Listener Feedback #130

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-326.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-326-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode #326, recorded November 9th, 2011: Your questions, Steve's answers, #130.

It's time for Security Now!, the show that protects you online. And here's the man who's doing the protecting, Mr. Steve Gibson of the Gibson Research Corporation. That sounds like a big enterprise, like you might have a tower downtown in L.A.

Steve Gibson: I have a large name and a short domain name, GRC.com. I love it when I...

Leo: Yeah, you got it early.

Steve: Yeah, when we first got it - actually, we got ours within a month of Microsoft getting Microsoft.com.

Leo: So that was, what, 1994, '95?

Steve: It was early. I remember the disappointment that Gibson.com was not available. But the guy that I had assigned to get us a domain name said, "But how about GRC?" I said, "Oh, I'll take it."

Leo: Let me do a whois on GRC.com because I think it would say...

Steve: Oh, yeah, it does. First registered...

Leo: First registered. It must be - if it was that early, it must be '94, '95. Record created 1991. Steve Gibson. You were an early mover. There was no Internet in 19...

Steve: I wasn't sure what I was going to do with it. But I had one.

Leo: No kidding. So, I mean, the World Wide Web, which of course this dotcom address was useful for, was created in 1989, and its first demo was, like, 1990. So you were literally, you were right at the beginning there.

Steve: Well, and we primarily used it for email. It was our email gateway. We used some funky - we were using cc:Mail was the client, and some funky gateway that we dialed up to, which was like a store-and-forward thing in San Jose or something. And it's, like, weird. But, yep, I was there, got GRC.com. It's funny, too, because sometimes people send me email: "Hi, would you be interested in selling your domain name? Because we like it." And then it's like, uh, no.

Leo: Well, we were talking about this on the radio show over the weekend because somebody called up, said "I have a three-letter domain name, and I'm wondering, how would I go about selling it?" I said, well, it's probably worth quite a bit because I don't think there are any more three-letter combinations available even. I mean, maybe there are. But mostly I would guess they're all owned by somebody who perhaps would sell them. And GRC is one of them. But I think GRC is a little less value than SEX or something like that.

Steve: Yeah. Speaking of which, the Triple X top-level domain is moving forward. There's some sort of a - you now can, if you're involved in the XXX-rated industry, you can petition with, not ISPs, with registrars using your trademark rights to get the second-level domain under the .xxx domain and begin to move that. Oh, and if there's competition for the same one, then there will be an auction to decide who gets it. So they're trying to do this thing right. And someone's going to make a lot of money, I imagine.

Leo: When we had this conversation, I was referred by somebody in the chatroom to Domain Name Journal, DNJournal.com, where they list the year-to-date top-selling domain name prices.

Steve: Ooh.

Leo: The No. 1 was sold in July of this year, was Social.com. You want to hazard a guess as to what Social.com was worth? $2.6 million. DomainName.com, $1 million. But...

Steve: DomainName.com?

Leo: I think - I don't know if they're really worth that. Gay.xxx, half a million dollars. Look.com, $400,000. And then numbers are popular: 11.com was more than half a million dollars; 33.com, $358,000. So SUV.com, that's one of the three-letter domain names that's worth something. See, GRC, I don't know who that would be worth - it's worth more to you than anybody probably else. But SUV sold for $210,000.

Steve: Wow.

Leo: I know. It's kind of a hot market right now. And there you go. Okay, enough of that. We have important things to talk about, thanks to Steve. We've got security news. We've got security updates. And this is a Q&A episode, so we've got...

Steve: Yeah, we have a bunch of news, enough that I'm not going to go in super detail on any of it, just sort of hit the high points so people know what's going on. We did have a - we are past a second Tuesday of the month with two very important Microsoft-related problems. A bad problem that's been discovered, a zero-day kernel exploit, which is one of the ways the Duqu worm is propagating. I tweeted about a Fixit patch that we'll be talking about. And also there's a new remote code execution vulnerability using UDP packets. And a bunch of other information, and a Q&A. So lots of good stuff.

Leo: You know, I should mention we had another exploit on our website this weekend.

Steve: It's actually in my notes was to ask you if something had crawled into your servers again.

Leo: Do you mind moving that up, and I'll explain it, and then we can get on with the updates and everything?

Steve: Yeah, do, do.

Leo: So any time you put new code in - and we're running a Drupal install on TWiT.tv - it's I think really important to make sure that when you put that new code in - and Drupal mostly works through modules. So we added - I think we added an RSS module. And it had an exploit on it. And I don't know - we're working now very closely with our sysadmins, who are great - Mike Taylor (Bear), and Chris Dieterle - and with IMAGEX Media, who does the web design. And IMAGEX put something new in, and it was immediately exploited. Which just shows you that there are people hammering on our system all the time. All the time.

And, I mean, literally, within minutes of it being implemented, they used it to embed a JavaScript exploit that - it's the same Java, I believe, my sense is it's the same

Java, old Java exploit that they were trying to put on people's systems. So you weren't vulnerable if you've been listening to this show and you've been updating Java. You'd have to have a very old version of Java on your system for this to even be a risk to you.

But what I really like is within, I don't know, an hour of this exploit being put on our system, Internet Explorer and Chrome both started warning people. Google is so good about, well, first of all, they keep us pretty indexed, I imagine. And so their crawler's always looking at our stuff. If I look at my logs, the Google crawler is always in there. They're just constantly crawling us. And I'm sure that's true of any large site. So they immediately flagged it because they see the malicious code. So I don't think anybody got bit. I hope nobody got bit. And we, of course, since that malicious code warning comes up, I get flooded with email. Thank you, everybody. I mean, if I didn't respond to you, I apologize because we got a lot of messages. But I do appreciate people letting us know that. And we got our...

Steve: Well, and from your perspective this probably gives you a better sense, relative to this podcast, of just how difficult it is these days to have a complex, multifaceted, interactive, social, I mean, dynamic site that has to be perfect. If there's a little mistake, that's all it takes.

Leo: That's right. That's exactly right.

Steve: Something will crawl in.

Leo: Whew. So we fixed it, and we have implemented - you know I say this every time. This is, I think, the second or third time this has happened. We've implemented, of course, new procedures to try to keep this from happening again. But it is very difficult because there are exploits, and it's PHP code, I mean, there are exploits everywhere. And it's hard to put something in production that you can say absolutely is bulletproof.

Steve: And even if there are not mistakes made, there are unintended consequences of the interaction of things. I mean, technically, allowing SQL commands to be interpreted by the server is not a mistake. But it is exploitable if a bad guy realizes that, when the server displays a page, it's going to run through the SQL interpreter, so they can post to a forum code that they want executed when their posting is displayed. I mean, so that it wasn't a bug anywhere. It was just a bad idea. I mean, it was something that was convenient, I mean, just as all get-out, super convenient from an implementation standpoint. But, whoops, it was abusable. There was a way to abuse that.

And it's just so difficult these days, as our systems become multiple layered. Authoring is coming from every different direction. Individual authors have different sets of expectations and assumptions which, if they don't match up, and you put these things together, clever people can find a way through. So, I mean, it really is a challenge.

Leo: Boy, I'll tell you. It makes me feel terrible. I mean, I hate to have that happen.

And it's embarrassing; but I also just, I mean, more than the embarrassment I just feel bad because I don't want anybody to get bit by visiting our site. So I'm really actually very grateful that Google and IE flag you so quickly. I think that's really good. But it is a big issue. This was not an SQL exploit. I believe, believe it or not, it was another JavaScript issue. But I'm not sure exactly how it happened.

Steve: Yup, scripting. You know how I feel about scripting, Leo. We don't have to go any further than that.

Leo: Anyway, so there's that story. And again, I apologize. And you'd think we'd know better, but it's so difficult to do. But I think we've put in new - I think part of the problem is IMAGEX, which is a great Drupal company, is not used to working with a site that is, like ours, under attack 24/7. And I think they just really didn't realize.

Steve: Well, and I don't know that it's a responsibility that you and the TWiT crew really ought to accept responsibility for. You are going to be using - almost no one is writing all the code themselves from scratch. Like me. But that's probably it. And it's just not feasible. To compete today, you have to be able to take modules from different places and just say, oh, this is a Drupal module. We want to add that service. Drop it in. Well, and we've seen, it's not practical to imagine you can audit it because...

Leo: You can't. You can't.

Steve: ...even people who know what they're doing read the code, and they go, oh, yeah, that looks fine. Until it doesn't. But all you can do is what you have always done, which is respond immediately and fix the problem. I mean, that's all anyone really can ask of anyone, you or any other group. So...

Leo: Thank you. But I feel the - I just - I feel bad. It shouldn't happen. But we're going to do our best. So, and then, by the way, it does make me feel better that it happens to everybody, including Microsoft. A zero-day kernel exploit? Holy cow.

Steve: Yeah. So in examining what Duqu was doing, and this is this sort of maybe it's related to Stuxnet. No one's really sure. There's now a question about whether there's actually any common authorship involved. Initially people said, oh, look, it's derived from the source code. It's like, well, maybe not. So, but in analyzing it, they saw it doing something, that is, propagating in a way that they had never seen before. And what it turns out was that it was using a previously unknown, which is to say zero-day, Microsoft Windows kernel flaw, which the authors of this worm obviously knew about. It's in the parsing of TrueType fonts.

And somebody, when this news broke, I saw a tweet that came to me and said, okay, and why are TrueType fonts being processed and parsed in the kernel? It's like, yes, I know, I know. It was a decision Microsoft made a while ago in the name of performance, back when Windows was still too sluggish, and we didn't have chips that were fast enough. And so Microsoft moved GDI, the Graphics Device Interface, into the kernel. The

proper architecture, the original architecture was to make that run outside as a subsystem. And Microsoft could not resist the temptation of moving it down into the kernel. And we've seen a series of problems that have resulted.

And you and I talk about these things sort of from a theoretical standpoint. But when we're talking about the Duqu worm, using this as an example, to propagate, I mean, these are machines that are actually being infected by this. So this is not - sure, it's an inconvenience for those of us who are not infected. We've got to update Windows and patch and so forth. But, I mean, it really does represent a problem when these things end up being the entry vector for malicious code into people's machines.

Now, this has happened so quickly that Microsoft has not been able to respond. This has not been fixed in yesterday's, that is to say the second Tuesday of the month's patch. A different bad problem was fixed. This one hasn't been. So I tweeted, as soon as Microsoft produced one of their single-click Fixits, I tweeted that to the - I think I have about 26,000 followers now, and I saw that it was heavily retweeted. This is a way of essentially denying access to one DLL. The DLL is t2embed.dll. And so people could go to either my - look at my Twitter feed, where it's still right there at the top, so that's Twitter.com/SGgrc. And you will see that and a few other things that I've been tweeting recently. So anyone with a web browser can do that. You don't have to be a subscriber to Twitter or anything. Just Twitter.com/SGgrc. And that contains a link to support.microsoft.com/kb/2639658. And that's the page with the one-click Fixit.

Now, there are some command line options which can be used. But they vary depending upon what version of operating system and what service pack and everything. So it's just easier to let this little Microsoft one-click Fixit solution go. And I would suggest, since we don't know whether Microsoft will do an out-of-cycle patch, this is bad enough I will be surprised if they don't. I can't imagine they're going to wait until December's Patch Tuesday for this because this is not good. This is being actively exploited by a worm which is having its way with Windows right now. So I expect we'll see an out-of-cycle patch, but we don't know when.

So people listening to this podcast can simply follow those links, click a button, and it will - basically this shuts down access to a DLL that we don't need. It's there. It offers some features. It involves embedding of TrueType fonts. There's both a Word file exploit and a web-based exploit. So Microsoft, in their executive summary, they said, "Microsoft is investigating a vulnerability in a Microsoft Windows component, the Win32k TrueType font parsing engine. An attacker who successfully exploited this vulnerability could run arbitrary code in the kernel. The attacker could then install programs; view, change or delete data; or create new accounts with full user rights. We are aware of targeted attacks that try to use the reported vulnerability. Overall, we see low customer impact at this time. This vulnerability is related to the Duqu malware."

And then in mitigating factors they said, "By default, all supported versions of Microsoft Outlook, Microsoft Outlook Express, and Windows Mail open HTML email messages in the restricted sites zone, which disables font download by default. If a user clicks a link in an email message, the user could still be vulnerable to exploitation of this vulnerability through the web-based attack scenario." So Microsoft recognizes the problem. I think they have to be working on getting this thing fixed quickly. But there is a way for us all to protect ourselves in the meantime.

What they did fix, I think there were four updates in yesterday's, that is to say, November 8th, which is the second Tuesday of the month, Windows patch. And the security community is telling everyone, do not be slow in updating Windows. There was one very bad problem, and surprising. It fits in with what we've been talking about with

Internets and packets and UDP and TCP and so forth. This is not being taken advantage of, exploited, in the wild yet. So this was a privately reported vulnerability which Microsoft says, ooh, crap, let's get this thing fixed right now.

Leo: Did they really say that, "Ooh"?

Steve: I think somewhere. Believe me...

Leo: Somebody said that.

Steve: ...probably more than that, even. I clean it up for the podcast. They said, quoting Microsoft, "This security update resolves a privately reported vulnerability in Microsoft Windows. The vulnerability could allow remote code execution if an attacker sends" - get this, Leo - "a continuous flow of specially crafted UDP packets to a closed port on a target system." So that's freaky. I mean, that's just - you would think...

Leo: Closed should just bounce off.

Steve: Exactly. It ought to have no effect whatsoever if a UDP packet hits Windows. And apparently the Windows firewall is no protection.

Leo: Oh, dear.

Steve: Yes. They said, "The security update addresses the vulnerability by modifying the way that the Windows TCP/IP stack keeps track of UDP packets within memory." And again, you'd think, okay. But, okay. So the fact is the packet's going to come into the machine and affect things. It's going to bounce around. It's going to do something until the system decides that there's nowhere for it to go, that is, there's some processing upstream of that port is being - that port is closed. For example, certainly there's a list of ports that are open. And so something's got to rifle through that list. So there's got to be list pointers that are being followed and so forth. So we're able to sort of stand back and say, well, the port's closed. How could it go anywhere? Well, how did it know it was closed? So some work was done when that packet came in. And that must be where this problem is.

So they said, "For more specific information about the vulnerability, see the FAQ subsection for the specific vulnerability entry under the next section." And it didn't say much. But it said, "An attacker could exploit this vulnerability by setting a continuous flow of specially crafted UDP packets to a closed port on the target system." For workarounds it said, "Block unused UDP ports at the perimeter firewall." Meaning like an Astaro...

Leo: The NAT router.

Steve: An Astaro sort of thing.

Leo: Or an Astaro box, yeah. Well, that answers the question because one of our chat people said, "Well, does this affect your router?" But, no. It's a Windows exploit.

Steve: Correct. And they said, "Blocking unused (closed) UDP ports at the perimeter firewall helps protect systems that are behind that firewall from attempts to exploit this vulnerability." So...

Leo: How bizarre, though. I mean, I don't get it.

Steve: I know.

Leo: It's not supposed to, if it's a closed port - you've been saying stealth your ports. Is stealth port, would that be - I guess you can't really stealth in Windows, can you.

Steve: Windows does some stealthing. I'm a little rusty on what they...

Leo: Or their firewall does.

Steve: ...stealth versus block. But, yeah, the firewall does. But it is the case that the port, the packet data is going to come into the interface and go into the stack in the kernel, and somebody has to decide to drop it. So it's something about - I saw something about a sequence counter or a sequence count or a state or something. So anyway, Microsoft's not talking about this. But this was fixed in today's, or in yesterday's, updates.

So in general, sometimes these are not very interesting fixes that Microsoft does. They go around with 29 of them in Word or Excel or something. But this is expected to be exploited before long. And your router will protect you. But your Windows firewall won't. And you don't want this thing to be left open. Oh, and XP is not vulnerable, Leo.

Leo: Just the newer ones.

Steve: Every other one, from Vista on. This was something that they introduced at that...

Leo: Remember we talked a lot about when - you were worried when they rewrote the TCP/IP stack from scratch that this kind of thing would sneak in.

Steve: Yup. And early on we talked about some longstanding old problems that have been solved that came back because that's just going to happen. So, ha ha ha, Adobe...

Leo: Yes, go ahead. I like that. Ha ha ha. Ha ha ha.

Steve: ...formally abandoned Flash...

Leo: Yes, thank god.

Steve: ...on mobile devices.

Leo: Oh, yeah, by the way, very important, that last clause. I interrupted. On mobile devices.

Steve: Yes. It is a little bit of a shame that Steve Jobs isn't still here with us to see his victory because you could argue that it's the iOS, the absolute refusal of iOS to support it. They're going to finish out with v11.1 for Android and BlackBerry PlayBook devices. But that's it. And the day before that was announced, they announced a 7 percent global workforce cut. They are eliminating 750 jobs...

Leo: Oh, dear. Oh, dear.

Steve: ...from their global workforce. And this morning the Adobe stock was down 12 percent at market open.

Leo: Yeah. Oh, dear.

Steve: So, yeah. It's tough. Adobe said they agree HTML5, which is what Apple has always been saying, is the proper solution for things like just displaying video. If all you want to do is, if you're YouTube, all you want to do is display video, you just don't need all of the everything else that Flash is. And it's everything else that it is that causes all these problems because it is, it's a very powerful interpretive environment which has been nice and portable and multiplatform and evolving over time. But as we know, it's very hard to do security right. And so Adobe's just going to give up on it for mobile devices.

Leo: Wow. Wow.

Steve: Newzbin, we talked about last week, was the site that had been asked by the MPAA, or, well, it was the site that the MPAA was requiring BT, British Telecom, to start blocking for their users. And I saw a little blurb that said that Newzbin was saying that almost all of their users are already using a workaround that renders whatever it is that BT is doing ineffective. And I did a little more research, because I was curious, and BT is just using apparently a relatively simple to work around technology called Cleanfeed, which they had already had in place to block child abuse sites. So essentially BT just added Newzbin to their existing site block list and said, okay, fine, we've complied with

the court order, knowing full well that Newzbin users since the fall have had some technology that just avoids this Cleanfeed thing, whatever it is.

And then, on the heels of that, another music industry trade group, BPI, has now asked BT to block access to the Pirate Bay and have threatened legal action if they don't immediately do so. I think they gave them a few weeks to do that. So I'm not sure where this is going to lead. It's going to be interesting to see because we have this question of, is it the ISP's responsibility to arbitrarily be compelled to block access to sites that industry trade groups would rather not be on the Internet? I mean, we're seeing an evolution here in thinking about whose responsibility this is.

Leo: Very interesting.

Steve: Yeah. Many people tweeted the news that on November 4th LastPass updated to v1.80.0 and added multifactor support, courtesy of the Google Authenticator. We talked about Google releasing a free and open source authentication system which Google supports on Android, iOS, and BlackBerry devices. But there's third-party support, because it is open source and anybody can support it, for Windows Phone, webOS, and Symbian. So it's got very good across-the-board support. And of course we like multifactor. Multifactor is good. The more factors you can have for authenticating, as long as it's not too burdensome, the better. So I did want to let anyone who hadn't picked up the news that Google Authenticator was available.

What that means, of course, is when you are proving to LastPass that it's you, for example, you went to a browser that you wanted to use LastPass to help you log in on, and you needed to tell the browser and convince the browser and LastPass itself that this is you, you would have to provide additional authentication beyond just your account name and password. You'd need to use the Google Authenticator, which uses - and we'll talk about this in our Q&A a little bit, this comes up again - the technology, the specific technology that they're using. But it's very much like the original time-based football that you and I discovered, Leo, the little PayPal football with the six-digit code that is varying constantly. And I've got the VeriSign Authenticator loaded in my BlackBerry, and it's the same thing, a six-digit code with a little clock that shows you how much longer that code will be valid, and then it changes. So anyway, anybody using LastPass can update to 1.80.0.

Leo: I think it does it automatically in most cases, doesn't it? Yeah.

Steve: Yeah. And we've had some more news and trouble over the certificate authority world. Mozilla, Microsoft and Google all together moved to remove a Malaysian intermediate CA from their browsers. So remember we've talked about certificate authorities. The root, the so-called "root certificate authorities" are the big guys like VeriSign and Entrust and DigiCert and so forth. And their certificates are what our browsers trust and all contain. It's possible for an intermediate certificate authority to be granted their ability to issue certificates by a root authority.

And so, for example, in this case this Malaysian intermediate CA was found to be issuing bad certificates. Not fraudulent, but they had weak keys, and they had some parameters missing. It was like their technology was behind the times, or it had been misconfigured, or maybe something got into their systems and deliberately weakened the certificates that they were producing. We're not really sure what the back story is. But their

certificate was signed by the Texas-based company Entrust.

And so, because this Malaysian certificate authority was found to be issuing certificates that the browsers did not feel was secure enough, they've been prevented from doing that. And a Dutch telecommunications certificate authority, KPN, has stopped issuing certificates when they discovered that the web servers which they were using as the front end for processing these certificates had been compromised, perhaps as long as four years ago.

Leo: Whoa.

Steve: Yeah. So it's like, whoops. Okay, we're not going to issue any more until we figure out what's been going on and how big a problem this is.

Leo: If a compromise happens in the forest, and there's no one there to observe it. I mean, really, it's been going on for four years and nobody ever noticed. I mean, I don't know what that means. It sounds like...

Steve: Yeah, well, they don't know. They don't know that any bad certs were issued. They just realized that there was - malware had been installed in the systems. And these guys are doing the right thing. This is as responsible as you can be. You'd rather not have anything in your server. But if it is, then you say, okay, whoops. And the only way the world knows is that they said, okay, we're going to stop issuing certs now because we have to figure out what this means. Which is all anyone could ask of them. They're absolutely behaving responsibly.

And I did pick up a little security news that Apple has postponed their enforcement of app sandboxing, iOS app sandboxing. So Touch, Phone, and iPad. It was going to go into effect around now, but they're moving it back to March 1st and then saying we're not moving it again. Now, the problem is that this is - it's a mixed blessing. It is an enhancement to the security of iOS and of all iOS apps at the inevitable cost of features. So developers are not happy and have not been implementing Apple's sandboxing because it is restricting. It's restrictive and restricting some things that they would like to be able to do, reaching out of their own file system zone in order to...

Leo: It's actually, Steve, it's worse than you think.

Steve: Okay.

Leo: It's not iOS, it's the desktop. They're talking about all apps sold in the Mac App Store. And I understand your confusion because you don't use an iPhone.

Steve: No.

Leo: They're not talking about iOS. That's already implemented. They're talking

about in the App Store for desktops. So it's really kind of a shocker. And it's something I'd actually been worried about for some time because, while you can still, and always probably will be able to - well, I shouldn't say always - for the time being be able to sell apps outside the App Store, there's so much convenience and value to buying apps in the App Store that I think a lot of users have moved to the App Store. So what Apple's now saying, they've said all along no demos, no betas. What Apple is now saying is, if you want to...

Steve: I'm stunned.

Leo: I'm stunned, too. If you want to sell apps in the App Store on the desktop, your apps must be sandboxed. We've talked about this on MacBreak Weekly. I think the iOS-ification of the desktop is where Apple's headed.

Steve: Ah.

Leo: Yeah. And so what they would like - and you're a security expert. I mean, there's certainly security value to doing this.

Steve: Oh, yeah, as I said, with the inevitable loss of features. Now I'm stunned.

Leo: Can you imagine an application that cannot write to the file system?

Steve: Holy moly.

Leo: I truly believe that Apple's intent is to get everybody using its desktop computers to essentially be in an iOS-style state. It will be undoubtedly secure. And I don't, at some point, I don't understand how the transition's going to occur because of course you can still - I can buy an app that can write to the file system and for the time being will continue to. At some point, for this to make any sense, Apple's going to have to turn that feature off and say, just as on iOS, you must buy from the App Store, unless you jailbreak it.

Steve: So maybe they're - okay. So...

Leo: Yeah, now you've got to think about this, don't you.

Steve: Whoa. Mac OS X apps, which I buy from the...

Leo: From the App Store.

Steve: ...from the Apple App Store...

Leo: Right, will be sandboxed as of March.

Steve: ...as of March 1st, will enforce sandboxing.

Leo: Yes.

Steve: So for a while the user will have a choice. They'll be able to say, well, everything I'm hearing says that the apps that I buy from the Mac App Store are safer.

Leo: They're secure.

Steve: They're more secure...

Leo: They may be a little inconvenient...

Steve: ...because of whatever.

Leo: ...but they're more secure.

Steve: Wow.

Leo: Now, my suspicion is that Apple's apps will not have to be sandboxed. So what I'm suspecting - here's what I think will happen. I don't - and they may back down on this.

Steve: Are these toy apps? I mean, these things...

Leo: No. No.

Steve: They're not toys?

Leo: Final Cut Pro 10...

Steve: No, that's not a toy.

Leo: ...is sold in the App Store. In fact, that's going - right now you can still buy a disk, but that's only - that was they bowed to pressure on that one.

Steve: Well, you can't have Final Cut Pro unable to reach out of its own little...

Leo: Am I wrong? If you sandbox, does that not mean that you cannot write to the file system? Isn't that what that means?

Steve: Well, I will know next week.

Leo: I think what's really happening, and I think - now, with Steve gone, this may change. There are already some changes happening. And I think that this was a Steve. But with Steve gone, some of this is up in the air. But here's what I think they were headed towards: making, essentially making - and by the way, Microsoft's kind of doing the same thing with Windows 8 - making the desktop essentially an iOS, which is more secure, more controlled. I suspect Apple's apps, just as on the iOS, Apple's apps can do things that other people can't, because we trust ourselves, I suspect that what this does is pushes you - and Apple's always wanted this - into Apple apps. Apple would like you to buy only Apple apps for...

Steve: But they don't have, I mean...

Leo: They do.

Steve: They don't even begin to have the breadth of what today's desktop has.

Leo: No, of course not. But they have office suites. They have the bulk of it. When they started creating iLife...

Steve: And it comes with so much.

Leo: And it comes with it. And it's very inexpensive. I don't, you know, who knows what this is. It's nuts. I think what will happen is that people who want a full operating system will have to migrate somewhere else.

Steve: I was going to say that. I was going to say that, if this continues, then this really changes the terrain, where these mainstream, high-volume, consumer OSes become closed systems, and Linux for the first time really starts to look like the place where the hackers...

Leo: If you want to do anything, yeah.

Steve: ...and the power users live, yeah.

Leo: Now, Knox Harrington says, well, wait a minute, isn't Chrome sandboxed? It is. I mean, tabs within Chrome are sandboxed. But that doesn't mean the app is sandboxed; right? The app can...

Steve: Well, and a browser is not trying to be Final Cut Pro. It's just - it's a viewer into things, basically, with the ability to send some stuff back to the mothership. But, wow. Wow. And I'm looking at my little line I quoted here: "Developers fear loss of useful features" [choking].

Leo: On the desktop.

Steve: You think? You think?

Leo: But they look at the success of the iPad. And the iPad is really the computer for the rest of us. And I think what they say is, well, the rest of us want a desktop operating system that's...

Steve: Well, some of them would like a keyboard also. So we're going to give you the - we're going to call it the iPad with a keyboard is your Macintosh.

Leo: So the idea is to minimize the damage the application could cause if it were malware or were compromised by malware. So that makes sense.

Steve: Precisely, the idea being that within the sandbox applications are - they're given, I mean, I understand a little bit how it works on iOS. You have this - because I looked and remember when iOS was beginning to happen, we talked about it. There's a sort of a pseudorandomly generated directory leaf off of the file system, and all of this is opaque. The application doesn't get a file system. It doesn't see a root. It doesn't see a hierarchy. It just sort of sees, here's your spot. Good luck with it.

Leo: Now, the question is also how Apple implements sandboxing. We're interpreting it in the most draconian, strictest form. And I'm looking at what they do right now in OS X Lion, and they do allow an app, for instance, to write to the hard drive. But they have to go through Apple's dialogue box to do so. They can't examine other people's files. In other words, it's almost like application-based permissions.

Steve: Yes. I'm looking at something here, it says, "To then meet the program's needs, the developer includes a sandbox rule called an 'entitlement.' That allows the program to access the needed resource defined in that entitlement. The entitlements are managed by Apple, and thereby allow Apple to centralize how sandboxed programs can access resources in OS X."

Leo: So I imagine Apple will not make it the most draconian possible sandboxing, at least initially.

Steve: Right. Says, "The developer can add as many entitlements as he wishes to give his program as much system access as is necessary. However, the idea is the developer only enables the entitlements that are needed to allow its program to run." So that's interesting. What this creates is, it's like having a firewall between the API of the OS and your application.

Leo: Right. And the API controls access.

Steve: Yes. Well, normally the API is everything. And so an application can use the OS's API to do anything it wants. But now imagine that we deliberately impose a layer between the application and the operating system's API, like a firewall.

Leo: And require you to use it.

Steve: And require you to use it. And it's like ports. In the same way that a firewall, you can open ports through a firewall, you would be able to specify which aspects of the API your application needs. And the idea would be that that would be as minimal a specification as possible to create insulation between your application and the operating system if something should go wrong with your application. If it should get infected or made malicious, anything that got in there, into your application, would be unable to alter those entitlements, which were defined for the app, and so could not access areas of the OS on an ad hoc basis. So that's what we're talking about.

Leo: Yeah. I think what'll happen is Apple understands that people will rebel against this. You know, it's interesting. There's a good article on this on Ars Technica, who interviews two different developers. Rich Siegel, who does Bare Bones Software, he does BBEdit, says this is going to be fine. Most of our apps will be okay. We'll just go through the API. But then they also talked to Panic, which does a really great FTP program called Transmit. And they say this will break. We can't do it. This will break it.

So, and I think that also what'll happen is Apple will interpret this liberally, will provide a liberal API. But there's no guarantee they won't continue to shrink capabilities and so forth. And there's also a debate in the security world about whether this will have a benefit in terms of security. So it's a really interesting debate. It's a shocker, to be honest. And it's not getting a lot of coverage. I could see your jaw drop when I said, no, no, this is the desktop they're talking about. Yeah.

Steve: Yeah.

Leo: It's very interesting, isn't it. And, you know, only Apple could do this. You can't

see Microsoft doing this.

Steve: No. No. Microsoft is dispositionally completely against the idea of ever removing anything like this. Remember, Apple in the past has obsoleted chunks of API. They've said, "We're not going to support that anymore." It's like, what?

Leo: All the time.

Steve: Microsoft never does that.

Leo: And I think Apple will appeal to a certain kind of consumer. And I will probably recommend it to a certain kind of consumer.

Steve: Yes.

Leo: But if you want to hack your system, or you want to play with it, or you want to develop software, you know...

Steve: And here we've been saying, well, look what's happening to the Mac.

Leo: It's one way to do it.

Steve: It's beginning to crumble under the pressure from malware. It's like, well, no. Apple is working to respond.

Leo: Yeah. And they will be more secure.

Steve: They must be looking at the fact that they're succeeding with this kind of protection on the iOS platform, so here's our migration strategy. Anyway, something big just happened, Leo.

Leo: Yeah.

Steve: No, no. It's something else.

Leo: Oh, what else?

Steve: We just passed, just now, two minutes ago...

Leo: 48 minutes into the show. 44?

Steve: ...400 hours of Security Now!.

Leo: It feels like that sometimes. 400 hours, wow.

Steve: Dejan from Stockholm, Sweden, when I was going through the mailbag pulling the Q&A together, the subject was "400 Hours of SN!!" And I thought, what? And so he says, "Hi, Steve and Leo. Congratulations on cumulative 400 hours of Security Now!. According to my calculations, you will fill and finish the 400th hour at 44:28 of Episode #326." He said, "For the record, I excluded Episode 185a, the 'Gray Hair Computing' episode, from this calculation."

Leo: Oh, yes, I forgot about that one, yeah.

Steve: So I thought that was very cool. Thanks, Dejan.

Leo: 400 hours.

Steve: And I got a nice note also in the mailbag from Jeff Leckemby, with the subject - caught my eye, not surprisingly - "SpinRite does it again." He said, "Steve, slightly over two weeks ago a college professor and coworker of mine explained that his home computer would no longer boot. With a exam coming up soon, it was extremely important to him that the content of the hard drive be recovered since this is where the exam materials all lived. I took a look at the machine and found that immediately after POST" - the Power-On Self-Test - "the screen would say no boot disk was found. Not good. I thought this was a job for SpinRite. I put SpinRite on a CD and rebooted the machine. I selected Level 2 and let it run. That was on October 6th. Well, on October 20th it finished."

Leo: Okay, 14 days, that's not too bad. Two days, two weeks.

Steve: "It ran for approximately 342 hours."

Leo: Almost as long as this show.

Steve: "There was one" - exactly. "There was one recovered sector, seven unrecovered but repaired. The CD was removed, and the computer was rebooted. Amazingly, the machine booted all the way into Windows, and the exam materials could all be used again. I was astonished. No, not that it worked. I've used SpinRite before. But because I thought after 14 days that the hard drive was going to have so many bad areas that it would likely not boot, and much would be lost anyway. The drive was in bad shape, but SpinRite recovered it. That was not the case. It's been my experience with SpinRite that

it usually takes a couple of hours to work its wonders. But this effort shows that patience pays off. Many thanks to you for producing such a topnotch program." And thank you, Jeff, for sharing that with me and our listeners.

Leo: Moving along, are you ready for questions?

Steve: You betcha.

Leo: I got 'em. Starting with an anonymous listener. He's wondering why higher levels of TLS break the lower ones. So, and we've talked about this a little bit when we talked about SSL and TLS. He says: I tried adding TLS v1.1 and 1.2 by enabling them on my system as you described. I also left TLS v1.0 and SSL v3.0 enabled. As also I think you recommended. But now I get connection errors with some sites. What happened?

Steve: Yeah. So [sighing]. That shouldn't happen.

Leo: Right, right.

Steve: That's not supposed to happen. We've talked about SSL and its evolution. And it was deliberately beautifully designed to handle exactly this situation, where there would be an interest in migrating - it was just presumed in the beginning there would be an interest in migrating - to future more advanced protocols, for whatever reason - more features, fixes to old things, who knows what. One way or another, the designers said we need to facilitate that.

So as we've discussed, when the client is initiating a connection to a server, it establishes a TCP connection first, which we've just been talking about. Then the first thing it does is an SSL handshake. The first packet it sends is a list of all the protocols that it knows about and would be happy to use, in order from most secure to least secure. And there have been attacks on SSL where, for example, a bad guy will intercept that and strip out all the more secure ones so that the client appears to only support very insecure ones. And in fact, when I say "very insecure," it goes all the way down to none. It goes all the way down to "do not encrypt." So you can actually have a nonencrypted communication over technically an SSL connection. But now that's not normally allowed or accepted. But the spec does support that.

So this list of the things the client would be willing to use, knows how to use, goes off to the server. The server is supposed to choose the most aggressive one, the best one, the most secure one from that list that it also knows how to use. And that's how this handshake is negotiated, how they end up arriving then at the highest level, the latest version of the protocol that they both understand.

So that means, if all this was working right, you could turn on - you, we, users of this podcast, listening to this, could go to our browsers and operating systems and turn on 1.1 and 1.2, knowing that they're better than SSL 3.0 and TLS 1.0, both that now have known problems. And it ought to just work seamlessly. Servers which are up to speed on 1.1 and 1.2 ought to negotiate those improved protocols, and those that aren't should be able to fall back to the ones they do know about, the best that they can do.

Get this, Leo. When people started actually trying this, they found out they couldn't connect at all. And that's what happened to our anonymous listener and many others. It turns out that this had never been tested, if you can believe it, at the server side. And there are servers on the Internet, popular ones, which, when they see TLS 1.1 and 1.2, send a reset on the connection. They interpret it as an attack, and they instantly drop the connection, and the browser says "could not connect, sorry." It's just it's a hard - they don't even fall back and renegotiate. It's just essentially the client is hung up on as quickly as the TCP protocol, the underlying protocol, allows, with the server sending an RST, a reset packet, which instantaneously terminates the connection. And so for anyone else who's listening who has tried this, that's what happened is there are buggy servers.

The good news is, now that we realize we really do need to move to 1.1, away from 1.0 of TLS, there will be pressure on servers to get themselves patched and to work correctly. And so we can presume this is hopefully a short-lived phenomenon. But that's what's going on right now. That's why we really can't...

Leo: Interesting oversight.

Steve: Yeah. Ooh.

Leo: Try it with a server first.

Steve: Yeah.

Leo: Oops.

Steve: See if that works.

Leo: Question 2, Frank C. in Mississauga, Canada wants to share news of, he says, quote, "The Best Version of Firefox to Date." Wow. Steve, I was listening to Episode 323, and I couldn't agree with you more about versions 4, 5, and 6 of Firefox when it comes to memory leaks. I was also ready to give up on Firefox, but they came out with version 7 just in time, and I was happy again. Now, I know you don't like using the latest version of things, but do try version 7. Really. The memory handling has been greatly improved, especially when multiple tabs are left open overnight, which was the issue you had. I usually have 25 to 30 tabs open at all times. Thanks for a great show.

Steve: Yeah. So I've heard people go both ways. I tried 7, and I still had trouble. I'm now, as you know, famously back on 3. And I'm happy. So 4, 5, and some - they broke something at version 4. Now, 8 has just happened. In fact, I got it, and I played with it a little bit a couple days ago when it was two days away from release. It was in late-stage beta at that point, so maybe it's already out. "But it takes a lot to move me off of somewhere I'm happy," he says, using XP. So I'll be staying with 3 for a while. It does everything I need. All my add-ons work just fine, and I have got no memory problems at all.

So, although I will say that the gizmo I talked about last week - and I'm trying to look at it to bring the name back up. Memory Fox does work. That little Memory Fox add-on, what it does is it pushes your in-memory allocation down, does not reduce the total size of the app. So virtual memory consumption remains the same, but it does free up working memory, so that can be a benefit and help speed up your computer. So the Memory Fox add-on is useful under all versions of Firefox. And so that I could say that I like.

Leo: I am now - I have version 8 on my system. I am going to open 25 tabs, leave it open, we'll see what happens.

Steve: Okay.

Leo: Of course, I'm on a Mac. Does that change anything? Is it the same issue?

Steve: The problem is over on the Mac. I think actually we have a Q&A later on about the Mac.

Leo: Okay. Stay tuned, boys and girls. Moving along to Question #3, a surprised IT admin wonders how iPads keep their cool: Steve and Leo, I've been a listener of Security Now! since Episode 1. And believe it or not, I used to watch TechTV in the day. Wow. You're a gray-hair. I still meet lots of people who watched TechTV, I'm happy to say. In fact, now what the latest thing is I meet people who say, "When I was a kid my dad and I..." or "I grew up watching." That's what I meet now. And these are adults, in their 20s. I often say, "You must have been a little kid when you were...." "Yes, I was a child."

Anyway, I've heard you and Steve - you, Leo, and Steve go on and on about iPad this and iPad that. One thing you both failed to talk about, at least that I can recall, is how well it handles heat. I work in IT at a public school district, and we've recently started allowing iPads to connect to our network. When we ordered them for compatibility testing with our network, I figured it would be nothing more than a paperweight because I love my Droid and my Windows tablet. Boy, was I wrong. I found that not only can I do just about everything my Windows tablet and Android can do, it can run much longer, and temperature-wise it runs much cooler. I've never felt my iPad get hot, actually. My question to you...

Steve: You mean your Droid or your iPad?

Leo: My iPad. My Droid gets really hot. So my question for you guys is how did Apple manage to keep the thing running so cool and efficient? Yeah, every smartphone I've had on the Android side gets pretty hot. But neither the - actually, my iPhone can get pretty hot. But I've never had the iPad get hot. But remember, the iPad has a large cooling surface on the back there.

Steve: Well, it does. What Apple did, Apple had the advantage of starting relatively late. Smartphones existed, BlackBerries and Palms and all these other things. So they were

able to say, okay, what do we care about? We know we need long battery life. And they had at that time Steve Jobs, the overlord, making sure that it would meet his very exacting standards. So it turns out that processors that are stopped don't use any energy. It's only switching that uses energy.

The way our technology works is that essentially you're dumping electrons into or pulling them out of essentially metal conductive areas inside the chip. But unless you are in the process of moving the electrons, there's no current flowing. And if there's no current flowing, there's no heat being generated. So what Apple did from the beginning was they arranged for the iPad, in much the same way that the Kindle gets its multi-weeks of life, these really long-life devices are stopped when they're not actually doing something. The way...

Leo: Yeah, Intel calls it "Speed Step."

Steve: Well, Speed Step has been around for a while. This is actually stop step.

Leo: Not just slowed down, just zero.

Steve: Yeah. It stops. The way these devices can play video is that they offload the entire job now to a video coprocessor, or at least to a video portion of their chip, if it's integrated, and because you've got to have processing power in order to decode H.264 and MPEG and so forth. So there's going to be some work done. But that's, I mean, in overall, that's relatively less work than what the processor is doing.

And so the processor, in either stopping completely, and it's very possible for these state-of-the-art, like the ARM-based processors, they could just stop. They have like a wakeup timer that will tickle them and bring them back to life out of stop every so often to see if anything has happened that needs their attention. So what you'll find is, if you were to do something with your iPad that really forced the processor to stay running all the time, you kept your fingers on the screen, you were moving things around, maybe like heavy-duty game play where you're really asking it to do a lot, you'll find that really will pull the battery down a lot faster. But if you do things like reading or scrolling a page and then letting go and it just sort of sits there, battery life is a lot longer because the processor's actually stopped when it knows it doesn't have to do something. And that's not something you can just add. That's the problem.

When I said that Apple had the advantage of coming along later, Windows has this dilemma, and that is that its architecture, its fundamental architecture makes assumptions all throughout it that the processor is running all the time. And Microsoft can try to play games, but this is why from the very beginning Windows CE had critical battery life problems, because it's got the word "Windows" in its name. And this is an old operating system that was designed pre-battery life, I mean, pre-portable device. And all of those assumptions are built in. You just can't throw them away. So that's an advantage that iOS has.

Leo: Yeah. And going back to that Flash story we were talking about, I was told, and I don't know much about this, but I was told that by knowing what applications you're going to run, you can also design the processor more efficiently. And so the

A4 and A5, because they knew they didn't have to support Flash, could be in fact fabbed differently. Does that make sense?

Steve: Maybe not Flash. But...

Leo: It seems like that's going to be kind of a generic thing.

Steve: Yeah. But certainly you could absolutely profile the processor so the things it does well are the things it is doing a lot. And that's really what you want. You want it to be as efficient as possible because if it has many more cycles to get the same work done, those are many more cycles of power being consumed, and longer before it's able to shut itself down. So if you know what the target applications are and the things they need to do, you can design the hardware to do those well, and not waste space on your chip, which is expensive and can consume heat, not waste space on your chip with things that are not going to be used very much. So, yes, tuning really does make sense.

Leo: And this is something Apple's taking very seriously. Of course they're using an ARM design as a foundation. But they bought PA-RISC, or PA Semi, rather, which was a...

Steve: They got their design talent.

Leo: They got design talent. And I was told, and I find this hard to believe, but that there are 1,000 people working on chip design at Apple. 1,000 people. That tells you how seriously they take this. Very interesting.

Anyway, moving on to Michael Landers in sunny California. He wonders whether Ars Technica is copying us. He points to an article on Ars from October - "When Passwords Attack: The Problem With Aggressive Password Policies" - that exactly echoes your comments on password change policies from a couple of weeks ago. It's a good article, since it echoes you. However, this isn't the first time I've seen an article from Ars that closely echoes your podcast, shortly after your podcast. Perhaps one of their writers is using your show for inspiration? I don't think so. I love Ars.

Steve: Well, I was going to say, Leo, you know what they say about...

Leo: Great minds.

Steve: Well, there's that, and also about flattery, is that imitation is the sincerest form. And I don't know whether anyone's getting ideas from the podcast. I hope so because I get great ideas from them. Ars is one of my main sources. In fact, they're one of the few Twitter feeds that I follow over on my account where I follow things. And I'm always glancing up to see what's going on. So I hope they're getting some value from me because I'm sure getting a lot from them.

Leo: I think that a lot of the same ideas in the security field go around. I don't know if you can say that - I would never accuse them. Sean Gallagher wrote that article. I would never accuse Sean or anybody else of stealing from us. And I'm with you. I think Ars is increasingly my go-to place for long-form intelligent articles on content. I actually pay for a premier subscription, which you don't need to do. But I just wanted to support them because I really feel like they're doing the kind of journalism, tech journalism I believe in.

Steve: Speaking of which, about paying for things, what's happening with Wikipedia? They need money again already?

Leo: Every year they do a - already. You know, every year they do a...

Steve: I gave them a bunch of money not long ago. I thought, wait a minute.

Leo: Every year. It seems like a long - I think it was a year ago they did the last time, where Jimmy Wales would show up on your front page saying, "Give me money."

Steve: Oh, he's back.

Leo: He's back.

Steve: He's back, yeah.

Leo: Well, I'll tell you, I don't mind. First of all, I think Wikipedia is one of the great resources of the Internet, if not the greatest single resource. And I use it almost every day. And I remember talking to Jimmy. And the board and others continually tell him and other people, look, we could make a million dollars a day if we just put ads on this page. We have that much traffic. And to his credit, Jimmy has consistently said there will be no ads on Wikipedia.

Steve: Oh, and think about that, context-based ads for, I mean, it's a natural. It really is. Well, we'll have them sooner or later. It's inevitable.

Leo: Well, no, I don't think so. And I think that...

Steve: I don't know, Leo.

Leo: There's a lot, they're turning their back on a lot of money. But it costs money to run Wikipedia. But that's why, whenever he asks, or whenever they ask - I

shouldn't say "he" because it's not just Jimmy anymore. But whenever Wikipedia asks, I always give them money because I believe that I get the value out of it. I think they're well worth it. Anyway, I'm on Wikipedia right now. I don't see any begging going on, so I don't know. Hey, your public radio station does this. Right?

Steve: Oh, my goodness, yes.

Leo: All the time.

Steve: Yes.

Leo: It's just like that. Moving along. Question #5 from listener Jo-Jo - hey, Jo-Jo - in the European Union. We were talking about dual Internets. He thinks it's a good idea: You've often mentioned how the original designers of the Internet were only focused on getting the mechanics of packet routing to work and paid no attention to security. So it's no wonder that pretty much everything we use today is a failure from security perspectives - Ethernet, TCP, BGP, DNS, SMTP, FTP and so on. Maybe the Apple approach of not lugging excessive baggage into the future should have been applied to the Internet and most of its crappy protocols 10 years ago. Are DNSSEC and IPv6 the only signs of improvement for Internet 1?

Steve: So I read this, and I thought, it's not so bad.

Leo: They're not that crappy.

Steve: I think that maybe, because this show is about focusing on problems, that that's what we focus on. But let's remember that everyone has a front door whose lock can be easily picked. I mean, security doesn't have to be perfect in order to provide value. And we're all getting an amazing amount of value from the Internet globally. And yes, it could be better. Yes, there are bad guys lurking around who are taking advantage of these things. And yes, it is true that the designers of the 'Net did not understand what was going to happen. And were we to start from scratch today, maybe we could make it better.

But I'm not even sure that we really can because, when we were brainstorming a little bit last week about Internet 2, when that article came up, both Microsoft and the FBI were saying, yeah, well, we need to, like, authenticate every packet that we put on. It's like, okay, stop. What are the implications of that? I mean, if that had - it could have never happened if that were the way it was designed. It did happen because it was just this big, spongy, happy, friendly, packet-passing blob that all kind of worked. So I'm glad we have it.

And I would suggest that there are security problems everywhere, which is why I had thought of the front door. We also have glass windows that can be broken. We've got people leaving car doors, like things in view inside a car that causes their windows to be smashed. And, I mean, and back when alarm systems used to use phone lines, the phone lines would get cut, and then the alarm would ring, and the police would not come

because the monitoring service had been disconnected. I mean, there wasn't - I guess my point is that we know that security is hard. Bad guys will always work to, and can, defeat security. I don't see how we can ever move past the Internet foundation we have. We can improve things incrementally. And this show focuses on those areas which need improvement, I would say. But, boy, I'm not unplugging. It's fantastic. And it works.

Leo: And it works. Yeah. Good. I like that. "Wogsy," who apparently is in "Flyover Cornfield" - I'm thinking Nebraska - comments about Internet bandwidth: I'm a little perturbed about this notion of selling tiered broadband or DSL service, based on the philosophy that a faster data rate costs more because it uses more bandwidth and encourages more use, therefore causes more congestion. I think it's a rather mischievous, if not fraudulent, interpretation of electronic throughput. Wouldn't the "tubes" be less congested if all packets were delivered more quickly, clearing buffers and giving routers and servers more free time to make fewer errors? I should think more clear open bandwidth would be available for all, if all routers, switches, and servers ran at maximum speed at a flat rate, rather than deliberately holding, limiting, buffering, or rerouting lower tier data. You know, that's a really interesting point. What do you think?

Steve: Well, all routers, switches, servers and so forth do run at a maximum flat rate. It's really not since modems that there was a variation in the rate at which the bits moved. And it was, as we've talked about in our switching from wired, point-to-point connections with modems, it's when we switched to packet-based communications that this changed. So the way bandwidth is apportioned is it's very much the way the cell services do, where as we have cell phones with larger bandwidth connections. They're not actually changing the frequency of the carrier that allows the bits to pass between the phone and the local cell any quicker. What they're doing is they're bundling up channels and using a greater percentage of the available bandwidth, for example, in the case of a cell phone, which is where you go from 1 to 2 to 3 to 4G. It's channel bundling, essentially.

The similar analogy, for example, with broadband is that your data bits are not flowing in any faster when you have a higher broadband connection. It's that they're flowing more often. So everything is packetized. Somebody who has a 1Mb broadband connection can be in the house next to you, and you've got 50Mb. Well, the bits are coming in, and they're moving over the wire just as fast. He's just getting more of them. He who has the 50Mb connection is getting a greater percentage of those bits.

So all of the switching and all of the operation that we've got on the Internet, it's all running as fast as it can. When you've got a 100Mb Ethernet connection, even if you're typing slowly, what you're sending is little tiny packets very quickly across that fast connection. And if you transfer a huge file, you're sending a lot of large packets across that same connection. But the actual speed is always the same, is always as fast as it can go. So it's just the case that providers, see, their logic is, well, this user is using a greater percentage of our capacity, so we want to charge them more. And if that's the model they want to use, I could see that it makes some sense.

Leo: I actually prefer - I think the solution of charging by data rate is far preferable to the other solution they're using, which is capping after a certain number of gigabytes are downloaded each month.

Steve: Yes. That just gives you...

Leo: That's more draconian.

Steve: That just gives me a queasy feeling. It's like, well, wait a minute. How much do I have left? Am I going to run out?

Leo: I'm willing to pay for faster service, God knows we do, and symmetric service, and that kind of thing. And I don't think that that's unreasonable to ask. Although I've had the debate many times with John C. Dvorak, I think essentially, once you pay for the infrastructure, the bits are pretty free. They're pretty cheap from the point of view of the ISP.

Steve: Whether they're being used or not.

Leo: Right.

Steve: Because the infrastructure still has to be there.

Leo: Right. But the infrastructure is a one-time capital cost that's amortized pretty quickly. It's almost like the Golden Gate Bridge. You build the bridge, and you've got to maintain it. But the building of the bridge is the expense. Then they put up a toll structure to charge by car; right? And the truth is the bridge costs no more if it's full of cars or it's got one car. So I just think that there's a, well, I mean, it's a business. ISPs are going to get as much as they can out of you.

Steve: Oh, and believe me, they're making money, Leo.

Leo: And I don't think they're going poor.

Steve: No.

Leo: No. Question 7, Jon in Lincoln, Nebraska - another cornfield flyover - worries about giving Google too many eggs: Steve and Leo, I love the podcast. I'm a long-time listener, LastPass and Vitamin D advocate. I have the entire family well educated and believing now. Yay. Over the weekend I saw that now LastPass - as we mentioned in the news - supports Google Authenticator. This is great news because I currently use the app on my Android phone to get into my Gmail account. I also know how much more secure two-factor authentication is, thanks to previous Security Now! episodes. But it makes me wonder if tying so many of my services to Google is a good idea or a potential security problem. What happens if Google were to go down for a few hours? Any thoughts or opinions on this would be appreciated. Am I putting too many eggs in Google's basket?

Oh, by the way, I should point out that, and I guess you have also, but LastPass has always had two-factor authentication, which I use. But the way they did it is they had a security program that you put on a USB key that you would run. And so this was - the second factor was kind of like a YubiKey, coming from a USB key, which I carry in my pocket.

Steve: Right. And so of course now using Google Authenticator...

Leo: I don't have to carry it, right.

Steve: Right. Precisely. And so this was always our goal was to have a single solution of some sort. We do know that any reliance on the cloud comes with benefits and costs. And the costs are availability and security. Is the stuff we're going to put up there secure so that nobody else can get it? We've talked about that endlessly recently with all these free uploading services and how they work. And then, is it going to be there when we need it? So the good news with the Authenticator, Google's Authenticator, is that it is not dependent upon them.

There are two approaches for this kind of device. There's the sequential one-time password and the time-based one-time password. The YubiKey is famously a sequential-based authenticator, and my little PayPal football is time based. Every time I turn it on, every 30 seconds it changes. Now, the model that VeriSign has adopted is the "we want to make money" model. And so anybody using VeriSign devices, that is, not the end user, but for example this PayPal football is a VeriSign dongle, and I've got also the VeriSign app on my BlackBerry. I love them, but I've had some conversations with people who were considering adopting the VeriSign solution, and they're really expensive. I mean, per authentication, PayPal is paying VeriSign some serious coin every time I use my little football to authenticate.

Google's model is different. Google has open-sourced this. And instead of what's called HOTP, which is the HMAC one-time password, which is the technology for doing it incrementally, and that's in RFC 4226 that was laid out, instead what I'm seeing becoming more popular is a time-based password. And that's what Google wants to use, expects its users to use. And the idea is that our phone knows what time it is, our computers know what time it is, and the authentication is inherently decentralized. If you're going to have a counter-based one-time password, some one entity, for example VeriSign, needs to maintain their copy of the count. And so by definition all authentication has to go through VeriSign. And if VeriSign were off the 'Net, nobody could authenticate. And that would be bad.

Similarly, Jon was wondering, he was assuming that Google was in the loop. And the point is Google is not in the loop. Nobody is in the loop. There's no loop at all. The Google Authenticator generates a six-digit code based on time, and anybody who has the algorithm and your authenticator's private key can generate for themselves the same thing Google authenticator shows.

Leo: That has to be. Otherwise LastPass wouldn't be able to use it.

Steve: Well, they wouldn't be able to use it for free. And they wouldn't want to be paying

VeriSign. They could support VeriSign's tokens. But they're not because, oh, boy, I mean, VeriSign's making money on that. And Google is saying, okay, this is not where we want to make money. We're going to enforce this. We're going to keep this open source. Here's all the source. It's going to be RFC based. Right now this time-based OTP, TOTP, is an IETF draft. And so it's being ratified. And actually VeriSign is participating in that process. So they're part of wanting to be involved in making this approach work.

And the beauty is that all an individual needs, there is also - Google makes the authentication side, a PAM module that does the authentication, a Plug-in Authentication Module. So anyone you wanted to authenticate to who supported the Google authentications solution only needs to know your device's secret. And that's part of setting up an account with them. Then they're able to make sure from then on that you actually have that device in your possession, a Google authentication device with the matching key. So it's a neat solution. I really think it's going to happen.

Leo: Is the secret tied to, like, the IMEI or something hardware-based on the phone? Or is it generated when you first run the app?

Steve: Generated when you first run the app.

Leo: Okay. So it doesn't in any way identify that piece of hardware.

Steve: Nope.

Leo: Except in the sense that a cookie would. It's created.

Steve: Correct. You need to share that with people who you want to be able to authenticate you. And in theory you could only share it with one person, and everybody else could use them. But the decentralized model of this is, I think, really what makes it go. And again, it doesn't have to be absolutely perfect to be much better than not having it at all. So again, we don't want the perfect to be the enemy of the really good enough.

Leo: Yeah. Fadele Adeolu in Nigeria, and I'm sure I'm butchering his name...

Steve: It's a good name.

Leo: I love it. Isn't it? He's been thinking about the diameter of the Internet: Steve, I work as an IT professional in Nigeria. You should know that many are benefiting from Security Now!, even in my part of the world. I was initiated into the Security Now! world in 2006. Since then it's been - excuse me, I had a sneeze there - addictive. I've also initiated many others who use the podcast as a learning tool. That is so great. So nice to know. My question is on the idea of Internet diameter discussed in one of the episodes you did on How the Internet Works. You mentioned, if we ever get more than 255 routers between two points on the Internet, no data would be able to get from one end to the other.

Steve: Right.

Leo: Because of what, it's an eight-bit descriptor?

Steve: Because the TTL is only a byte. And so there's no way, even if you set it up to 255, when it decrements to zero, routers drop the packet.

Leo: Interesting. Oh, that's really cool. I wasn't paying attention when you mentioned that. This very much caught my interest. I've been thinking about it. You didn't sound like there's any possibility of ever reaching this 255-router limit. You didn't sound worried. But I would like to know the extent of the limitation and if there's a way out. So I'd like to ask, if this diameter is limited by the one-byte header of TCP v4 that's used for setting the TTL, as you just described, how flexible is it to enhance the implementation to increase the diameter? What will the Internet diameter, for instance, be with IPv6? Steve, this Internet thing could be a mystery but for you guys who are committed to demystifying many topics. Your good job is really making significant impact on the Internet community. Keep it up. Bye for now. Ade. That is nice. Thank you, Ade.

Steve: So with every opportunity to increase the size of the field - which is now called the "hop limit." TTL kind of - it was never about time. But it sort of sounded like it was about time.

Leo: Time To Live kind of gives away that implication.

Steve: It does give you that impression, doesn't it. Now it's called the "hop limit" in the IPv6 spec. And it's eight bits.

Leo: It doesn't change.

Steve: Doesn't change.

Leo: So they, like you, are sanguine about this issue.

Steve: Yes. When I trace from where I am to GRC's servers, and for reasons of architecture it goes up to Northern California and then comes back down, it's, like, 10 hops. I mean, people have seen traceroutes. You've done traceroutes to places. And it's, like, seven. It's 12, maybe. But it doesn't scroll off the screen and keep going off, 255. We're just no - we are so far away from that being a problem that where they took the IP size from 32 bits and went to 128, this would be the time to increase the hop limit if you had any concern that 255 would ever be a problem. And, clearly, nobody's worried about that.

Leo: I hope they weren't shortsighted. I hope that they're, I mean...

Steve: That would be a problem.

Leo: I mean, we just went - we're going through this thing.

Steve: We've got so many IPs. Now we can't get to them. Errgghh.

Leo: Ade, now I'm worried. Peter H. in Wiltshire - as in "shear," not "shire" - Wiltshire, England, says what about LastPass? Steve and Leo, great show. Thanks, love it. I know you both previously have said good things about LastPass. But with all the talk about Password Haystacks, Latin Squares, and the pros and cons of frequently changing passwords, I've gained the impression that neither of you actually uses LastPass. No, I do. I do. That's all I use. Why? They've solved the problem so elegantly, and the job is done. I'm aware of the network traffic anomalies that LastPass detected and understand the implications and possible consequences and limitations of any impacts, and as such am still happy with them. Are there other concerns that I've missed, or do you simply prefer alternative apps, plug-ins, services, et cetera? I'd really appreciate some feedback, even if you don't select the question for the podcast, because I think LastPass is great and have persuaded family members to adopt it. So I don't want to give those near and dear to me bad advice, if there's something I've missed. Thank you. Nothing you've missed. Actually, you did miss something: why Steve did this.

Steve: I did it because I could. I did it because it was there. Seriously, Peter, I run my life on LastPass. And Leo does. I mean, it is so good, and it is still - it's what I use. But I just had an itch, and I had the question, could a paper-based crypto be created? And Off The Grid happened, using Latin Squares. So it's not at all that I think LastPass doesn't solve the problem. Actually I've talked about the two together. I would use Off The Grid to generate the password which LastPass would then use automatically for me, the advantage being that I always have the ability to go back to paper if I ever need to. I can have this thing in my wallet if I ever am without LastPass for some reason.

So it's not that they're really competing technologies. But in the case of Password Haystacks, that was like, okay, wait a minute, let's think about what - do passwords that are secure also have to be really hard? No. That was kind of a cool thing to realize. And for Latin Squares, it's like, hey, it's possible to do real good crypto with a piece of paper. So they were just sort of cool technology things. But in terms of what I use, I use LastPass.

Leo: It's just that simple.

Steve: Yup.

Leo: Yeah, and I do, too. And I use the password generation. It's just that - I guess

that Haystacks is one thing, and a good point. You know you could use, in fact I do, the LastPass-generated password and just add a standard haystack few characters to it to make it just that much more strong. So I do that. I have a certain set of characters that I use that I will never tell.

Steve: No. Don't tell anybody.

Leo: And the whole idea of Perfect Paper Passwords is if you're offline, or the Latin Squares is if you're offline. And LastPass, of course, requires you to be online to use it.

Mario Arce in New York wonders about sending Steve Twitter messages: I hear on your Security Now! podcast you say people send you tweets about security-related events. Once, a while ago, I tried to make you aware of something, but I could not send you a message via Twitter. Did I do something wrong? Or do you need to follow someone to be able to receive Twitter direct messages from that person? You do need to follow people to get DMed. But I don't think you are saying "DM me."

Steve: No. And I don't follow anyone on my SGgrc account because I can't. I mean, if I were to follow everybody who follows me...

Leo: [Indiscernible] all the time.

Steve: Well, or I'd be seeing 26,000 individual Twitter feeds, and nothing would happen. I mean, I'd just not even going to get out of bed in the morning. So Mario, to answer your question, the way people get things to me is they mention me in their own tweets. They just put an "at" sign, SGgrc, and then say whatever they want to say. They're actually tweeting that to everyone who follows them. But my feed picks that up as a mention of my Twitter account, SGgrc, and so I see it. And so there's a whole bunch of people who from time to time run across something they want to make sure I knew. And so they tweet themselves whatever they want to say and just mention, by doing @SGgrc, and I will see that also. So that's how I do it.

Leo: That's how you do it.

Steve: Works great.

Leo: Same thing for me. I read my - anything you "@" to me, I read that usually. Not religiously. Hey, we've got a bonus, Question 11 from Paul Bone in Melbourne, Australia. It's the Happy Camper of the Week story: Oh my god, oh my god, oh my god, spelled OMG OMG OMG. K2pdfopt. Thanks for sharing this on Security Now! 324. I'm visually impaired, having about 12.5 percent of normal vision, born cataract blind. And after having my cataracts removed, I developed glaucoma - oh, my god, I'm so sorry - have since developed other vision problems. I bought a Kindle DX with the hope that it would help me read scientific papers more easily. Of course that's

the big format Kindle. I'm working on my Ph.D. in computer science. Sadly, I found that zooming into PDFs was too clumsy, although I love the eInk screen and reading novels on it. Thanks for your recommendations, and I use Audible.

I've pretty much given up trying to read PDFs on my Kindle and have since been waiting for a tablet PC that does this well, maybe the Kindle Fire. But now I've heard about and used K2pdfopt, I'm so extremely happy. I found the website and saw an example where they'd converted a PDF into large print format. I was so happy that I might be able to read my PDFs comfortably - and believe me, there are a lot to read for this Ph.D. - that I cried a little. Inside.

Anyway, I know you've simply passed on a message rather than created the product. But I wouldn't have found out about it if you didn't mention it on Security Now! So thanks to you and to the listener who wrote in about it - K2pdfopt. And remind me again, that's a PDF converter? Is that what it is?

Steve: Yeah. It is a processor of PDF files that basically it strips out a lot of the formatting and does make PDFs readable on much smaller screens. You lose that - the PDF itself is a page-based layout. And so if you're going to show a PDF page on a small screen, you're sort of stuck because it's got to fit the whole page on one screen. It's a screen at a time sort of deal. And so this just sort of says, no. We're just going to grab the text and preserve what we can, but legibility and readability on small screens is our goal. And they really achieved it.

Leo: Well, there you go. And it's from Willus.com. Steve, we have completed your assignment, 11 questions good and true. And you've answered each and every one with your usual grace, verve, and savoir-faire.

Steve: Okay. And I'm exhausted.

Leo: @SGgrc, that's the Twitter account, @SGgrc, GRC being the Gibson Research Corporation, GRC.com being the website - certainly the place to go if you, as most people who use hard drives, are looking for the best hard drive maintenance utility of all time. SpinRite. Go there. Buy it.

Steve: Yay.

Leo: Yay. You also will find, as a bonus, many freebies, lots of great stuff. Just browse around GRC.com. And of course, if you've got a question for two episodes from now, there's a form there, GRC.com/feedback, is a great place. And the podcast's there. Now, audio only. Steve does, for us, he does his own 16Kb version for people who have bandwidth issues. So we've got a 64 and a 16Kb version there. Also transcripts, which Steve does. Thank you, Steve. We also have video available at TWiT.tv, along with all the other versions. GRC.com or TWiT.tv. We do this show every Wednesday, 11:00 a.m. Pacific, that's 2:00 p.m. Eastern, or 18 - now, I guess UTC hasn't changed, but because our time fell back it would be...

Steve: Right. We changed.

Leo: We changed. So we're now minus eight.

Steve: Right.

Leo: You figure it out.

Steve: As opposed to seven.

Leo: I think that means 1900 UTC, but I could be wrong. Yes, Sparky says 1900 UTC. So I will now add eight to the Pacific time. 1900 UTC at TWiT.tv.

Steve: And Sparky knows of what he speaks.

Leo: Apparently he does.

Steve: He's a good contributor over in the newsgroup.

Leo: Oh, good. Thank you, Sparky.

Steve: Paul Byford. Next week I'm with you, Leo, in the studio.

Leo: Yay.

Steve: There's a cool Internet identity conference on Tuesday, the day before, which I am going to be up in Northern California to attend. And I figured since I was, I'd see my family and spend the night and then be in the studio with you. And that will be the topic, will be what happened, what I learned, what I saw, what's going on in Internet identity. We'll be talking about that next week.

Leo: I think that's a really interesting subject.

Steve: Oh, it's the - it's THE challenge, as we all know, authentication.

Leo: Yeah. Wonderful, Steve. I'm glad we're going to see you. I'll order the burgundy right now. And we thank you all for joining us. We will see you next week, in studio, for Security Now!. Bye bye.