Listener Feedback #139

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-344.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-344-lq.mp3

SHOW TEASE: It's time for Security Now!. We've got a lot to talk about. Not just coffee, not just Vitamin D, not just health, but even some security news, the new Microsoft updates for Patch Tuesday. We'll talk a lot about SPDY and answer 13 questions from you, our listeners. Security Now! is next.

Leo Laporte: It's time for Security Now! with Steve Gibson, Episode 344, recorded March 14th, Pi Day, 2012: Your questions, Steve's answers, #139.

It's time for Security Now!, to protect yourself online. Mr. Steve Gibson is here, our Explainer in Chief. This is Episode 344, and we thought we only really would have enough security news for about a dozen shows.

Steve Gibson: Yup.

Leo: Were we wrong. Hey, Steve.

Steve: Hey, Leo. It's great to be with you again, as always. And you know, normally before we begin, I ask you if you're recording. And I didn't ask you this time, but...

Leo: I am recording, sir. I haven't forgotten to record this show in ages.

Steve: And actually, except for the live viewers, everyone listening already knows the answer to that question, when you think about it.

Leo: That's true. That's true, because it's recorded.

Steve: Because they wouldn't be hearing it if the answer were not yes.

Leo: But the good news is I am. You know, it's funny you asked that because a couple of weeks ago we did forget to record Game On!, which is a very expensive, elaborate show with expensive hosts and expensive production values. And the whole thing was not recorded. Oh, boy. My heart broke.

Steve: And you, when we connected initially here, you said "Happy Pi Day."

Leo: Yeah. And it's not just because I have a pecan pie in my hot little hands. It is actually because this is 3.14 that we're recording on. Actually, 3.1415 - in 2015 will be really the true Pi Day, won't it.

Steve: Ooh.

Leo: Ooh. This is only 3.1412. But three years will be the real one.

Steve: And I guess way back in 3.14159 something...

Leo: But did they even know about pi?

Steve: They probably weren't aware of the significance of that date.

Leo: They didn't know. They actually do celebrate Pi Day now in schools. And before the show my daughter Abby, who's now 19, came in, and she - in fifth grade they had a contest, in fifth grade, so what was that, eight years ago, for who could memorize pi to the most places. And she just ripped it off about, I don't know, it was about 25, 30 places. She says, "I'm not sure about the last few places." The one who won knew it to 50 places, five zero places.

Steve: Yeah, we had a guy in high school who would annoy us.

Leo: It's annoying. It's annoying.

Steve: The first time it's fascinating. Second time it's like, okay, yeah, Rick, we have already heard you do that.

Leo: Well, that's why we have Pi Day. Once a year, on March 14th, you get to do it. So Happy Pi Day. And you know what the best part about Pi Day is, we all get to have pie. And in fact there's a new pie shop around the corner, the All American Pie Company.

Steve: Before you eat that, Leo, before you eat that, you may want to listen to some of my news.

Leo: Oh dear, oh dear, oh dear. Well, we are having pie for lunch, and the entire Twit Brick House will have hundreds of pies. But maybe you'd better tell us why. Anyway, we've got news, and this is also a question-and-answer episode. So we have questions for you from our audience.

Steve: We got a bunch. I've got 13, which is more than usual because some are just little announcements and updates and things. So some of them are going to go quick. But I just kept running across interesting tidbits that I thought, oh, I have to share that. Oh, I have to share that. So we'll do a lot of sharing.

Leo: Well, let's get started. Let's get going.

Steve: Okay.

Leo: Is there anything big happening in security this morning?

Steve: Is anyone sponsoring this show, by the way?

Leo: Yes, Ford. The Ford Motor Company.

Steve: We'll talk about them in a little bit?

Leo: We've got time. We've got time.

Steve: Okay. So first of all, I got so carried away with the technology of SPDY that I failed to update myself on the deployment of SPDY. And so of course I was deluged with tweets from people who said, "Hey, Steve, thanks for the info. Now I know what it is that my Chrome browser has had since last April."

Leo: Aha.

Steve: It's like, uh, oh, that's nice.

Leo: Well, we talked a little bit about that, about enabling it and everything.

Steve: Well, yeah. And during the show we saw that it was there. But in fact it has been, and Google has deployed it throughout their entire server farm. I saw something, but it was a bit dated, that talked about how maps might not yet be using SPDY. But all the other regular Google services do. There is a funky "chrome://" URL you can put in which will show your SPDY sessions. And sure enough, if you go poke around Google anything for a while, and then go open a tab and put this SPDY sessions URL in [chrome://net-internals/#events&q=type:SPDY_SESSION%20is:active], you see an enumeration of all of the connections which were SPDY-enabled between your browser and Google. And just last week Twitter added SPDY support to their site.

Also Firefox 11, has just been released, and it has SPDY in it, but it's not enabled by default. So people who are staying current with Firefox - I'm famously not, I'm just staying where I am, back on 3.something. But people who are getting 11, I'm sure somewhere in there - oh, in fact, I've got it in my notes a little bit later is where you go. You've got to go into the about:config and find it. But you can turn SPDY support on. Let's see, what else do I have? Oh, it will be enabled by default in Firefox 13. So they're putting it in 11, but not turning it on, to sort of step into this gently. The Kindle Silk browser in the Kindle Fire has been using SPDY all along.

Leo: Was that the speed-up stuff in Silk? Or was there other stuff?

Steve: There's other stuff, too. They're explicitly doing their own stuff. But when they were looking for performance, they thought, hey, let's use SPDY connections between the Amazon backend and our Kindle Fire. So that's there. Also there's a mod-spdy module for Apache that's been around since December of last year. So only for about four months, but still it exists.

Now, it's worth noting, though, that to say you have SPDY and to actually be getting the advantages are two different things. So I don't want people to get too excited, nor too disappointed if they turn it on and don't see things speed up, the point being that, if you think about what we learned last week, using SPDY, that is, leveraging SPDY requires much more than just changing the protocol and announcing at each end that, oh, we're going to use a SPDY connection. And this is why I'm a little suspicious of, for example, the depth of implementation that Apache may have gone to. I'm not meaning to cast any aspersions on Apache. Maybe they really did reengineer their server around SPDY.

But my point is, that's what it kind of takes. You need to add a whole bunch of server-side intelligence to back up the support of the protocol in order to, for example, provide - to look at the page you're sending out and then to provide client hints of the resources that the client is probably going to want, or to automatically be sending down to the client in advance using the server-push features the things you know the client is going to ask for. So just saying that, oh, yes, we're SPDY-enabled, doesn't necessarily mean that you're going to get the same kind of performance that Google's testing did. And you know on Google's benchmarking they would have taken advantage of all that the SPDY protocol had to offer because that's of course what they were trying to measure.

So my guess is that we're going to see - and first of all, I'm super happy that this is happening because this just represents the evolution of the web. Also it's worth noting that Google's efforts at deploying this are succeeding. I have seen people on the 'Net

saying, wait a minute, we already have it, why aren't we hearing more about it? So of course with this podcast we're beginning to help make that happen. But the next version of the HTTP protocol - remember that we started off with /1.0. Then we went to HTTP/1.1. Well, 2.0 is in the works. And it is probably going to incorporate SPDY as part of the protocol. So that moves us into mainstream standardization, which is great. I just don't see a downside to it at all.

Leo: That's great. That's great.

Steve: So I wanted to catch everybody up on that.

Leo: There's the latest on SPDY. You know, we probably should mention, before we go any farther...

Steve: Oh, yes, I meant to, yes, yes, yes.

Leo: For the last 36 hours, should be gone by now, if you went to any site that had TWiT.tv in the domain name, you'd get this warning. This is the Chrome warning, but there's also similar warnings from Firefox, Safari. If you had antimalware software you might have gotten a similar warning from NOD32 and so forth.

Steve: Even a Google search turned up a warning link in your results.

Leo: Google uses a service, there are actually a number of services that track websites. I think Google does their own thing, but there are other bad software/malware databases. The warning, for those of you listening, says "Something's not right here. TWiT.tv contains malware. Your computer might catch a virus if you visit this site. Google has found malicious software may be installed on your computer if you proceed." You can proceed, or you can go back. They say, "We've already notified TWiT.tv we found malware on the site." No. The notification comes from about the eight millionth person who sees this and says - tweets me and emails me. And by the way, I'm glad you do, thank you.

And we worked on this quite a bit. This happens about once a month. And at first I was about to blame code on our site or a problem, something badly done on our site, or our site wasn't fully secured. And then I had a little - and I want to run this by you. I had a little come to Jesus meeting with Bear, Mike Taylor, who's one of the best sysadmins in the world, a part-timer for us, although his part-time job is often full-time, as it was yesterday. Bear and Chris Dieterle, who works with him, as well, immediately went in, they rooted out the malware, they modified the code that was the exploit. Often what happens is you have to figure out, well, where did the exploit happen, and get an updated module.

Steve: How'd it get in.

Leo: Yeah, how'd they get in. So there's two things you need to find out: how'd they get in, and what did they leave.

Steve: And what did it do.

Leo: Yeah, exactly. Now, in this case we were able to get rid of the malware very quickly. It takes a little longer for Google to clear this. They only update this warning every 24 hours. But there is a window of opportunity. If you fix it fast enough, you don't get in this. But because - what happened is we found it, we fixed it, and then inadvertently we reenabled it, the malware. And we thought we'd fixed it and there was a miscommunication, and somebody on my staff turned it back on. And so it ran all yesterday, Monday, what is it, this is Wednesday, so it would have run all Monday night.

Steve: And then got picked up.

Leo: And then Google put up a more serious warning that is harder to clear, takes 24 hours to clear. So that's why the warning persisted for so long. It is gone now. But the malware was removed very quickly, and the exploit I believe has been patched. So I asked Bear, I said, "Well, Bear, this is not acceptable. Who do I fire? Who do I blame for this?" He said, "Leo, this happens all the time." He said, "You and every other site is hacked constantly. What usually happens is the minute we get word, usually from somebody saying, hey, there's a malware alert on your site, we'll go in, we'll fix it, problem solved, move on. Because of the failure here, it lasted longer than it normally does, so you became aware of it."

He said, "But we're doing this at least monthly, and it's not just your site." And Bear works on a lot of very well-known sites, has friends that work on other very well-known sites. He said, "The bigger your site, the more often this happens. And anytime a site is updated in any way, it's likely that you're going to get another exploit." I said, "Well, what's the problem?" He said, "PHP is the problem." The nature of PHP, you remember in the good old days of CGI scripts you had a special directory that was specially permissioned that code could run out of, and nowhere else could it run. It can only run from the web server.

PHP is designed to run and be executed from any directory, any time. So if somebody can modify your file system, they can put a PHP script in an arbitrary folder and then point their browser to it and execute it. That script can then install malware. So that's very commonly how it gets exploited. And it's in the nature of PHP. And he said, "There's no one to fire here. You can blame us all for not getting this cleared sooner, but we do this all the time." I said, "What?" He said, "Every site is being hacked all the time. And there's not much you can do about it except be very proactive about scanning it on a regular basis." He said, "Most big sites have at least one, if not more, full-time employees whose only job is to look for file system modifications, be checking the log, constantly vigilant against this. It's the only way to prevent it." Now, you are in a special case, Steve, because you've written all the code on your site. I bet you it's CGI scripts. It's all...

Steve: No, actually, it's all assembly language, and it's pre- and post-server filters and

back ends. So it's actually a single monolithic DLL so that I don't have the overhead of loading and unloading and stopping and starting CGI.

Leo: So that's even more secure. For somebody to be able to access your file system on your server, they'd have to find a hole in IIS, I would guess.

Steve: And there have been some. But I have something - I have my own web filter which is upstream of IIS, and I scrutinize everything coming in through that, also. So, yeah.

Leo: Right. You're much more safe because you're not using off-the-shelf software that people know. You wrote your own.

Steve: I was going to say, and I have a much less exciting site than you do.

Leo: Well, and that's what Bear said. Bear said the reason this is happening more is because you're more popular. He said - I want to name names. Well, I'll give you an example. A friend of his - I don't know if I should say this, anyway, a very, very, very, very well-known site - does this for them. And he said they're hacked daily. And there's not - and even though that this site has ample resources, let's put it this way, the resources of the entire federal government, they still get hacked all the time. It's kind of the nature of the beast.

I was ready - I said, "But no, wait a minute. You can secure a site, can't you?" He said, "No, all you can do is constantly monitor it." So, and people are saying, well, is it because you've taunted Anonymous? No, has nothing to do with it. It's not, was not probably targeted at our site. It was some scanner that's running all the time; right? And there's many of them. I'm sure there's thousands of scanners running on the Internet constantly that are looking for these kinds of exploits. So the minute a modification is made to your site that opens a hole, you're going to get exploited, period.

So in any event, it's probably not something that was targeted at us. It's just this is part of doing business on the Internet. A public server is always going to be attacked all the time. Our code is modified quite a bit, but it is in fact most of it commonly well-known code in PHP. And so it's a full-time job just to secure it. And in fact this happens all the time. Now, the good news is, well, first of all, I think it's great that these browser alerts are happening because it prevents people from going into sites that have been compromised. And all the browsers do that now. You should, if you see it, as much as you might want to watch TWiT, you should absolutely back out.

Steve: Oh, I'll bet you that not a single one of our Security Now! listeners is going to say, oh, yeah, I don't care that there's an infection there. I'm just going to go through anyway.

Leo: Well, a number, yeah, a number of our - I have heard from a number of people who did. The good news is the exploit was a very old Java exploit, and I'm sure that

everybody who listens to any of our shows knows enough to keep their system up to date. So it's highly unlikely that anybody had any malware executed on their system. However, you should, as always, you should be proactive, scan it. And really, seriously, I understand. If you see this on any site, but I understand if you see it on our site, it's fine to email me, tweet me and let me know, and not go in. Please, back out. It will be cleared as soon as we've cleared the malware. It does take, unfortunately, if you don't do it right away, it takes longer to get that cleared, 24 hours to get that cleared. But it is cleared now. If you're still getting it, just restart your browser, you shouldn't see it anymore. And I apologize. But apparently there's no one to blame. It's just the way of the web. Does that sound right to you, Steve? Do you believe it's possible to fully secure a site?

Steve: Yes, I do. It's just math.

Leo: That's what I thought.

Steve: Yeah, it is possible. It's probably not convenient. If you restricted the execution rights of the directories for scripting, then you wouldn't be able to load things in default directories. I mean, you wouldn't be able to be nearly as casual as it's convenient to be.

Leo: Well, you secure your site, so we know it's possible.

Steve: Yeah. And I would argue it is...

Leo: Some of your security I would say is through obscurity because you're not using commonly well-known code.

Steve: Well, that's not obscure. It's just smart.

Leo: Well, but if your code were published on the 'Net, it'd be more likely to be a problem. But the way you've structured it, of course, makes it much less likely.

Steve: Well, but I mean, in defense of your approach, you would never have the site you have if it were up to me. So you would never be hacked, but you wouldn't have all the features you have.

Leo: Right, right.

Steve: And it would take about three lifetimes for me to implement everything you have in a way that would make me happy.

Leo: Well, we'd have to custom program everything, of course.

Steve: For example, you're pulling libraries, third-party libraries from their servers on the fly into the pages of the visitors, of the people who are visiting you. Not you alone, I mean, that's common practice now. I see this, and when I see it I just cringe because I see some site that is loading JavaScript on my pages from a URL of some other domain. And I think, well, okay, yeah, it's convenient for them. The theory is that, when that library is updated, they're automatically updated. The problem of course is, when that library is compromised, everybody using that is compromised. So, I mean, what's happened is we've gone for convenience over security. And this show is all about how those two, there's a constant tension between convenience and security. And the fact is, to do it right would end up meaning that it was never done at all. And that wouldn't work for you either.

Leo: So one of the questions I had is, well, why is it that we get this malware warning and nobody else does? And he says, well, there's a number of reasons. First of all, if you have a full-time person doing this, you can clear this before the malware alerts pop up. Second of all, some big, most big sites have an inside line to Google and an inside line to the malware sites, the stop adware sites, and can - see, Google does not, despite what it says there, notify us. The way we get notified is by the malware alert popping up, and then we act upon it. Most other sites will get an internal - this is what I'm told - an internal notification, hey, you've got a problem. They'll fix it. And that's why you don't see these.

The truth is, and we knew this about banking, banks are hacked all the time. It's bad business to talk about it. Sites are hacked all the time. It's bad business to talk about it. In fact, it's probably a mistake for me to talk about it because it just attracts attention and more hacks. The best thing to do, if you can, is act as if you're never hacked, you're a hundred percent secure, and just don't tell anybody. But that's not, as you know, how I operate.

Steve: Well, and it doesn't work in our model, where we have such vigilant listeners and viewers, and they're tweeting, hey, what's happened here?

Leo: Right. So one of the things that we are going to do, and I think we should have done this, but we couldn't afford to, is we are actually hiring - I'm hoping it's going to be Bear. We're going to extend him an offer. But we're hiring a full-time sysadmin to monitor this at all times. It's a fairly high cost. People like Bear are not cheap because they're really good. But it costs us money. We lost considerable audience yesterday. People didn't watch live, and we lost considerable ad revenue, as a result. So we can't afford to have these - not to mention it's embarrassing.

So there's the story. I do feel like - Bear is in the chatroom, if people have questions for him or want to get more clarification on this. But I do feel like this is something that's a little bit the dirty little secret of the Internet. And I was, when Bear told me this yesterday, I was flabbergasted. I thought that we were doing something wrong, that this was something we could fix. And he said, well, it's just the cost of doing business on the 'Net. Unless you're Steve Gibson.

Steve: Yeah, well, again, or unless you expend a phenomenal amount of resources in order to keep it from happening. And unfortunately - for example, PHP, it's very nice and very convenient and horribly insecure when it's not very carefully deployed and managed. So it's a powerful tool. But with that power comes responsibility. And, I mean, the whole model, I mean, we've talked about this. The idea, for example, that someone can send through a forum an SQL backend database command which will be executed by the server when it delivers that page, I mean, that's insane. It's insane that it was ever allowed to be done that way. Why was it? Because it was convenient for the people who were implementing it.

So there are major decisions which have been made which were absolutely wrong, by policy, not by mistake. And this was, for the longest time, this was my argument with Microsoft was they had insecure policies that were causing their problems, like having services that were enabled and running that no one needed. That was dumb. And the consequence, we had all these worms for some period of time. Finally they turned - they put a firewall in Windows and turned it on by default, and all of that problem just went away, bang, oh, it's a miracle. No, they finally fixed their policy.

So the problem is we're still in a position where convenience is trumping security. And the idea that your own database would execute commands that your website visitors gave it, that's just nuts. But the architecture enforces, I mean, it encourages that almost. And similarly, the idea that somebody could put a PHP script on your server, which your server would then execute, that's nuts. I mean, that's just crazy to accept executable commands from a passerby. But the fundamental architecture says, oh, look, here's PHP. I'm supposed to run that because I recognize the extension on that file. Just lunacy. But that's the way these sites are built now. And they're just not secure by design. They're secure by constant vigilance, which is exactly what Bear is talking about. And it's too bad that that's the state of the art, but that's where we are today.

Leo: And people want to blame Drupal or our web designers, and blame PHP. And certainly you already expressed the case. But it is just the way it is. I mean, it's not Drupal's fault. It's not PHP's fault particularly. I agree it was kind of a crazy way to do it. I long for the days of locked-down CGI script folders. But, oh, well. And we are making somewhat of a mistake by talking about it because it does attract attention. And that brings more attempts to hack you. And it's funny, Bear...

Steve: Well, but also, to the degree that we have other webmasters listening who are thinking, hmm, maybe I need to give some better thought to the security side of this...

Leo: My point exactly.

Steve: The problem, yeah, the problem is that people who are building these libraries, they're doing what they need to. The problem is the systems are just not secure by default.

Leo: Right. And Bear tells me that every time we do this, we talk about these kinds of things, that the server logs show a real spike in attacks. So there are absolutely people listening who see this as a challenge. And Bear just said in the chatroom, he says, "I cringe every time you do this." But he understands that that's what we do,

and that's what we need to do. And that's, I think, one of the reasons Bear likes working for us. So we will, I'm sure, have a full-time security czar on the site as soon as we can do that. Moving on.

Steve: So last week we moved the podcast from Wednesday to Tuesday to make room for Wednesday. And I just wanted to touch briefly on the fact that my two first third-generation iPads are sitting in Ontario, California, at the FedEx depot, patiently waiting for Friday morning, when they'll be loaded on...

Leo: Me, too. Me, too.

Steve: They'll be loaded on a FedEx van and...

Leo: Mine's right next to yours in Ontario.

Steve: Oh, no kidding. Is that where it is?

Leo: Yeah.

Steve: Oh, okay.

Leo: I've heard from people who have them in Nashville, but that's the FedEx hub. And Ontario, it's almost like the town was - you and I know, because we went there for Podcast Expo a few years ago.

Steve: Yes.

Leo: It's the town that was built to be a hub. It's nothing but big truck depots.

Steve: Well, yeah, and it's the Ontario Airport surrounded by huge warehousing from which everything spreads. And Ontario Airport is a perfect place for freight planes to be coming and going because pretty much nobody else is coming and going there.

Leo: Right, right, right, it's great, I love it. So we'll get it on Friday.

Steve: And someone said, someone tweeted, "Steve, I didn't think you were a sheep," or something to that effect. And as if I'd bought into this. And I've been very clear from the beginning, first of all, I love the iPad. I think it's - it's my portable platform of choice. I've got a Fire. I've played with it. It just doesn't have nearly the fit and finish, but of course it's one third the price. For me it's the screen. That retina screen is all I want, 2048x1536, and I'm done. I will be absolutely happy to have that. So it's going to be my

little portable pad with that screen. One will live in the car, and one will live in the house. And I'm just going to be a happy camper. Do you know, Leo, by the way, I'm still grandfathered into the original unlimited AT&T data plan. But I've wondered, as they move to 4G and LTE...

Leo: Yeah, they'll find a way to get you off of that plan. I am, too.

Steve: Yeah.

Leo: And what they said, which is interesting, is that they are not throttling the iPad. Because you know the way they're handling the, quote, "unlimited" data on the iPhone is if you go over 1.5 or 2GB, forget it. You go down to edge speed. But apparently they're not doing that on the iPad. There is a reason to go with Verizon, though. Verizon has more LTE. I don't know about where you live, but all...

Steve: I'm a Verizon customer. It was only the fact that the pad was only available on AT&T at the beginning.

Leo: Well, that's I'm sure why AT&T keeps this unlimited thing grandfathered in. They're hoping not to lose you because Verizon has a more compelling offer, and they're going to allow hotspotting. Now, for a lot of people, having a hotspot on your iPad may not be an important selling point. But the fact is AT&T is not going to allow it, so.

Steve: Right.

Leo: Yeah, as far as I know, they're continuing to do it. But I can only imagine there's a strategic strike force in the executive offices at AT&T, trying to figure out, how can we get people off this unlimited plan?

Steve: Now, you don't know when yours is arriving, so we don't know if there's going to be a TWiT unboxing live video.

Leo: There will be. So here's the deal. Mine says, as yours does, "Before 3:00 p.m."

Steve: Okay.

Leo: I hope they're not lying because I know that that sometimes does not happen. We have scheduled - we moved iPad Today from Thursday to Friday at 4:00 p.m. So with any luck - it's going to be very embarrassing if we're sitting there at iPad Today, and no iPads have arrived. But we have a total, I think, of four or six ordered. I'm hoping at least one of them will arrive by 4:00 p.m. Friday so we can have an unboxing on iPad Today.

Steve: And we have heard that the screen is unbelievable, haven't we?

Leo: Yeah. Ryan Block at Engadget was at the event, showed us the screen. And even on Skype you could tell the difference. When he zoomed in, you could tell the crispness. There have been videos, there's a video from Vietnam that is almost certainly legitimate of an unboxing. But again, I think, I don't - it's pure speculation because I haven't seen it. But it's my sense that when you see this, you'll say, boy, it feels like you're looking at real objects.

Steve: Not pixels.

Leo: Not pixels, yeah. So but we'll see. We'll find out. Estimated delivery - ooh, look at this one. That must be an error. That has to be a mistake. Estimated delivery March 15th.

Steve: What?

Leo: They do this every time, okay? I've just got to tell you, they do this every time, where then - so it says March 15th, that's tomorrow, by 3:00 p.m. Then they will have a little thing that says "Held at request of shipper." But they haven't said that yet on this one. It is at Ontario, next to yours. Got there this morning. Wouldn't that be nice?

Steve: Oh. Anyway, we'll have it, and I'm done then. I am a happy camper. I don't need anything else ever again.

Leo: Oh, yeah, until the next time.

Steve: I don't know.

Leo: I have a feeling this is - would you use this in lieu of a Kindle?

Steve: That will be the big question because I'm reading with my DX, my big Kindle, twice a day when I leave the house to grab a bite of food. I mean, I love the big form factor. I found a white one which they no longer make on eBay that had barely been used. I gave it to Mom this last Christmas. The prior Christmas I gave her the previous small one. And so she called that her "little friend," and this is now her "big friend."

Leo: "Say hello to my little friend."

Steve: And she, too, likes it. I really like it. And so...

Leo: Is your mom a fan of "Scarface"?

Steve: Mom's been around. So anyway, so I'll want to see whether the crispness of the text on this so-called "retina display" moves me from the convenience of the reflective Kindle display. That's one of the things I want to see, is this what I switch to for all of my book reading? So, don't know.

Leo: Is the DX screen - how big is it? That's eight inches; right? Eight and a half inches, something like that?

Steve: No, it's actually - it's a little more. It's longer than the pad is tall, and a little narrower. So it's a little more non-square aspect ratio. But I just - it's very nice having a big page of text. And one of the restaurants I eat at sometimes turns the lights down too low, which is annoying because of course I need light for the Kindle. So I used the pad the other day, and it was fine. But it's just the Kindle is lighter and sort of just easier to - I prefer it for reading books. But the pad is for everything else. We'll see if that changes after I have this amazing, amazing screen.

So we're just past the second Tuesday of the month, and we had a very noisy Microsoft update cycle because Microsoft was running around flapping their arms, warning everyone about the problem that we've already talked about a couple weeks ago that I kind of yawned about. Quoting from Microsoft's own blog post, they said:

"Hello. Today we're releasing six security bulletins - one critical-class, four important and one moderate - addressing seven issues in Microsoft Windows, Visual Studio, and Expression Design. We recommend that customers focus on MS12-020, our sole critical-class bulletin, as the March deployment priority. Here's a little more MS12-020:

"This bulletin addresses one critical-class issue and one moderate-class issue in Remote Desktop Protocol (RDP). Both issues were cooperatively disclosed to Microsoft, and we know of no active exploitation in the wild. The critical-class issue applies to a fairly specific subset of systems, those running RDP" - the Remote Desktop - "and is less problematic for those systems with Network Level Authentication (NLA) enabled. That said, we strongly recommend that customers examine and prepare to apply this bulletin as soon as possible. The critical-class issue could allow a would-be attacker to achieve remote code execution on a machine running RDP," which they mention is a non-default configuration. "If the machine does not have NLA enabled, the attacker would not require authentication for access."

And then, finally, elsewhere, Microsoft was quoted saying - or I'm quoting them saying: "This issue is potentially reachable over the network by an attacker before authentication is required. RDP is commonly allowed through firewalls due to its utility. The service runs in kernel-mode as 'system' by default on nearly all platforms.... During our investigation, we determined that this vulnerability is directly exploitable for code execution. Developing a working exploit will not be trivial. We would be surprised to see one developed in the next few days. However, we expect" - and this is why everyone's running around with their hair on fire. Microsoft said, "We expect to see working exploit code developed within the next 30 days."

Okay. So first of all, the reason I had already discounted this when we discussed this, this has been a known vulnerability for a while. I said, okay, first of all, Remote Desktop

is not enabled by default. Remote Assistance is enabled, but that's not the same thing. So this is not a problem with the Remote Assistance. This is with Remote Desktop, which is not on by default. Even if you turned it on you're still, well, let's see. Probably it punches a hole through your Windows firewall because it knows that it needs to be able to receive incoming connections on port 3389. So that's the default port for Remote Desktop is 3389. But I can't imagine anybody doesn't have now a SOHO router, a small office/home office router. And that's definitely going to protect you from any unsolicited incoming probes to port 3389.

That is to say, even if you've got all your machines at home with Remote Desktop enabled, it's only if an outbound connection is initiated on that port that incoming traffic would be allowed to come back through, which would not be the case because outbound connections actually would be initiated to that port on someone else's machine, not from your 3389 port. So there just isn't a vulnerability unless you explicitly have your world set up because you roam around out in the world, and you want to be able to access your computer's Remote Desktop remotely. That's the danger.

But this network-level awareness is an additional protocol which has been implemented since Vista. It's not available in XP. I think it can be turned on in Service Pack 3. So in XP SP3, they added it, but it's not on. There is on Microsoft's site a quick-fix button that allows you to turn that on for XP. It's in the UI of Vista and Windows 7 and Windows 8. So turning that on enforces a level of authentication prior to the exploit being able to function. The problem we have now is that it's possible to exploit the vulnerability prior to authenticating with Remote Desktop, which is really bad.

So I don't - I'm glad Microsoft has fixed this now. I don't expect it to be a huge problem. But for any people who know that they deliberately mapped port 3389 through their NAT router in order to get to their desktop when they're out roaming around, those are the people, if they don't have this NLA enabled, they're at risk. But all you have to do is apply today's patches, bring your system up to current, and you're fine, too.

For corporate installations that cannot, for whatever reason, deploy a patch instantly, Microsoft does have a Fixit button which allows you to turn on NLA to temporarily bring up a barrier. If you still need, if you desperately need to have access to Remote Desktop in the meantime, and are unable to apply the patch, turning on this network-level authentication, that solves the problem, too. So as always, install your Windows patches promptly, and you'll be okay.

And no doubt in the next few weeks we'll have some - someone's going to try to do a worm or an attack or something. A worm is potentially a concern because we know that the default port is 3389. So you just scan the Internet for anything that accepts a connection on 3389. That's more than likely going to be Windows Remote Desktop. And once you figure out what Microsoft did and changed, you'll be able to figure out the exploit, which is no doubt what some hackers are working on right now, to find people who don't install patches, who aren't listening to this podcast, who for some reason do have Remote Desktop enabled, and they'd like to crawl into their computers. Unfortunately, this gives them a way to do that.

We recently had the sixth annual CanSecWest's Pwn2Own contest. We've had fun talking about this in years past. And what was embarrassing was that Chrome was immediately brought to its knees. It took minutes, and an exploit was developed that escaped the much-vaunted sandbox of Chrome. So the fact that Chrome is sandboxed, we've said that's good, it makes things much more difficult to exploit, but we can't count on it perfectly. And this was an example. There was an exploit that was developed. However, it's worth mentioning that it apparently used a vulnerability in Chrome's version of Flash.

So once again, Flash and the complexity of Flash was the underlying problem.

But the good news is, and I salute Google for this, they had it fixed within 24 hours. The moment that they understood what the problem was, they were able to push out a patch and fix Chrome quickly. So this is a little bit like the model we were just talking about with your website, Leo, and websites in general that are so complex that there are going to be ways to get in. But if that's the case, the best thing you can do is watch them carefully and fix them quickly. And that's really the on-the-fly patching model that Chrome from the beginning has adopted, and of course we're seeing other people, Mozilla is talking about being much better about that, as well. Speaking of which, there was a Pwn2Own vulnerability found in Firefox 11, just before its deployment, which has just happened. It turns out once they saw what it was, they already knew about it, they were in the process of fixing it, so it's fixed. So it's there.

Oh, and I also have had a note that both - relative back to SPDY again, I think I touched on it, but there is something called SPDY Indicator, available both for Chrome and for Firefox, that is, Firefox 11 and on, which supports SPDY. And it puts a little green bolt of lightning in your address bar, just to indicate, as pure information, when you're visiting a site that is supporting the SPDY protocol. So for people who want to play with SPDY, you need to enable it, as I said, in Firefox 11. It'll be on by default, the plan is, in 13. But it is, it's a Firefox add-on, so go search for SPDY Indicator. You can find an add-on for Firefox, and there exists also one for Chrome.

And following up on the disgrace of GoDaddy, this sort of came across my radar, and I thought it was interesting. Michelle Paulson, who is legal counsel, blogged on March 9th, so just a few days ago, she said: "After months of deliberation and a complicated transfer, the Wikimedia Foundation" - as in Wikipedia - "the Wikimedia Foundation domain portfolio has been successfully transferred [away] from GoDaddy.... The portfolio transfer was formally completed on Friday, March 9th, 2012. The transfers were done seamlessly, and our sites did not experience any interruption of service or other issues during the procedure.

"As the provider of the fifth most visited web properties in the world, the Foundation" - that is, the Wikimedia Foundation - "cares deeply about who handles our domain names. We had been deliberating a move [away] from GoDaddy for some time our legal department felt the company was not the best fit for our domain needs and we began actively seeking other domain management providers in December 2011. GoDaddy's initial support for the Stop Online Piracy Act (SOPA), the controversial anti-piracy legislation in the U.S. House of Representatives, reaffirmed our decision to end the relationship."

Leo: Yay.

Steve: Yay again, yes.

Leo: That was just inappropriate.

Steve: Wikipedia. And I think it's good that they saw that, and good that everybody else saw that. It's like, learn from that. So little quickies from the Twitterverse. Clifford Williams tweeted as @thegdot, don't know what that is. But he said, "@SGgrc, the Cherokee web server already has support for SPDY." I meant to check out what the

Cherokee web server was, what platforms it runs on and so forth, but didn't get around to it. But if anyone knows what that is, there's news there. And that was before I learned that not only that, but Apache has support of some kind. And of course Google has it widely deployed across their site.

And, oh, @Guysmiley777, he tweeted, "Love my AeroPress. And you're right, a consistent burr grinder makes a HUGE difference." He says, "I was shocked at the difference." And so I did want to let people know, I created a shortcut for anyone who's curious. I think we've noted before that our coffee discussion, which was pre...

Leo: Huge, huge.

Steve: ...podcast recording a few weeks ago, has made it to YouTube. You can just put in, I think, "Steve Gibson coffee," and you can find it. But I put in also - I made a bit.ly shortcut that was explicit. So it's bit.ly/SGcoffee. So SGcoffee, a bit.ly shortcut, will take you directly to it. And it's just a nice little - I think it's 18 minutes of you and me, Leo, talking about our passion for all things caffeine.

Leo: And I blame you because I just bought the most ridiculous burr grinder in the history of mankind.

Steve: Yay.

Leo: Well, I had one before. I had a Krups. And they're, like, 20 bucks at Costco.

Steve: Wait, burr grinder for 20 bucks? I'm surprised, Leo.

Leo: Well, I know. So I went online, and I ordered the KitchenAid Pro Line series burr coffee mill. This thing is - it's like a Briggs & Stratton engine attached to a burr grinder.

Steve: Wait, and that's not the one I got? Because I did get a KitchenAid...

Leo: Well, maybe it is the one you've got.

Steve: ...Pro Line.

Leo: Yeah, well, this is it.

Steve: Does it have, like, a big black knob in the front?

Leo: Yeah, it looks like a sausage grinder. It's the same one you got. And it's your fault, at 158 bucks.

Steve: That's the one, yep.

Leo: I blame you, Steve. But it's good coffee. I had to have some this morning, and it's amazing. And then the other thing I got is a coffee vault. Because you buy a pound of beans, but you don't grind them. And so as long as we're talking coffee, I bought this thing, it's called a...

Steve: We're doing it again, aren't we.

Leo: ...a Friis coffee vault. And it's sealed, but it has something interesting. It has a CO2 exchanger in the lid that you change every month. They say in the materials that, yes, you want to keep it in a cool, dry place. You want to limit exposure to air. But you also want to vent CO2 that's coming off the coffee. So this thing has a CO2 vent.

Steve: It is true that Starbucks bean bags, when you buy pounds...

Leo: They have a little vent.

Steve: There's a bellybutton on the bag, and that's what it's for, it's to vent the CO2.

Leo: Yeah. So this is a little canister that I can buy a pound at a time and seal it. And then - it's your fault. I blame you.

Steve: I'm happy to take responsibility for you having a burr grinder, Leo.

Leo: Pretty damn good coffee.

Steve: I'm not feeling sorry for you. Sorry about that.

Leo: Pretty darn good, I'll tell you. That thing...

Steve: How many iPad 3s did you buy?

Leo: Yeah, you're right. 168 bucks for a burr grinder, big deal. I mean, the funny thing is the plunger is like 20 bucks, the AeroPress. So...

Steve: Right. Oh, and you're right, nothing slows that grinder down. Boy, you turn it on...

Leo: [Mimicking an engine revving]

Steve: Okay, now, when the announcement happened on Wednesday, Apple's site came up, and their store was down. And I knew that I wanted to get in instantly. They put up that placeholding page that said "The Apple Store is offline. We're making updates." Well, everyone in the universe knew what they were doing. They were getting ready to put the third-generation iPad on sale. So I was clicking Refresh for a while. But I'm thinking, how long am I going to be clicking Refresh?

So I thought, there's got to be a web page change detector. Now, I had one that I had used for years. It was a little standalone Windows app. And I tried to use it, but it wasn't happy with Apple's page. Whatever the technology was that the page was putting it through confused it. So it didn't work. So I thought, what about an add-on for Firefox, which is still my primary browser? Well, I found something fantastic that I wanted to share with our users for anything like this in the future because I was notified the moment, actually within 30 seconds of Apple's site switching to the commerce site coming back online, this thing told me, and I was able to get my purchase in before everybody else figured out that Apple's site was back up.

It's called "Check4Changes," aka C4C. Check4Changes, and actually the URL is addons.mozilla.org/en-US/firefox/addon/check4change. So there it's not plural. Maybe it is just Check4Change. Anyway, so what I like, it's very nice. It's a lightweight add-on for Firefox. You mark some text on the page that you want it to be looking at, and then you right-click on the tab and say "Start checking." You're able to choose any of five intervals, and those five are configurable in the options. And I had it, it was defaulted for 30 seconds, and I thought, well, that's often enough.

And then what it does is it changes the little icon on the tab to a countdown clock, just so you can see, and it goes 29, 28, 27, 26, and when it hits zero, it reloads the page and then checks the marked text to see if it's changed. And then you can have it play some alert sound if it detects a change. So I liked it because it's integrated into the browser. That's kind of where you want that kind of thing to be. It had no trouble with Apple's page because, of course, it is a browser, and it knows how to read pages; whereas this much more simple-minded, simple HTTP query thing that I had before just was lost in oblivion. It was probably getting 19 levels of indirection or something before it finally got to the, who knows, cookie exchanges and things before it was able to get there. And so this worked really nicely and I wanted to let people know.

I have a request from Consumer Reports. I did an interview with them last week, talking about Facebook privacy stuff. And they had a couple questions that could only be answered really by a Facebook developer, which I am not. So if we have in our audience anyone who knows how to develop for Facebook, who knows the ins and outs of the Facebook API for doing Facebook apps, I'm sure any such person would also be on Twitter, so please drop me a tweet at @SGgrc because I'd like to put you in touch with a producer at Consumer Reports who would like to put your name in lights and interview you.

Leo: Cool.

Steve: And I wanted to mention that Elaine, it turns out, our illustrious transcriber, also proofreads, and at a very reasonable price. Jenny, whom you've met, Leo, is a prolific writer, and she just finished a screenplay, and this is the second thing she's had Elaine proofread. And I can't quote the price, I don't know what it is, but we didn't even identify Jenny as Jenny to Elaine when she quoted because I didn't want to take advantage of Elaine in any way. And Jenny could not be happier with the results that she's had. So I wanted just to let our listeners know that, if they need something professionally proofread, there's someone who can do it, and that's Elaine. It's On-SiteMedia.com. I think that's what it is.

Okay. And lastly, well, not lastly. I'm still going. We're going to run out of time before we even get to the questions. I am more than halfway through Book 13 of the unending, never-ending Honor Harrington series. And I take back any frustration I ever had with it. Oh, my goodness. Twelve began to accelerate, and even though there was a sort of a quiet period, maybe like books six through nine or ten, oh, my goodness, David Weber - the good news is he's written nothing since, except there are ancillary books, but I'm not going to go into those because I have this other stuff I have to get to.

Leo: There are limits, man.

Steve: Yeah. But oh, my goodness, it has been worth everything. We have this incredibly well-defined, huge, mature, interesting galaxy, steeped in politics. And I happen to like politics. I like, I mean, these are - the people do realistic things. And I'm just - I couldn't be more pleased with this investment that I've made. So I did want to say that, boy, it's been worth the journey. I am loving where this has been going. And now that I'm all caught up, because he's been writing these things since 1984 or something, I mean, like forever, they go way back, it's been fun to see the terminology catch up with present time, too, because I've caught him using some terms that weren't in common use when the series began, so of course he couldn't use them back then.

Leo: Like cell phone? Fax machine? Automobile?

Steve: Yeah, it's been interesting. But anyway, oh, wow. And for a while I was annoyed that we weren't seeing more of Honor Harrington because I fell I love with her at the beginning. Now, I mean, and she stayed around, and other things have come in. Oh, my goodness. It's just fantastic. And the one thing I'll - no, I can't say anything more because I can't do a spoiler.

Leo: No spoilers. No spoilers.

Steve: But, oh, it's just - I'm clapping. That's me clapping.

Leo: Yay, yay.

Steve: Yay. It's really good.

Leo: It's not much of spoiler if you have to read 13 novels to get to it. But I think we should still...

Steve: Yeah, I'll wait a year. A year from now I'll let people know what it was because I'll figure by then you've either read them, or you've given up, or you got burned out.

Okay. So, Leo, I've mentioned to you a few times that I have been reading about carbohydrates. And I warned you about that pie on Pi Day.

Leo: Wait a minute. You're not going to say anything that makes it undesirable to eat this fine pecan pie, are you?

Steve: Well, we all know that I do experiments on myself. And I played with Vitamin D. Actually there's a lot you haven't heard about that I've done that someday we'll get around to doing a series of health podcasts. But I needed to share something with you. The reason I'm excited about getting through with this Honor Harrington series is that I've got four books stacked up on the science of carbohydrate metabolism that I really want to get to. But I read enough of them, you just can tell when something makes sense. And they made sense.

So five weeks ago I made a dramatic and deliberate change in my diet. I have eaten since then absolutely no manmade simple carbohydrates or starches, that is, no potatoes. And I felt self-conscious. For example, there's a vegetable soup that I like at this place that's got some potatoes in it. Well, I've never done this in my life, but I sort of, when I encounter one on my spoon, I'll push it off, out of the bowl, onto the plate that is next to it. And I feel like I'm being a picky eater. But, I mean, I've decided, if I was doing an experiment, I would be rigid about it.

So what does this mean? Well, this means beef, chicken, fish, shellfish, and salad, vegetables, essentially. It's all I've eaten. No bread, no grain, rice, no oat bran, no whole wheat, none of that. Friday of last week I did another round of blood tests. And I've done, I mean, they know me at the lab.

Leo: They're used to you. "Oh, what is it now, Steve?"

Steve: It's very funny because I show up early so that I'm the first person there on the earliest day that they open.

Leo: Right, because it's going to take a while.

Steve: The doors are normally locked. Anyway, I went up in the elevator with the phlebotomist, who once he finally turned to me, he said, "What's wrong with you?"

Leo: Do you like getting your blood drawn?

Steve: Anyway, so Friday was testing day after five weeks avoiding carbs. I mean, not avoiding, I mean absolutely, no exception, zero carbs. My bad cholesterol, my LDL, dropped nearly in half.

Leo: Wow.

Steve: It is at 54.5 percent of what it last was. And I know why. And I could go into the science, but this is not the podcast for that. But for what it's worth, I wanted to share with our listeners. There are people out there I would imagine in our community of listeners who feel that their cholesterol is too high. Maybe they're on statins to bring it down, which is unfortunate because statins have a lot of negative consequences. But for what it's worth, my own limited individualized personal experiment, I have never had my bad cholesterol, my LDL, this low. And it did nothing to lower my HDL, the high-density lipoprotein good cholesterol. It stayed exactly where it was. But LDL, just it crashed it. It just collapsed. So...

Leo: And I just want to give the usual disclaimer. Steve's not a physician. He's not making any recommendations. Please don't follow his recommendations. But it's funny because this conversation has been going on for over a year. Paul Thurrott read the same book that you've read, the "Good Calories, Bad Calories" book. But I have to say, the study came out, but did you see the Harvard study that came out this week?

Steve: Oh, financed by the wheat growers of America.

Leo: No, no. It was a long-term, longitudinal, 15-year study with over 100,000 men and women who were free of cardiovascular disease and cancer at baseline. And what it concluded was that there is absolutely no safe amount of red meat, period. That basically any red meat you eat increases your risk factors measurably.

Steve: And I can tell you why that study is nonsense.

Leo: Okay.

Steve: But I can't tell you now. No, I saw the study. I read the report extensively. And in fact the "Good Calories, Bad Calories" science writer, Gary Taubes, has a beautiful, clear explanation of the problems with that kind of study. That was a dietary questionnaire study. And the problem is those kinds of studies, you just have to take them with a grain of salt.

Leo: That's what the Meat Institute said, too, by the way. Just in case you're curious. There is an American Meat Institute, and they said that "relying on notoriously unreliable self-reporting about what was eaten and obtuse methods to apply statistical analysis to the data" is classically an error.

Steve: Yeah. There's a guy, a pseudo-scientist named Ancel Keys, who is responsible pretty much for this whole belief that saturated fat is bad for us.

Leo: Now, you could, by the way, I presume, do a no-carb diet that didn't have red meat and bacon and salami and hot dogs in it. You could eat other forms of protein.

Steve: Oh, and I'm not a big meat eater at all.

Leo: So it might be, based on this, it might be - but anyway, we shouldn't get into it.

Steve: I really like fish, and so...

Leo: Let's just point out that there are disagreements among people who do this for a living.

Steve: Here's what I would say, Leo, is because the other thing we are is all individual, it is vastly - one of the things I have learned in the seven years I've been studying nutrition as a background hobby, is the incredible variety that exists among people. So there are people who have sensitivities to one thing or the other that others don't.

Leo: Well, that's why a 15-year longitudinal study of 100,000 people is better than...

Steve: Tells you nothing about you.

Leo: Okay.

Steve: That's the key.

Leo: Right.

Steve: Nothing about you. So, and that's my big problem with the fact that doctors diagnose based on these huge studies...

Leo: Public health, yeah.

Steve: ...that are not about you. The thing they do ask, which is good, is what's your family history because now they're beginning to zero in on who you are. So the only thing I want to say is what happened to me was that. And it's simple for someone to try. It's just, I mean, I understand people have carbohydrate cravings. Well, there's a reason for that, too. But my...

Leo: Gary Taubes also points out that cholesterol is not necessarily a valuable indicator.

Steve: I agree with that, as a matter of fact.

Leo: But that's a long - now we're getting a really long story, so we won't go there. But he predicted, he does in that book predict exactly what, well, actually you got a remarkable result. But he does predict a surprising result, which is a lowering of cholesterol and lipid in blood tests.

Steve: Well, triglycerides.

Leo: Triglycerides, yeah.

Steve: Yes, because your liver turns glucose into triglycerides, and that lowers the size of the lipoproteins. And the tiny VLDL and LDL lipoproteins are much more prone to something called "glycation" and oxidation, both of which are bad things for your arteries.

Leo: But we will talk more about that some other time because we've got 13 questions and half an hour.

Steve: We're never going to get it done. I have to briefly say, somebody tweeted me, "Hey, Steve, I heard you talking about SpinRite 6.1. Should I wait to order SpinRite?" And, okay, well, the phrase "till the cows come home" may be generic or applies here. First of all, all SpinRite 6 owners are going to get 6.1 for free whenever I get around to doing it. There aren't any bugs that I know if in SpinRite. I just want to catch it up with things that have happened since I finished it in 2004.

And there's been a lot of changes since then, much stronger use of SATA, serial ATA, over parallel ATA. Much larger drives. BIOSes are becoming increasingly buggy, so SpinRite's dependence on the BIOS is some thing I want to remove. I want to build in Ultra DMA. We got that Western Digital hybrid drive that uses some flash and some storage. SpinRite already does the right thing, but it could do a better job with it. We've got the big sector drives, or maybe it's the WD big sector drives that use 4K sectors. Again, SpinRite works with those, but I could detect that and do a better job in terms of buffer sizing and to make it better.

So I'm going to update SpinRite. But waiting to purchase it makes no sense because it works fine now. I'm just going to make it work better. And I'm not starting on it now. There's only one of me, and right now I'm still finishing up this really interesting project on finding the longest repeating strings in files, which is part of the project of putting SpinRite testimonials on the site which I want to get done. Then I've got to get finished with the Off The Grid, I've got to finish those pages up. Then I want to finally get the cookie stuff published. And then I'm done. And then I'm going to start on SpinRite. But I move slowly. So if you think you need SpinRite, believe me, from a preventative maintenance standpoint, get it, and you'll still get 6.1 as soon as it's ready. So I just

wanted to make sure I didn't confuse people by saying, oh, something fantastic and new is coming along that I want to wait for. It's like, well, you'll get it anyway. And all it is is just sort of catching up with things that have happened in the last eight years.

Leo: Steve, I've got questions. Are you in the mood?

Steve: Well, I am. I'm a little worried about the time, though.

Leo: We'll go until we run out.

Steve: Yeah. Because, I mean, yeah, okay, we've given our listeners quite a podcast so far. Not quite the Q&A that we had planned, but go until you need to do the next podcast.

Leo: Till we've got to go. We've got This Week In Google at 1:00. We've got time.

Steve: Yup, okay.

Leo: Question 1, Stephen in North Yorkshire, U.K. He wonders if his use of TrueCrypt is fully secure: Steve, I have all my banking details for several accounts for myself and my wife in a single text file on a TrueCrypt-encrypted USB card, or chip. Whilst I realize that is safe, as it has a 26-character passphrase employing your suggestions, et cetera, once I OPEN the text file in Word or any other editor, am I storing the data unencrypted in various cache files and log files on my PC? If so, is there any way around this? That's a good question. Thanks to both you and Leo and his staff for the hard work you put into producing Security Now!. Stephen in North Yorkshire, U.K.

Steve: It's a great question.

Leo: Yeah.

Steve: And I often notice, for example, when I open a Word document, that Word opens up a temp file right in that same directory that's got some variant of that document in it. So there the document was written to the disk on the fly by the word processor.

Leo: Unencrypted.

Steve: Yes, exactly, unencrypted. So Stephen raises a great point. Here you've got the file encrypted with TrueCrypt, but the act of viewing the contents is causing it to be written to your drive. Now, what I would suggest is - first of all, the problem, of course, is we're not using secure operating systems. These started off as a toy OS on the desktop and never really lost that flavor. So there's nothing secure about the systems

we're using. We've added features here and there, trying to increase their security. But fundamentally they're not secure. So the one thing you could do would be to use a lightweight viewer, for example Notepad, which does only keep its contents in RAM. And I know that, for example, Notepad is a RAM-based viewer. It doesn't write anything to the disk.

The problem you still have, of course, is swapping because, if the RAM that Notepad was using happened to be swapped to your paging file, then that would go onto the hard disk for that reason. But normally that's only going to happen if you're heavily using your system, like you're actively paging RAM in and out. And probably, if you just use Notepad to open it up and copy, cut and paste or whatever you were doing, and closed it, the chances of Notepad being swapped out are minimal because that's the app that you're using, and then you're probably okay. But it is definitely, I thought, a very good point. There just isn't - there isn't a perfect solution that I know of. But having a really secure viewer is an interesting feature for a product that I kind of have in mind for the future. So I'll keep that in mind.

Leo: Yeah, anytime it writes the file, even though Word deletes a temp file when it closes it, it crashes, it doesn't delete it. And even...

Steve: It's still on your hard drive.

Leo: ...if it deletes it, it's still on the hard drive. In fact, you might even overwrite it, and then it'd be in slack space. I mean, it still exists. So as soon as it's written, you're in trouble. That's one of the reasons PGP has a secure viewer.

Steve: Ah, do they.

Leo: So, yeah. So when you unencrypt a PGP email message, you can have it open in a secure viewer that is stored in RAM, never writes to disk. And I think they overwrite the RAM. That's the whole idea, is that - for this reason. You can view it, but you view it securely.

Steve Coakley in Phoenix wonders about a site-to-site tracking blocker: Steve, I've noticed that, if I search for an item on Amazon or Google, and then leave and go to different sites, that ads for the exact item I was searching for keep following me and showing up everywhere I go. Horrors! That seems pretty creepy. Well, all right. It's not exactly like a doll chasing you. I wonder if there's a privacy setting to turn off that kind of tracking. I'm not even sure where to start looking, though. If I use an ad blocker, will the ads still be there but I just won't see them? I want them to stop altogether, even if I can't see them, says Steve.

Steve: Well, of course we stepped into this with our recent dialogues about tracking. I just wanted to make sure that Steve knew that probably just disabling third-party cookies will solve that problem, and definitely give it a try. All web browsers give you the option of turning off third-party cookies. And that probably solves the problem. So this is a perfect example of what third-party cookies enable, and browsers all let you turn them off. So that's all you have to do.

Leo: Question 3, Adam Fourney in Waterloo, Canada adds a bit more salt to his hash: Steve, I just heard your response regarding a question which discussed the issues with a database containing both hashes and salts, stored in the same database. In this case, I'm guessing that the primary reason the admins chose to use unique salts for each user is to thwart pre-computation attacks, i.e., rainbow tables. Each salt effectively gives a new hashing function. With this threat model, storing salts and hashes in the same table is no big deal. Generating a unique rainbow table for each user is at least as complex as a direct brute-force attack, says Adam Fourney, Ph.D. Candidate, Waterloo, Canada.

Steve: Right. I wanted to - several people wrote in about this, the idea of putting the salt along with the hash. And so I wanted to make sure that I didn't overly complicate this when I was talking about it before. Having a salt is way better than not having one because, if you simply used a well-known hashing function, then exactly as Adam mentions, and as we've discussed many times, existing tables for, like, SHA-1 and MD5, exist. And so it would be simple, if they could get a hold of the hash, to look up a matching password for that hash, for that hash function. Adding a salt, exactly as Adam says, dramatically increases the security because essentially you're scrambling, individually scrambling each instance of the use of the hash function per user. So that's a much better thing to do.

Now, storing them together is the question. And, yes, storing them together is not as secure as storing them apart. But so store them apart, if you can. But if you can't, it's better to have a salt than not to have one. So again, it's that traditional tradeoff between convenience and security. It's less convenient to have the hash somewhere, to have the salt somewhere else. You have to go get it and then use it to figure out if the password's correct or not. And you want to keep that out of the hackers' hands. Whereas, presumably, they could get to the hash. Of course you want to keep everything out of the hackers' hands, if you can. So anyway, I think that issue is settled.

Leo: Question 4, Sunil Joshi in Chicago, Illinois. He wonders about SSL keys bigger than 2048: During the feedback Q&A last time you mentioned that the longest RSA public key that had been factored was 768 bits, and that we currently use 1024 or 2048 bits for higher security. I understand that every one bit increases the complexity exponentially. However, this is true only until very advanced computational devices are invented. It is only with the current processing power that it will take an inhumanly long time to factor a 2048-bit key, or even a 1024-bit key.

But why not stop worrying about where the edge is in this cat-and-mouse game and generate SSL keys that are way bigger than 2048? What about 4096 or 8192? Isn't the key generation merely a computational process? What's stopping us from making the keys as long as possible? I am sure there is some concern or limitation I'm not aware of. Could you throw some light on it? And thanks for making the Security Now! show. I cannot wait for it every week. Sunil Joshi.

Steve: So, yeah, I wanted to make sure about this because I've seen this question also pop up a lot. It's true that computing the public key using a pair of hopefully very randomly generated primes - and we understand now that that's more easily said than apparently done in our industry. That's a one-time process, to create the public and private key pair. And so, sure, you could do that to any arbitrary bit length. The cost, though, is in every time it's used. We mentioned a couple weeks ago that it's about five

times more computationally expensive to use a 2048-bit key over a 1024-bit key. And that approximate ratio continues. So it would be about five times harder to use 4096, and five times harder than that to use 8192. So you start multiplying those fives, two fives gives you 25, and another one gives you 125, so now it's 125 times harder to use an 8192-bit key than a 1024.

In certain applications, that's just going to be a deal breaker. You've got a low-powered processor in a smartcard or a heavily loaded server that has now switched over to using SSL all the time. Imposing 125 times the connection overhead for negotiating an SSL dialogue, a handshake, that becomes a problem. And here's the point. Yeah, it does mean we never have to worry about security ever again. But what we've got is massive overkill. And it's massive overkill that we're paying for needlessly every single time we use it. So the fact is, today, 1024 is plenty strong. As far as we know, we haven't even come close to factoring that. And even the 768-bit factorization was much easier than 1024, and we are moving to 2048, which is radically harder to factor. So the point is we really, really have enough security, and there's just no reason to throw away all that computation time without getting any real security benefit in return. Once you've got, I mean, really enough security, anything more is just wasted time.

Leo: Moving along to Question 5 from Ron Kurr in New Hampshire. He wonders about determining the physical location of an IP address: Steve and Leo, on a recent Tech News Today they were discussing the lawsuit brought against the cloud-based TV service Aereo. One of Aereo's legal positions is that their services are restricted to New York City customers only. I'm a developer by trade and understand the basics of networking, and I don't ever recall any specification talking about embedding the physical location of the machine attached to an IP address. I'm assuming that Aereo isn't lying to the public. So my question is, how do they know that my IP comes from a computer in New York and not Tokyo? I appreciate the work on the show. Look forward to hearing your thoughts. Thanks, Ron.

Steve: That's a really good point. Okay. We know what IP addresses are. IP addresses are hierarchical in nature, that is, big ISPs get big chunks of IP space. For example, Level 3 has all of the 4-dot space. And HP had famously had, I think it was like 14-dot and 15-dot. They were really greedy because they were involved in the Internet in the beginning and had much more space than they were ever going to use. And the good people have been giving back IP space that they don't need. Many universities also got huge chunks of IP space that they never used. And they've been giving it back as our IP space is becoming more - as the IPv4 IP space is becoming more depleted.

So the point is that the IP is used for routing, and there have been attempts to geolocate based on IP. So, for example, if you're a Cox cable user you can maybe do reverse DNS on the IP and see what the DNS says, and that can give you a clue. Or there are just - there are forward indexes that say IPs in this range are in this region of Southern California. IPs in this range are in this region of Northern California. The point is they tend to be very inaccurate, at least the closer you try to get. I know that the block I have here in Southern California is identified by these various services as being well away from me, up in Northern California. So it's not very accurate. Sometimes they are; sometimes they're not.

Leo: All they really know is who your Internet Service Provider is.

Steve: Yes, that they could definitely determine. So if you had a regional ISP, then you could presume that all the IPs they own are going to be serviced within that region.

Leo: That might be easier in New York City because the primary ISP, Cablevision, only works in Manhattan, Long Island. It is a regional provider.

Steve: Right.

Leo: So, but do they say that it's IP addresses? I mean, I bet you they have credit card information, too, if these people are customers.

Steve: Well, and it is also the case, although Ron didn't ask, as I mentioned before, Reverse DNS often is a treasure trove. You ask for the DNS of the IP, and it'll say, like for example, on my cable modem is oc.oc.cox.net.

Leo: Oh, you're in Orange County.

Steve: Orange County, yeah, exactly.

Leo: So it's an imperfect science, at best.

Steve: Yup.

Leo: But more can be deduced than one might think.

Steve: That's true.

Leo: Charles Hill in Washington, D.C. observes that IPs can't replace hostnames. Dangit. In a recent episode you discussed what happens if DNS were to go down or be attacked. Some people suggest, well, just use the IP address to get somewhere. But you might want to mention that, for the millions of shared-hosting sites, that will not work.

Steve: So on behalf of Charles and all the people...

Leo: Everybody said that, huh?

Steve: ...our very astute listeners who said, uh, Steve, you can't. Remember that what they're talking about, and they're all right, is that many, many hosting providers have many fewer IPs than they have domains that they host. So when you look up, for example, MyOwnSite - I'm making that up - MyOwnSite.com, it may give you an IP that's

the same as YourOwnSite.com. So both of those domain names point to the same IP. The way the server disambiguates - I love it when I can use that word - disambiguates those two is that in the individual request, the individual web browsers asking for those two different domains, all make a TCP connection to the same IP address.

But in the request headers, in part of the URL there's a host header. And so one of the host headers will say MyOwnDomain.com, and the other one will say YourOwnDomain.com. So that tells the server which directory, essentially, to serve that request from. And the server's broken down into all the domains with different hostnames that share the same IP. So clearly, if we weren't using DNS anymore, if we were using IPs, that whole model of using the hostname to disambiguate the individual websites that are in a shared environment would not work.

Leo: But that's not - go ahead.

Steve: I was going to say, Charles and everybody else, tip of the hat to you. You're exactly right.

Leo: But that's not done through DNS. That's done at the hosting server, which disambiguates.

Steve: Correct.

Leo: So you could, I presume there would be a way to use the IP address and then a slash and then the hostname or something like that. There must be some way to signal to the server that you're looking for a particular site.

Steve: Yes. The only thing you could do, like on the spur of the moment, would be to edit your hosts file and basically make your own...

Leo: Ah, because the hosts file would pass along the information.

Steve: Yes, make your own little private DNS. Then you could put MyOwnDomain.com into your browser. It would look up the IP, and so it would make the correct connection to the correct IP. But it would think it was connecting to MyOwnDomain.com rather than an explicit IP. So it would put a hosts header in that would allow the remote shared hosting service to give you the correct website.

Leo: Clever. Just modify your hosts file.

Steve: Yeah.

Leo: Instead of entering the IP directly. Although most of the big pirate sites are not

shared hosting.

Steve: Correct.

Leo: So it's still going to work. And I think that was the context for that. No, actually the context was March 31st taking down the Internet, that's right, yeah.

Steve: Right.

Leo: Greg Williams in Brisbane, Australia offers a clarification about 2048-bit SSL: Asymmetric keys are only used during the negotiation, but not for the lifetime of the connection. That uses negotiated symmetric keys. Therefore, you only pay the performance penalty at the start, and the rest is not slowed down. Obviously, if HTTP pipelining isn't used - unfortunately it's disabled by default on almost every browser - there'll be a penalty for the multitude of request establishments, but it won't slow down the data transfer.

Steve: Yeah, I just wanted to toss that in in case anyone was confused about that. That is the case. And in fact, I'll go a little bit further and say that, thanks to SSL caching, you only need to negotiate, you only need to go through the public key negotiation the first time you're connecting to a remote server, no matter how many connections you then subsequently make as you're browsing around on that server because they will verify that they still have this SSL credential valid, and there's no need to redo it. So not only is your individual flow not slowed down, but subsequent connections are also not slowed down. It's really not that big a problem.

Leo: All right. Moving on to another question, this is #8. Bruce in Washington, D.C. says he's been thinking about SSL and WiFi tracking: Love the show. In all the recent discussion of third-party cookies, it occurred to me that your own public key would make a nice, nearly unique "cookie" for tracking purposes. Right? So isn't there a privacy tradeoff to HTTPS? In other words, when I have a secure connection, I'm kind of identifying myself. Also on the subject of tracking, we know that Google and others have mapped WiFi hotspots based upon their unique MAC addresses.

Can companies also do the reverse with us? For instance, smartphones and laptops can automatically search for WiFi hotspots. But when they do that, they share their MAC address; right? So Starbucks, for example, could keep track of my visits, even if I paid cash, based on my phone just kind of saying, hey, I'm here. They might even be able to track me as I walk around town, as I pass various Starbucks locations. I haven't heard of companies doing this, but I don't see why they couldn't. Thanks, Bruce. Is that possible?

Steve: So the first part of his question is not possible because he's got this backwards. It's the server that we connect to that provides its credentials to prove to us that it is the server we are intending to connect to. In a one-sided authentication, which is what that is, where the server is authenticating, we're not providing our credentials. It is possible for SSL to be used in a double-ended authentication, where the user would have a

certificate that is being used to authenticate itself to the server. But that's a special case, corporate environments, corporate networks and so forth. That's not the normal model, where we're anonymously connecting to an authenticated server. Part 2, he's absolutely right. It is the case...

Leo: Hmm, that's interesting.

Steve: Yeah. It is the case that all of our devices, our WiFi devices, have unique MAC addresses, and they're known to the hotspots wherever they're within range. So just as...

Leo: So do I have to have logged into that hotspot ever? Or do they just sense it?

Steve: They just sense it. And that's, I mean, when...

Leo: So that conversation is going on as I walk through town. My MAC address is being announced by my phone everywhere it sees an open - or an access point.

Steve: Yes, exactly.

Leo: Or is it just broadcasting it?

Steve: Yes. Essentially, in the same way that if you had...

Leo: But this is good. This is juicy. Wait a minute. You're saying, as I'm walking down the street, my phone is saying - giving out my unique, and it is unique, MAC address constantly.

Steve: Yes.

Leo: Well, now, why are people worried about third-party cookies? So then his supposition that Starbucks or anybody who had many locations - maybe this is why Starbucks has so many locations. I've often wondered why they'll have a Starbucks across the street.

Steve: Are they a front for the CIA.

Leo: They must be tracking us all. They've got a database to say, well, I can tell you who was downtown. Wow. So if you turn off the WiFi, of course that won't happen.

Steve: Well, and let's step back a little bit. Let's remember that our cell phones are

identifying where we are all the time anyway.

Leo: To the cell company.

Steve: With cell tower triangulation.

Leo: Right.

Steve: That's the way some of these location services work is "You are here." And it's like, oh, yes, I am.

Leo: And we know that all the major carriers, wireless carriers, have database portals for law enforcement, where law enforcement can go, using, what is it called, a pen warrant.

Steve: Yeah, and how many movies is this now in, where we all know that your cell phone is tracking you, and they'll send their final message, and then they'll smash the cell phone down and stomp on it in order to keep it from tracking them any further. Or toss it into the back of a garbage truck, and now they're going off and tracking the wrong car.

Leo: I'd be more worried about that than anything else.

Steve: But it is the case that MAC address is unique, and that's certainly trackable, too.

Leo: Bill in Michigan, who's a regular in our chatroom, we love him, shares some thoughts about the consequences of SPDY. He says: I'm getting bad vibes about SPDY. Here's the way I see it. The third-party nature of advertising sourcing and ad tracking, being sourced by third-party servers, keeps the main site and ad servers separated. But when pages start SPDYing up - if that word gets coined, remember when you heard it here first - who will want to go to the slow sites which link to offsite ads? The pressure will be on to source those ads from the host server's SPDY stream. Wouldn't this dramatically change the way things are done? First thing I thought was there would be no such thing as a third-party script, cookie or resources. They'd cease to exist. Everything is first party.

In the GRC newsgroups, Alan Cameron brilliantly came up with another loss, the old HOSTS file. Its effect will be meaningless. The problem is that "pressure is on" phenomenon to keep pages fast. Some new inventions or protocols will be needed so main servers and ad servers get their content in sync, so they can be just as fast. Perhaps this will be done with some sort of side-channel communication. This has actually always been a worry, but the status quo has held it back. Now SPDY's pressure may make it happen. Do you see what I see? That SPDY changes a lot more than web page speed? Your thoughts, please! Bill.

Steve: So what he's suggesting is something which has been discussed from time to time, and that is that here we're also worried about third-party things. All that a website would have to do in order to not have third-party things blocked is essentially funnel them through itself. That is, rather than providing a URL to a third-party asset, it would provide a URL in its own domain which, when the web browser turned around and asked for it, it would go and fetch that third-party resource and feed it back to the browser as if it was coming from that first-party domain.

So Bill's exactly right. There is a way to collapse this whole third-party deal. And Alan was right that right now people, for example, have DoubleClick.net blocked using their hosts file, but this would allow DoubleClick ads to sneak in as a first-party, as if the ad were being sourced and served by the server they were visiting. So that's the bad news.

The good news is third-party cookies don't work, either. That is, no one could track you if that was being done because your browser would just give back the first-party cookie. It would give back the cookie for the domain you're on, not the domain for the advertiser. And if we assume that advertisers desperately want to track us around the Internet, then they have to hold their third-party status in order to be able to provide us with a cookie unique to them, not unique to the site we're visiting. That's their whole deal. So I think that there's some back pressure on amalgamating everything through a single-party site. Probably we're not going to see it happen. And boy, what a pain it would be. You think you've got problems now, Leo, with your server. Wait till you start trying to pump other people's content through it and out to the browser. Ugh.

Leo: Such an exciting world we live in.

Steve: Ah, yes.

Leo: Question 10, Richard Covington in Redondo Beach, California. He used the subject "CURSE YOU, STEVE GIBSON!!!" to get your attention, and it worked: Hey, Steve. Well, now that I have your attention, I'd like to thank you for the work that you, Leo, and Elaine put in to such a great show. Additionally, and the reason for the note, I'd like to thank you for your decision to have Elaine make transcriptions of each episode. Having these transcriptions available has resulted in an invaluable resource, yet you have set the bar for all other netcasts extremely high. That's true, because it's expensive, and Steve does that out of his own pocket. Which we should talk about.

But anyway, unfortunately, none that I listen to have even come close to your exemplary transcriptions. However, because of these high-quality transcriptions, I've come to expect that I can go to any show that I've listened to in the car to either - and by the way, none of our other shows, either - that I've listened to in the car and either pick up a link or check out a subject that was discussed. To my immense displeasure, the information just isn't there. CURSE YOU, STEVE GIBSON for setting the bar so high! I'd like to extend my highest gratitude and "job well done," and look forward to future amazing netcasts. I've been a listener since Episode 1 and a proud owner of SpinRite. Even though I've been designing computers for over 30 years, I still find the information not only timely, but very interesting. Richard Covington.

Steve: So I just wanted to remind our listeners - first of all, thank you, Richard. I'm happy that we here have spoiled you from all other netcasts. And I want to remind our

listeners, because I often run across people sending questions or tweeting things that are discoverable instantly by going to GRC.com/sn, which is that Security Now! page, and we use Google-hosted search, and I pay Google annually for the privilege, and I'm paying them more because people are using it more. But thanks to the transcripts, remember that everything is searchable. The other day I didn't remember which episode it was where I explained - I did that little snippet on what SpinRite does. And so I just put into my own search term, "What does SpinRite do?" And bang, it was Episode 336. And it's like, oh, thank you, Steve. Now I've found it.

Leo: He's thanking himself.

Steve: I'm thanking myself. I'm recursive, Leo.

Leo: No, it is, it's a very good point. And it's one of the real advantages that the transcription gives you is that it makes everything in the show searchable. It's really a disadvantage of audio and video media is they're just not searchable unless you do that. And you know, it's expensive. We probably should do it for all our shows. You have set the bar. We don't currently, but maybe we will. We've looked at automated systems. They don't work very well.

Steve: No. And I would say, too, that this podcast probably more than others, we're laying down foundational, long-term stuff. And my feeling was it was always an archive we were building from the get-go, as opposed to, for example, Tech News Today. I'm not sure that the stuff you guys discuss, I mean, it would be nice to have it, but is it worth paying the price to have it.

Leo: We have three more, and we're going to do them pretty quick here because we're running a little bit late, but that's fine. Josh in Greenville, South Carolina says: I don't have the iLuv Anti-Glare Screen Protector for my new iPad. I'd like to get it. I love it on iPad 2. But they don't make it anymore. Have you found a similar product?

Steve: I love it, too. When people see my iPad, which has the antiglare screen, they just go, oh, that's so much better. And my only real serious pet peeve with Jobs and Apple is that they just go for this high-gloss screen. So I wanted to take the opportunity to remind our listeners that I have a separate Twitter feed, @SGpad. And it will be - it has been active in the last week. It will be active as I actually get the arrival of my pads. And I'll be tweeting some stuff. So anyone who's interested in following my pad stuff, I'm not going to clog my main @SGgrc stream with that, just over on @SGpad. Follow me if you're interested. And I will definitely find a replacement for the iLuv Anti-Glare Screen Protector. And I'll be tweeting the things that I find over at @SGpad.

Leo: It's odd that they stopped making that, actually. Let's see, here. Moving along to Question 12, our penultimate question. Dean Murray, Sydney, Australia, gives us a Tip of the Week. He says that you are very comprehensible when being played back at double speed, which you can do on many devices, including any iOS 5.1 device. I've been doing it this way for the last 100 or so episodes, and it's made my binge-listening approach much more productive. He said: I cannot even begin to

imagine doing that today with what we recently heard is 400-plus hours of content. So he's got to speed it up. By the way, Tom does sound a bit faster than Leo and Steve, plus I've been told by some Americans that Australians are fast talkers anyway. It's actually well known cognitive science that faster speech is more intelligible.

Steve: Yup.

Leo: So check on your device - iPhones, iPods, iPads all can play back podcasts at 1.5 or even 2x faster. Audible also supports that in their apps. Finally...

Steve: And if you are stuck in traffic, Leo, you can play it back slower in order to...

Leo: If you've got more time.

Steve: So you don't run out of podcast before you get to where you're going.

Leo: That's what I've found. We debate how long shows should be. We're an hour and 55 minutes into this show. And I have found, I have learned that people don't care how long it is as long as it's not shorter than their commute. They want it to cover the commute. That's my supposition, anyway.

Jim Michael in St. Louis, Missouri with our last question: Have you heard of buffer bloat? He found a YouTube video talking about it. He said he thought it was quite interesting. We actually talked about it with Bram Cohen, the inventor of BitTorrent. And it's a big issue for BitTorrent Live because buffer bloat really can up latency on real-time stuff, including Skype. He says, I understand much of what the presenter says on the video. There's some things I don't get, and I would love to have it "Gibsonized" for us mere mortals, or at least debunked if it's not real. Anyway, I thought I'd bring it up here in the hope you could explain this information. Thanks for the great podcasts.

Steve: And that's next week's topic.

Leo: Yay. Yay.

Steve: We're going to go into what buffer bloat is, where the problem came from, how well-intended designers didn't actually understand, unfortunately, the TCP protocol and the problems it is creating for us, why you can actually end up getting much less performance than your connection can provide right there at home. That's next week on Security Now!.

Leo: Yeah, I had no idea until Bram described it. And Vint Cerf has weighed in

against it. I mean, it's a big issue.

Steve: Yup. And what I love about it, Leo, is we've already paved all the foundation. All of our How the Internet Works series discussing TCP performance and slow start and throttling, everything we need to know in order to add this next bit of subtlety is in place. So I think our listeners are going to get a good kick out of it next week.

Leo: Good. Oh, I can't wait. Because I kind of understood the issue, and I'm wondering - unfortunately the stats about what buffer sizes are on most routers is not published.

Steve: No.

Leo: So we need to cut through this. Steve Gibson is at GRC.com. That's where you'll find his great stuff, including SpinRite, the world's best hard drive maintenance and recovery utility, and of course all his free security programs. And this show. He has 16Kb versions available, as we mentioned; transcriptions, as well, a great search feature. Go to GRC.com. And if you've got a question for future episodes - we do Q&A every other episode - GRC.com/feedback has a form just for you. That's the preferred way to communicate with Mr. G.

Follow him on Twitter, @SGgrc, and a little iPad activity there at @SGpad on Twitter. And we'll be back next week to talk about many things, including buffer bloat. We do this show every Wednesday, 11:00 a.m. Pacific, 2:00 p.m. Eastern time at TWiT.tv. You can watch live, but you can always download it. 16Kb versions from Steve in audio, but we have audio and video available in higher quality formats at TWiT.tv. Well done, Steve. Bravo.

Steve: Thank you, my friend. On to the next podcast.

Leo: Go eat some more meat, and I'll talk to you next week...

Steve: Bye-bye.

Leo: ...on Security Now!. Bye-bye.