving the Internet for dropping packets. Yet it doesn't worry about a packet dropped a minute ago. We've already moved past that. It sort of fills in, literally, truly fills in the audio as best it can, interpolating the audio of missed packets. So Skype keeps them small so that packets don't carry too much audio because it recognizes they may get lost. But, if so, it doesn't care. So Skype and all other streaming services like this, real-time streaming services, do not use TCP because they don't want TCP in the way, mucking things up, which is what it would do if it stalled the connection and waited for, like, lost packets to get resent before it could move forward again. Instead, UDP being a non-connection-oriented, non-reliable protocol, they just go off, and if they get there, good. And if not, oh, well, that moment passed, and we pretty much understand what we're saying to each other anyway.

Leo: I think that's why UDP was created; right? I mean, that's basically the purpose of UDP.

Steve: I think it probably predated any sort of real-time streaming. These geniuses who put this all together just sort of said, well, we probably need one of these, and we need one of those. And they were right.

Leo: Sometimes you don't care. Sometimes you don't want to error correct because that'll just slow you down.

Steve: Like with DNS, for example, which runs on top of UDP, and you're responsible, I mean, the beauty is there's all that overhead of, like, SYN, SYN/ACK, ACK and all that, all that overhead associated with creating reliability in a packet-based network. Sometimes your request is so short: What is the IP for GRC.com? That can go in a tiny packet. And the answer is even smaller. Actually it's not because it contains the answer, and DNS contains the query as part of it, so it gets a little bit bigger. But the point is there you don't want to bother with all that overhead and handshaking and byte-numbering and everything. You just want to say "Tell me this real quick," and DNS says "Here you go." And so these geniuses realized sometimes we don't need reliability. If we don't get the answer, we'll ask again. But most of the time we will, and we don't want - so we don't want all that overhead for something that just doesn't need it.

Leo: It does have a checksum or something, so you know the packet's correct. I mean, it's not that there's no error checking. It's just that there's no resending.

Steve: Correct.

Leo: Bruce Harrison - oh, that was Bruce. Sorry, Bruce. Now it's Robert Hickman's turn. He is not our last question, but he's our last real question because right after him we're going to set up for next week. Robert in Bristol, U.K., voices his concerns about single sign-on, or OneID solutions: From the sound of it, OneID is going to be yet another closed-down, proprietary, siloed system. And if that's the case, I'm not touching it. Like the Internet as a whole, any kind of widespread authentication system must be open and not tied to any single provider to be trustworthy. We do have a single sign-on system; right? We have OpenID.

Steve: Well, and he's talking about OneID.com...

Leo: Which we talked about last week.

Steve: ...which we talked about last week that came out of the privacy conference. And I have to say, much as I hope something succeeds, I kind of have the same feeling. I mean, I like the concept of it. But I like, maybe, I hope, fingers crossed, what we're about to talk about next week even more. Which leads us into Tom Jones.

Leo: Yay. By the way, Web7064 in our chatroom says he could tell us a UDP joke, but we might not get it.

Steve: Yes.

Leo: And if you get that, then you got it. Tom Jones in Europe points us to a more promising alternative to OneID.com. Well, thank you, Tom. And it will be, as you say, next week's in-depth show topic: Steve, please take a look at Mozilla's initiative BrowserID.org that is, in my humble opinion, a much superior concept to OneID.com, at least in the following five ways: It's not centralized, a requirement for wide success on the web. It doesn't depend at all on Mozilla or any single vendor. Privacy: Vendor is not informed of visits to third parties. It's not vaporware because you can actually use it right now. And it's free forever. Bonus interest points for Security Now!: It basically uses crypto to solve a hard problem in a demonstrably clever way. He passes along a link to WebFWD.org. There's a 12-minute video there. Or go to Identity.Mozilla.com to learn more about it. Tom from Europe. Steve has created a short Bit.ly link to the video which is bit.ly/b-id.

Steve: Short for BrowserID. If any of our listeners are curious, this is a cool little MP4 video file, 12 minutes long, that sort of - it's a little slide show, gives people sort of a simple, rough overview. It does not in any way preempt the podcast that we'll do next week's in-depth look at how this works. It does a great job, though, of selling its benefits. And this has been on my radar for some time. I've been intending to get to it.

We're going to get to it next week. We'll all have digested our Thanksgiving dinners, those of us who celebrate Thanksgiving, and be ready to tackle some nice technology.

I'm excited about this for all the reasons that Tom mentions. And I have to agree with him, too, that any solution to this problem that attempts to be proprietary, to be owned by one organization, as any of these things do, suffers from exactly that. I think that that's a liability for something that sort of really does need to be open. And the Mozilla guys have really done a nice job. So next week we're going to do a propellerhead episode, looking in detail at a very cool, I mean really cool, completely out of your hair sort of solution for this. I can't wait to talk about it.

Leo: Good. That sounds exciting. So we'll do this next week. We do it every week. Usually we do it, as we did this week, at 11:00 a.m. Pacific on Wednesdays, that's 2:00 p.m. Eastern time, 1900 UTC at TWiT.tv. So watch live, if you can. If you're at work you can tell the boss this is research. But you can always listen or watch after the fact. We make audio and video available on our site, TWiT.tv. Steve does a little extra thing. He makes a 16Kb version available at GRC.com. That's for the bandwidth impaired. He also has full transcripts, which are nice. Again, as we mentioned earlier, because of Thanksgiving, Elaine's going to get a day to eat turkey, so we'll probably get the transcript up on Friday, I guess, Steve; right? Or Saturday.

Steve: Yeah.

Leo: But you'll find that at GRC.com, along with all the other great stuff Steve does, including his bread and butter, SpinRite, the world's best, finest, must-have, hard drive maintenance and recovery utility. GRC, Gibson Research Corporation, dotcom. You can also follow Steve on the Twitter. He doesn't follow you. He won't. He's following zero people.

Steve: Yeah.

Leo: He doesn't want to hear from you. No, you can, actually, if you ask Steve on Twitter, he sees his "@" replies. So it's @SGgrc.

Steve: So you "mention" me, as it's called, and I see them. And I would again encourage people to check my Twitter feed, even if you are not following me or not a Twitter user - you can just go to Twitter.com/SGgrc - because during the week I do tweet useful and interesting things often. And so you can sort of stay current that way, as well.

Leo: That's a good thing to do. I like that.

Steve: And let's remind everybody about the - put their thinking caps on about snippets and segments from past Security Now! podcasts that they think would bear repeating for whatever reason.

Leo: Yes, yes. TWiT.tv/bestof to contribute those because we want to do a "best of" so that Steve and I do not have to work the week after Christmas. We want to take one week off a year. That's it. Just one. That's all we ask. Hey, Steve, thank you so much. Great job, as always. I look forward to next week. I'll see you right here on Security Now!.