Listener Feedback #141

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-348.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-348-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is going to join us in just a moment. We're going to get updates on, boy, some big security flaws, both Mac and Windows, a big Windows update, a big Macintosh update. And then we'll answer some great questions from you, the audience. In particular, buffer bloat will be one of the many topics we'll talk about. Stay tuned. Security Now! is now.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 348, recorded April 11th, 2012: Your questions, Steve's answers, #141.

It's time, and there has never been a better time, for Security Now!, the show that covers keeping you safe online with our Explainer in Chief, Mr. Steven Gibson of the Gibson Research Corporation, the creator of SpinRite, the world's finest hard drive maintenance utility. He's also the author of a great many free security programs. And it's a Q&A time, isn't it Steve. Hey, Steve.

Steve Gibson: It is, Leo. It's great to be with you again, as always. Elaine, who apparently not only, as we know, transcribes these podcasts but actually listens to them, she immediately emailed me last week when you had referred to me as Explainer in Chief...

Leo: Yes.

Steve: ...and were trying to remember...

Leo: Right. What was the other word I used?

Steve: It was Debunker in Chief.

Leo: Ah ha ha. You are. You are the Debunker and Explainer.

Steve: We don't do that much debunking, but when there's something that needs it, we deliver.

Leo: When debunking is needed, the man is here.

Steve: So we've got - this is a Q&A episode - a bunch of great feedback from our listeners, some thoughts and comments and questions. But also, boy, has it been a busy week in security land. We haven't actually had a busy week for a while. But the big news over on the Mac side, of course, is that reportedly on the order of 1 percent, which is a big percentage, of all the Macs have been infected with this - it's variously known as either "Flashback" or "Flashfake."

Leo: Both of which are misnomers because it no longer works that way.

Steve: Correct. It started out back in September of last year, so September of 2011, with a fake Flash update prompt for users. And they had to provide their administrator password on Macs - this was all Mac. And so this used an exploit in order to get into their machine. And it was installing a botnet. Well, over the months it has evolved its capabilities and finally began using a Java exploit, that is, exploiting a Java vulnerability, which Oracle patched for everybody else back in February. But because Apple kind of went their own way with Java, I'm sure this was some screaming fit that Steve Jobs had at one point where it's like, we're not going to ship somebody - we need the source. We need to maintain this ourselves.

So Apple has their own Java build. And I don't know what the mechanism is for them maintaining synchronization with Oracle. But for whatever reason, Apple is running their own ship, Java-wise. But of course we know now they're dropping it. 10.7 no longer ships with Java. Users who need it have to go and install it themselves. So what happened was, as a consequence of this known but unpatched vulnerability, a huge number of Mac systems have been infected with this Flashback/Flashfake botnet. And the guy at Kaspersky Labs did something really cool. They reverse-engineered the latest version.

And it uses something we have seen before, I think it might have been Confickr, where it's a date-based cryptographic algorithm to generate domain names. So that the bots use the date and a sophisticated crypto algorithm to come up with a domain name which is predictable only if you had all of the information, so if you were a bot, or if you were a Kaspersky Labs reverse-engineering guru. So they understood the algorithm. They picked a date a few days in the future, determined that on that date, I think it was last Friday, in fact, the botnet would start looking for the domain, the dotcom domain krymbrjasnof.com.

Leo: What's that?

Steve: That's the result of the crypto algorithm. And so what Kaspersky did was they acquired that domain name and set up a honeypot.

Leo: You mean that the other guys didn't have it? I mean, isn't that where they control the botnet from?

Steve: Yeah, but they don't need it every day. So that one they hadn't grabbed, probably because they're - and the bots are always straddling several. So the bots are looking at existing domains that they have and also prospectively looking at new ones in order to be basically anti-shutdown tolerant. So the bots began reporting in on schedule at this domain name that Kaspersky grabbed. Thus they were able to exactly determine the count and location because of course they got the IP addresses of the queries coming in for these bots. They found, during their period of looking, 670,000 Mac machines. 670,000 machines.

Leo: That's going up. I mean, it was 500,000 when Dr. Web reported it, then 600,000. So it's still going up.

Steve: Yes. So a little less than half are in the U.S. They offer on their site a breakdown. It's sort of interesting: 300,000 are in the U.S.; 94, 95 if I round it, thousand in Canada. So Canada is second strongest with a little less than a third of what the U.S. has. Again, half again fewer, about 47,000 in the U.K. Then Australia has 41,000. Then it drops quickly - France down at 7,800 or 7,900, and Italy at 6,500, and down from there. So and it was funny because I'm looking at that: U.S., Canada, U.K., Australia. I thought, well, that's pretty much the demographics of TWiT.

Leo: Well, it's the English language speaking populace.

Steve: Exactly. When I saw that it dropped in France, I said, oh, of course, because this is going to be to a lesser degree an English-speaking attack. Then they used, and they were careful to say, "PASSIVE OS fingerprinting," meaning they didn't inject anything back onto the Mac machines or do anything active on the Mac machines that were querying, but they did passive fingerprinting to verify that, of those incoming queries, they could confirm 98.41 percent were Mac OS X v10.something. So that they confirmed. Then they did something very cool, which is - and you ought to click that next link there, the second one down that says "Flashfake Removal Tool and online checking site," Leo, just if you want to show...

Leo: Yeah, I got it, yeah.

Steve: ...our people. What they've done is, every query - the reason they were able to get these counts and to know that they were unique is the queries to this funky domain by the bots contain the Mac's own UUID, its unique identifying string that is absolutely

unique. So Kaspersky built a database of these 670,000 Mac machines. And any user who is interested can put their machine's UUID into this site and check whether it's logged as infected. So it's a quick way of confirming it.

Leo: That's interesting. I mean, it's easy enough to determine it with some simple terminal commands. People have written Apple scripts that execute those commands for you. But this would be another way to do it, I guess. Kind of a backend way of doing it. But you could see if you were ever infected, I guess.

Steve: Yes. FlashbackCheck.com is the domain. So FlashbackCheck.com with no spaces or dashes or anything, if you're curious. Now, they also have a removal tool. Kaspersky has a removal tool, small little thing, 164K zip file, which Mac users could use. There are, as you said, online instructions. They're a little complex. And so automating this with a single click is a nice thing to do. And Apple has said that - of course obviously Apple has responded to this, as you can imagine. They have patched the problem with Java, but only for OS X versions 10.6 and 10.7, not anything earlier. And they don't apparently plan to do so.

So anybody - and in fact they said on their page, "For Macs running Mac OS X v10.5 or earlier, you can better protect yourself" - yeah, better - "you can better protect yourself from this malware by disabling Java in your web browser's preferences." And of course listeners to this podcast know that we've been long recommending that everyone disable Java, or uninstall it if you really don't need it. But if nothing else, turn it off.

So this is a, once again, it's a JavaScript which then invokes Java in order to make this happen. So if you're also running with scripting disabled by default, you would have that first line of protection, and just doing that would have protected you. But then...

Leo: But of course Mac people don't assume that they're in any danger. So they probably don't run with scripting off; right?

Steve: Which is a great segue because there have been a lot of articles in the last week saying, wait a minute, we thought...

Leo: We didn't have to worry.

Steve: ...Macs didn't get, yeah, Macs didn't get infected. And so I look around, and I guess as I've mentioned before it's because I am in a university town with UCI right next to me, at Starbucks all I see is Macs now. I mean, there's been a huge shift, with Mac adoption obviously rising. And clearly I'm looking at a skewed demographic with college students who get a Mac when they go off to school as opposed to a PC more so these days than before. So there is a much larger target now for the bad guys. And where there's a known exploit, and maybe arguably, as you said, Leo, Mac users assume they're safe, so they're not going to be running with JavaScript disabled. Not like a large percentage of regular Windows users do that either. I mean our audience certainly does, but in general Windows users aren't.

So, yeah, I think we're seeing the inevitable catch-up, for lack of a better term, of the Mac situation. I mean, there is, unfortunately, there is nothing fundamentally more

secure about the Mac. It always was based on UNIX; whereas Windows, back in the days of 95, didn't have a secure OS security model. And even though NT did, it wasn't really used strongly until XP came along and Microsoft decided, oh, boy, with that Service Pack 2 release of XP where they finally turned the firewall on, and they really began to get serious about security.

I did read that the Mac support of ASLR, Address Space Layout Randomization, was less robust than Windows, and that when it was first released, Apple said several years ago that they were going to improve it, but they haven't done so yet. But I don't know that that's a huge issue.

But basically these are all PCs of one ilk or another. And we're seeing things like plug-ins such as Java or exploits in JavaScript and browser exploits. Those are now the low-hanging fruit. It's no longer open ports, which you could argue would differentiate one OS more from another because they have completely different internal architectures, Windows versus the Mac. Now it's the stuff running on the OS which is where our problems are coming from, and they're cross-platform. Java runs on everything. And clearly whatever it was that was wrong with Java was just as wrong under Windows as it was under the Mac. Oracle fixed it in February. Apple didn't get around to it. And you can imagine they wished they had now because this is a bit of a black eye.

Leo: It's embarrassing, yeah. Now, what happens if you get infected by this? It isn't a particularly destructive virus, is it?

Steve: No. It doesn't want to hurt you. It wants to use you. It wants to set up camp in your machine, check in with these wacky domain names from time to time, and participate in spamming and DDoSing, the standard botnet stuff. So it wants to commandeer your machine. It wants to use...

Leo: Although, because all of these servers that it's supposed to hook up to have all been kind of closed or taken, it's unlikely to do anything if you've got it. It can't join those botnets.

Steve: Now that it has become as high-profile as it is, there's a huge effort now to shut this down. Apple doesn't yet have, but has said they will have, a removal tool. I imagine next week we'll be talking about that because I would think they would do it quickly since everybody else already has for users. But there will be an official one from Apple. And all it'll - it just removes the required files from the OS, and then you're fine. So it's not some horrible pernicious rootkit-y kind of thing that you can't ever get rid of or you have to reformat your hard drive and reinstall your OS or anything. It will leave peacefully.

So Microsoft, also, we have just passed our second Tuesday. And, oh, another big one. They did six bulletins, four of which were critical. But one is super critical because of its pervasiveness through the Windows ecosystem. There's something known as Common Controls, which have existed from the beginning of Windows. We're familiar with how Windows has sort of always been modular, with so-called DLLs, Dynamic Link Libraries. Windows itself has a kernel DLL and a GDI DLL and a user DLL. So it's built from these modules, these dynamic link libraries.

Well, a library which has always existed in Windows is called Common Controls. And those are the things that all Windows apps have and use, like the scroll bars on the side,

horizontal and vertical, and the menu system itself, and the individual window control widgets, the dropdown list box and the spin dialogues, and even text fields which appear in a dialogue, all of that is common control. That is to say, all of the chrome that embellishes a window is essentially the common control. You have the window frame, and then all of that other stuff. Buttons, obviously, are common controls also.

Anyway, all of that is in a library which, over the years, has been evolving. And at one point they switched their technology from DLLs to so-called ActiveX controls, where then the extension changed from .dll to .ocx. So what was found - and let me get the timing on this right. Microsoft has known that there were some exploits of this for a while, but kept it quiet. So it's being called a zero-day flaw, but I don't really know that that fits so much.

But the problem is that the incredibly highly used ListView and TreeView are two controls in the current common control library which can be exploited. And so of course the TreeView is what everyone's used to in Windows where it's like that hierarchical view with the plus signs you click, and it opens up. It's that outline-y kind of thing. That's a common control in this library. And then the ListView is anything that is like a list of things, like a spreadsheet is a list and uses that common control library. So there's a problem in that in every version of Windows.

And it's worse than this, though, because developers want to make sure that their software running on some random Windows machine has the most recent common control, or the one that they want, because sometimes these things change, they've got bugs and so forth. The developer designs an application, and even Microsoft does this, designs an application with a given version of the common control. Well, they bundle it with their application because what Windows will do is it will look in the directory where the executable ran from for various things that it needs before it goes out and looks system-wide, in the Windows system directory and the Windows directory and so forth. So if an application brings along with it the version of the common control library that it wants and expects, then it knows that's the one that will be loaded for it when it runs.

But what this means is that this isn't just Microsoft that needs to fix this problem. This is everybody, all applications that have bundled this now known to be buggy common control are potentially vulnerable. Now, we have to mitigate that somewhat because of course the bad guys would have to get this application to invoke the TreeView or the ListView somehow themselves. And that's much less likely to happen. The big targets of opportunity, of course, is IE, Internet Explorer. It's using those common controls. And Office is doing so, too. And this can be either exploited by IE or, naturally, by clicking on a link in email that invokes this common control which is present.

So we're recording this on Wednesday the 11th. This all happened, the updates, on the 10th. There are mitigating measures. For example, in Microsoft's own FAQ they said, under their FAQ for this, they ask themselves the question: "I'm a third-party application developer, and I use the ActiveX control in my application. Is my application vulnerable, and how do I update it?" Microsoft's own answer is: "Developers who redistribute the ActiveX control should ensure that they update the version of the ActiveX control installed with their application by downloading the update provided in this bulletin."

So this is Microsoft formally acknowledging that everybody who is sending this out on their own has to take some responsibility. But again, it's not clear how the bad guys would get to that particular instance of the bad control. For anyone who's curious, this is a file called mscomctl.ocx. And I wouldn't be surprised, if you just searched your file system for it, if you find a bunch of them littered around because different applications will have brought them. Under workarounds, Microsoft said: "To prevent the vulnerable

Active X controls from being run in Internet Explorer, you can disable attempts to instantiate the mscomctl TreeView, TreeView2, ListView, and ListView2 controls in Internet Explorer by setting the infamous kill bit for the controls in the registry."

And we've talked about the kill bit for years because this is an area where Microsoft is beginning to get better with their security, but everything used to just be enabled by default, so it's touchy to come back in and turn things off, even though now they wish they hadn't been as liberal with allowing everything to run. The problem is, that is with doing this, is that some websites really do want to use, well, at least the ListView, if less commonly the TreeView control. So that would break those sites that do that.

So bottom line is it has been fixed. This is an important Patch Tuesday. So don't wait too long to update this and restart your machine. When I fired up my Win7 box in order to run Skype for the podcast, Leo, of course it immediately popped up and said, whoa, we've got important things to do. And I said, yes, update yourself right now. Get it done.

And not to be left out, Adobe, also yesterday, released new versions of Reader and Acrobat. There's no end of - and I should mention, I didn't quite finish saying that this is being, this zero-day, if it's so-called zero-day, this common control problem is being actively exploited in the wild. So this is important. This is going to be jumped on by the bad guys. So again, it's not theoretical. I remembered to say that because the fixes that Adobe has just offered for Reader and Acrobat are like, yeah, update, but nothing horrible happening with them as far as we know.

Reader and Acrobat 10, versions 10.1.2 and earlier, need to be updated for Windows and Mac; v9.5 and earlier, also for Windows and Mac. And then over on Linux, since there's no Acrobat there, it's just Reader v9.4.6 and earlier need to be updated on Linux. So keep that in mind for Reader and Acrobat users. I switched over to, I think it's Sumatra, as my plug-in, my PDF plug-in for Firefox. And I like it very much. I prefer it. I have Acrobat installed on this system, and so Acrobat brought its own plug-in. It doesn't use the Reader plug-in, it uses the Acrobat plug-in. And it was, I don't know, always bothering me with things, trying to launch a separate window and doing strange things. So finally I got tired of it, and I disabled it, and I looked around, and I used Sumatra on Firefox. And I'm very happy with it.

Oh, and I did want to follow up that last week, when we were talking about iOS password managers - we've got some questions, not surprisingly, in the Q&A approaching. I mentioned that 1Password, which you specifically asked about, Leo...

Leo: It's I think the most popular Mac program, yeah.

Steve: Yes. And I mentioned that they were just on the cusp, I had read, of increasing its security. And they have. On April 9th, which would make that Monday of this week, they released 3.6.5. And the blog posting is "1Password iOS PBKDF2 Goodness." And of course we know that PBKDF2 is the password-based password-strengthening algorithm. And so under their list of things they have improved, they said, "Improved security. Now using 10,000 PBKDF2 iterations to protect the encryption key. Dropbox authentication tokens are now stored in the system keychain. Better support for iPad retina display" - on iPad 3, of course - "and improved log-in filling and some bug fixes." So 1Password, as you say, Leo, the most popular password manager over on the Mac side, just got better.

Leo: Yay.

Steve: And I wanted to let our listeners know that. Also many people have tweeted me about a recent Dropbox tech blog. It's tech.dropbox.com/?p=165. And you can probably also, by this time, Google. They've used a sort of a funky pseudo password, tongue in cheek. It's "zxcvbn: realistic password strength estimation." Well, this is very interesting. So many people have brought it to my attention, I just wanted to let everyone know I'm aware of it. This is the Dropbox guys commenting that more and more they are seeing password strength meters wherever they've being asked to create a login on a website.

Leo: Yes, yeah, you see it all the time, yeah. And I'm never good enough.

Steve: Well, and what they've done is they have developed a very nice one and offered it. It's like, here it is. And so for next week I will have a complete analysis of it. I thought I would have time today, but as I was scrolling down through it, it's like, whoa, I'm going to have to read this, I mean, really think about it because it looks like it's extremely comprehensive. And it looks like they've done a great job. I salute them because this helps other developers. Down in the comments to the blog posting there's a lot of guys saying, whoa, thanks so much, I'm going to use it.

Leo: I could use this, yeah. As long as it's reliable. As long as it's, you know, it reflects accurately the true strength of the password.

Steve: Correct. And that's why I can't say anything about it today. I'll have a complete, an Explainer in Chief evaluation of it for next week. And, really briefly, I just wanted to mention something that might be of some interest to our listeners. Leo, I know you, I had heard you on some other podcast talking about genetic testing.

Leo: Yeah. I did 23andMe, yeah.

Steve: But it's now $209. Wasn't it a lot more?

Leo: It's actually - is it really 200? Because I thought it was closer to a hundred. Anyway, yes, it's a lot cheaper.

Steve: Yeah, so...

Leo: And it was worth doing. Although I will say that it is not the full genome. It's not as cool as one would hope. But it's a start.

Steve: I had thought that it was more like a thousand dollars.

Leo: It's $99 for the test. And then they encourage you to subscribe for $10 a month because then they will notify you - and I think you have to do it, so you're right, it's $200 when you include the $9 a month subscription because then they notify you when they find out new stuff about any of the genes that you have.

Steve: Right. So anyway, I didn't realize it was so inexpensive. I immediately ordered it, and I'm waiting to get my saliva kit so I can spit in it and see what they can tell me.

Leo: Yeah, it's really easy. In fact, I should show you, for those who are interested, I'll give you an idea. I'll show you my page. One of the things that they do is they give you surveys because they want to match your phenotype with your genotype, so your history, your sexual orientation. They have a cancer family history survey. So they encourage you to take these. But if you want to then look at your disease risk, for instance, based on genetics - and remember, we don't know exactly what the connection is between certain genetic markers. My risk of coronary heart disease is 58.1 percent, but that's because I'm a white European male. So men of European ethnicity...

Steve: Well, for example, it says nothing about your diet. And so it's like, clearly...

Leo: As we know, most disease is systemic, and it's a combination of genetics and environment. But it's interesting. It's great. And there is some stuff that is great. For instance, here's carrier status. These are genetic markers that you might pass on to your kids, things like Tay-Sachs disease. And then drug responsiveness, traits, whether I'm likely to become bald. It's fascinating. My alcohol flush reaction? None. I am unlikely to taste bitter perception. I have wet earwax. My eye color, likely brown? Yes. My hair curl, slightly curlier hair on average, absolutely. Likely lactose tolerant, yup. That's probably because I have some Nordic or Germanic heritage. Malaria resistant, not resistant. Male pattern baldness, typical odds. Muscle performance, I'm a likely sprinter [laughing]. Sorry. You could see how some of this stuff doesn't - is more environmental.

Steve: Yeah. I just - I didn't realize it was so inexpensive.

Leo: Yeah, and well worth doing, as long as you understand it's not 100 percent complete, it's not your full genome, it's a few markers. But it's really - I think it's great. It's well worth doing. And if you get your family members to do it, get your mom to do it and stuff, then you get additional information.

Steve: Okay, Mom. Spit in this vial.

Leo: Yeah. Why not?

Steve: Okay. So two little goodies from the Twitterverse. Kyle Skrinak, who's in Apex, North Carolina, and he's @skrinakcreative, he said: "@SGgrc, the OpenDNS DNSCrypt

uses 112MB of RAM on my Mac. Does that sound resource-hungry to you?" It's like, oh, 112MB. Well, yeah. I look at that as a fraction of a gig, and it's a large fraction of a gig. It's, wow, 11 percent. So it's like, yeah, that seems like a lot. I don't know where it stands in its development release. I guess was it in beta on Windows and released for the Mac? I don't remember know where it is. But wow, that's a lot of memory to give it. So maybe they'll work on paring that down.

And then Mark Cipriano in Australia, he said: "@SGgrc Question. How can I store important docs, e.g., birth certificate, etc., online? Encryption? Dropbox? Evernote? Can this be done safely?" And it caught my eye because next week we're going to talk about SpiderOak, and I can't think of the other one. There's two of them.

Leo: I use SpiderOak, and I really like it. Perhaps Wuala?

Steve: Oh, BoxCryptor is the other one I wanted to...

Leo: Box, okay, that's Box.net, yeah.

Steve: SpiderCrypt or...

Leo: SpiderOak.

Steve: SpiderOak, yes. SpiderOak looks very nice, and BoxCryptor. So I'm going to do the full, tear it down, look at the technology. I've established a dialogue with the SpiderOak people, so they're standing by to answer my techie questions, which I'm sure I'll have, and I've already got some, in fact.

Leo: I have some.

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

Leo: It's got a little bug. I have a SpiderOak account. I think I pay for 200GB. I pay for a lot. And for some reason, on one of my machines, even though it's on the same account, it doesn't recognize the other machines. It's kind of weird. Anyway, I'd be very interested. SpiderOak is one of those that offers pre-ingress encryption.

Steve: Yes, that's what caught my attention. And I like it very much, just from a philosophical standpoint. They are rigorous and ruthless about individual security. I mean, they bend over backwards making it absolutely clear that this, as you said, is pre-egress, or pre-Internet encryption, that they do not have your password. They don't want it. They're going to go to every length possible not to be able to ever decrypt what you provide. So it is encryption, and it really looks like it is secure cloud storage.

Leo: I've been pretty happy with it.

Steve: Yeah. And then BoxCryptor is different somehow. I don't know how, but I will next week. So I'll have the full tune-up there. And I did get a nice note from a William Lorman, and he said - his subject was "A classic SpinRite story with an iMac twist," which is nice.

He said, "Hi, Steve. I'm a long-time follower of your work and customer since 2006." And he sent this on April 2nd, so this is recent. He said, "I just had a scenario I thought you and Mac users would appreciate. One of my customers' iMacs would not boot, and I could not get it back with two well-known Mac tools that reported hardware problems. This happened when the customer forgot to configure the new disk they attached for Time Machine, so they were two weeks since backup. I took the Mac to the Apple Store for a warranty claim, and the 'genius' gave me a not-so-nice response when I asked for the bad drive back. He said, if the Mac apps I used didn't work, nothing would. I said, 'I want to try SpinRite.' He had never heard of it and said there is no way a PC product could work on an Apple part or recover an Apple system."

Leo: Oh, dear. Not such a genius.

Steve: It is in quotes here. "I put the drive in a PC, booted SpinRite, and I'm guessing you know what happened from there. SpinRite Level 2 got the drive back. I mounted it in an external enclosure, and the Mac's Migration Assistant program got the user's whole world right back where it was before it crashed."

Leo: Wow. So they did give him the drive, they just were skeptical.

Steve: Yes. "The user was very pleased to become one of your customers" - of course I thank you, William, for encouraging that, since he did get the benefit from it - "and claims they will never forget to configure Time Machine again. Sincerely, Wm. Lorman." So, neat story, thank you.

Leo: I love that. That is a great story. You don't look at the file system. You don't look at the operating system. You don't care if it's HFS+, NTFS, FAT32. You're not looking at that level.

Steve: Or TiVo.

Leo: Or TiVo, right. Which I think uses the Linux file system.

Steve: It does, except some of them are byte swapped because they were PowerPC based, so they were big-endian.

Leo: Whoa. They're big-endian.

Steve: Yes, exactly.

Leo: Wow. That's interesting. But so you don't care because you just look at sectors.

Steve: Don't care.

Leo: You're asking the drive - the only thing, the reason you don't have a Mac version, I know, is because you use an interrupt that's only in the PC BIOS. You haven't done the EFI version.

Steve: Right. A lot of people want it. And it's definitely in our future.

Leo: Nontrivial, I think.

Steve: Nontrivial.

Leo: You kind of have to write your own INT 13, I think, is what you have to do.

Steve: Oh, I can't wait.

Leo: It's not hard. I bet you INT 13 is just a few lines of code. I mean, how complicated could it be? Steve, for you, I have questions.

Steve: Yeah. We got some great ones.

Leo: Let's get right to them. Oh, I closed them. And that was a foolish thing to do. Let's reopen them and take a look. Security Now! Q&A 141. Question 1 from Shane in Phoenix, Arizona. Is secure deletion necessary when using full disk encryption? Oh, you know what, that's a great question: Steve and Leo, first off, I love listening to Security Now!. It's great. I have a quick question. I use a MacBook running Lion. I've enabled full-disk encryption with Apple's built-in FileVault feature. Given all that information is stored as encrypted gobbledygook, is it necessary for me to use OS X's secure delete tool when erasing sensitive files? Namely, if my encryption key were compromised, would it be feasible to recover encrypted data that has been erased? My first impulse is it's not feasible, since any information on the drive is just noise. But I defer to your expertise on this one. Thank you. Shane.

Steve: Well, it's interesting because - and it's an important question because the answer is yes, you still do need secure deletion.

Leo: Oh. That's not what I would have thought.

Steve: And the reason is, while all of the data is stored on the disk as scrambled bits, when viewed from inside the operating system, it just looks like a file system because the encryption occurs on the way out, and the decryption occurs on the way back in. So if regular file deletion is what typical OSes do, that is, they just mark those regions unused now, they're undeletable. So you can undelete files even if it's on a secured encrypted file system because, again, the files stored are gibberish. But the operating system, which is on the inside, sees everything decrypted. So what secure deletion does, of course, is it overwrites the file so that the data is actually gone. And then you're fine. But you definitely want to do that even with full-disk encryption.

So to be clear, Shane said "if my encryption key were compromised," meaning if the full-disk encryption was decrypted, do I wish I had been using secure deletion, and the answer is yes because any file that can be undeleted, or the disk scanned from inside the operating system to find debris from temp files and other things that were not securely deleted, those are still available if you're on the inside. Which is where you'd be if your encryption key were compromised. So, yeah, it's still worth doing that.

Leo: Interesting. Mark Martin, Lansing, Michigan, has a follow-up from last week: I went looking for more information about the ElcomSoft presentation at Black Hat Europe that you mentioned on our last episode. I found the slides at media.blackhat.com. Actually, I've clicked this link, and I don't think they're there. But anyway, in them they discuss the complexity of the master password validation. And a table at the end summarizes this again, includes rates for master password validation, for all password managers, and the length of passcode that can be exhaustively tested in 24 hours. I was surprised to see that LastPass had an "average" placement on the list. Given that they determined that a 12.2-digit passcode could be tested in 24 hours, that would seem to indicate that we'd want a longer passcode to become more resistant to offline attacks. Are their conclusions about LastPass accurate, and should I start memorizing my 12 characters of entropy and pad it out to about 32 to feel safe?

Steve: Well, there was a table in the original document which I used as my reference for last week's podcast. And you're right, Leo, I tried that link, too, and it...

Leo: It's dead.

Steve: ...doesn't look like it's there. What's a little misleading is that they're talking about digits, meaning specifically zero through nine. And even in Mark's own text here he says 12.2-digit passcode, then later he says 12 characters. So as we know, the strength of a passphrase is a function of the size of the alphabet. So, for example, if your passphrase only used digits zero and one, then each character position can only have two states, and so it's an alphabet size of two, so you'd need a really long passphrase in order to get enough possible states.

So most users are going to be using an alphanumeric passphrase, in which case you're about 10 times, you're roughly 10 times stronger because you go from an alphabet of 10 to an alphabet of not quite a hundred, but like 94 or something. So then the number of

digits, instead of, like, 10 to the power of how many characters, it's 100 to the power of how many characters, and vastly stronger. So I didn't talk about it last week specifically because it was like, okay, well, that's not really relevant to most people. Most people are going to use both a complex and a sufficiently long passcode.

Leo: Right. And I did find the link. Actually somebody in our chatroom, who was probably the guy who wrote the email, just passed it along to me. So I guess it was just a typo. If you go to media.blackhat.com, they'll give you an XML - a link to an RSS feed that has everything, and you can just do that, and you'll be able to find the announcement or the slides, if you're that interested to read them.

Steve: Well, and for anybody - there was a lot of interest, you can imagine, in last week's topic. And so there is great information there. So I would encourage people who want more than they got on the podcast to go there.

Leo: If you want more, we've got more. Oops, I did it again. I closed the PDF. One more time. Open her up. Questions. Here we go. This is Question 3. Steve C. in Rochester, New York says: Do mobile devices have a built-in firewall? Steve and Leo, thanks for the great podcast. I've been listening for about four years, and I really enjoy the show. I have a question about iOS and Android mobile devices. Do any of these phones or tablets have a software firewall as Windows does? I've been under the impression that, if I'm using an iOS mobile device to connect to the Internet over a public unsecured WiFi hotspot, that I'm safe as long as I do simple web browsing. That is, I'm safe as long as I don't attempt to log into a secure website over such a connection. Is there any way that an attacker can inject malware into my device if I'm logged into a public WiFi hotspot? Thanks, Steve C., Rochester, New York.

Steve: Well, that's a kind of a complex question.

Leo: Yeah.

Steve: One thing to remember is that I'm sure without exception any WiFi hotspot is also a NAT router. So it will have one IP on the public Internet, and it'll be distributing private IPs, probably either 10.x.x.x or 192.168.something.something. So you automatically get the benefit of the NAT router being a hardware firewall that prevents unsolicited incoming traffic. But that shouldn't make anyone feel safe for two other reasons, which is that most exploits that we're seeing now are not our grandfather's Internet exploits, where there were open ports with bad services running which bad guys could send packets to and take over your computer. Most of them are like what we were talking about at the top of the show, where there's something wrong with your particular software in your own mobile platform, and you click on a link that takes advantage of that. So there you're going out to a site and essentially asking for trouble without knowing it. So software firewalls, while present, aren't giving us any protection.

And then the last aspect of this is, in an unsecured WiFi environment, remember that everybody is on, essentially, that unencrypted Ethernet. And so it's not just remote bad guys, but many of the things we've talked about, for example the infamous Firesheep, which allowed people to trivially find other people's Facebook logons and so forth, before Facebook and Twitter and Google began bringing up SSL all the time, which increasingly

is being done, really thanks to Firesheep representing such a threat.

So the dangers are multiple and various, not only from someone outside, who probably can't get in, thanks to the fact that the WiFi hotspot is also a NAT router, but mostly it's from people right there sitting at the table next to you or across from you. They have access to your network traffic. If you're using login credentials with a nonsecured authentication cookie, then you are immediately hijackable, unfortunately. But more than that, it's things you do that are leveraging defects in your software which represents today the greatest danger.

Leo: An anonymous listener - we have many of those. In fact, you're all anonymous unless you tell us. An anonymous listeners says, actually asks about iOS's built-in Password Safe. Did you know? Steve, a quick question about iOS and passwords. Safari is always asking me if I want to remember my passwords for websites. That's even on the iPad, the iPod Touch, the iPhone. Sometimes this is the most convenient way. Even as a LastPass user, typing my master password is bad enough on a PC, let alone a mobile device. But how secure is it? Should I say no?

From your last show it seems like Apple's doing almost all the right things to protect my data. If both use secure encryption and good implementations, then the only real difference comes down to LastPass requiring your master password, which adds another layer of authentication. But LastPass also lets you tell it to remember the master password, which pretty much just means you're now invested in never losing that device, or in being confident the device could never be hacked or broken into.

Considering all that, is having Safari remember my passwords any less secure than having LastPass on my iPad and having it remember my master password?

Steve: That's a great question. And I came away from last week's analysis of the most recent iOS-based devices, that is, everything from the iPhone 3GS forward, where Apple implemented hardware-assisted AES encryption and built unavailable keys, the keys that are - and I've verified this since - unavailable to the software in any way. You can use the encryption, but nothing on one of these more recent Apple platforms, meaning the iPad 2, iPhone 4 and 5, are able in any way - wait, is there an iPhone 5? Is it iPhone 4? It's 4. Wait. What's the latest iPhone? I don't remember. Anyway, none of these things are able to access the keys that are stored in the hardware, embedded into the hardware and unique for every device.

So I'm very impressed with this. What this means is that my feeling is it's not clearly less secure as long as you understand that you don't want somebody to get a hold of your device, and you need a good code to protect access to the device's UI. We now know that Apple ties that into the encrypted file system so that everything is being encrypted on the device, and it's tied to you entering that code correctly, and that they use good password strengthening to slow down cracking. So maybe the right thing to do is to consider a compromise, consider that Safari's own password storage is secure, Apple has done a good job of encrypting it, it's in the encrypted keychain, I mean, it really is safe from somebody who doesn't have access to your device.

What you obviously need to do then is use the complex login, the complex entry screen, not just a four-digit passcode. I still don't think that's safe. I'd go for the full keyboard and do something that you can easily do every time, but it automatically means that an attacker is going to have a much worse time. And assuming that they're doing it from the UI and don't have some sort of jailbreak or some way around it, the machine is going to

wipe itself.

And that was the other reason that Apple implemented this, by the way, in hardware is that it means wiping is instantaneous. Apple simply has to wipe the decryption key on the file system, which it can do instantly, rather than having to physically overwrite all of the flash memory in the device, which as these things become 64GB and bigger, can take a long time. So I really think it is safe, with the caveat that you make sure that getting into the device is going to be difficult, that is, in terms of getting past that opening screen. At that point, I've been very impressed with what Apple has done. I think it's very safe.

Leo: That's great.

Steve: Yeah.

Leo: Because I would like to say yes when it asks. That really is a convenience.

Steve: Yeah, it is. And I'm very impressed with Apple, I mean, they've really taken this seriously. So I wouldn't hesitate.

Leo: Good. Robert Berry, North Carolina, has a great tip and a note about Windows Defender Offline: Not sure if you've mentioned this, Steve, but I thought it might be worth letting your listeners know that the Microsoft offline malware removal tool, which used to be called System Sweeper, is now out of beta. However, it's been renamed Windows Defender Offline. I had some difficulty finding it because of the name change, so I thought it might be a good idea to spread the word. You mentioned it, I guess, some episodes ago, 303, last year. So search for "Windows Defender Offline."

Steve: Yes. And you will immediately find it. I verified that. And just to remind our listeners, it's a nice tool because it is bootable.

Leo: Oh, so you can make a disk out of it.

Steve: Yes. You can either USB install or a CD. So you boot it, and it's a standalone, outside of Windows malware remover, which is great for rootkits because it's before the rootkit has a chance to get itself set up and hidden from the operating system. So this is the offline, meaning outside of Windows, scanner from Microsoft. And of course it's free. So...

Leo: Yeah, Paul and I talked about this, I remember now, in December.

Steve: Right.

Leo: And he has an article about it on the SuperSite for Windows. And it is, I guess, still in beta. Let me go there and see. Oh, it wants me to log in. Never mind. I'll leave that as an exercise for the listener.

Steve: Yeah, it is out of beta now.

Leo: Oh, okay.

Steve: And thus the name change.

Leo: Got it.

Steve: It was System Sweeper. Now it's Windows Defender Offline. They're keeping it with the Windows Defender name, which they've already established. But offline, that's what they mean by "offline" is not running under Windows, running outside of Windows. You boot it in order to scan your machine.

Leo: And that's the best way to do it, obviously.

Steve: Absolutely.

Leo: Adam Jenkins, Question 6, he wonders about legacy iOS file system encryption. You've got a lot of iOS questions because of your piece, I guess, last week: Steve, are you sure? Are you sure that upgrading from iOS 3 to 4 to 5 would not have encrypted the device's file system? I'm pretty sure upgrading between major iOS versions has always required a full wipe and restore of the contents, which would have allowed for that encryption to take place. Only minor updates are ever done in place.

Steve: Well, Adam and everybody else, I'm not sure. I went back and tried to find the reference, which I'm sure I encountered during my research for last week's podcast, where some security types had said that, if you didn't wipe and restore under the iOS 4 or 5, that is, if you were at iOS 3, which was not encrypting, that your file system would never be encrypted. But I couldn't find the reference again because I wanted to see if I could learn anything more about it in order to answer Adam Jenkins's question more definitely. So I've got to say I'm not sure. What Adam says makes sense. And in fact, if as he says, a major version update does require...

Leo: I think that's the case, yeah.

Steve: ...a wipe - it certainly sounds right to me. And it feels like that other would have been - what I said last week would have represented a loose end that Apple would not have allowed. I mean, it's hard to understand why they would have done it, although

there wasn't hardware encryption support until the 3GS phone and the later iPad. So I wonder about encrypting the file system. I guess they do, even if you don't have it, if you've got iOS 4 or 5 on older hardware. But still, I've been unable to find anything definitive. So I wanted to back away from what I had said and not scare people because I can't confirm that.

Leo: It does stand to reason, I think, that if you wiped the disk during a major upgrade like that, that in the process of wiping the disk the encryption's going to be turned on when it restores; yes?

Steve: Yes. That's what I would think, too, that when you are installing - both 4 and 5 support whole file system encryption. And so you would think that that would - 4 or 5 would come alive; they would turn their things on. And then when you restore from iTunes - and we know that the export would have been encrypted by the phone, so on import it would be sucking it in and storing it in that encrypted format, backup encryption being different from local file system encryption. But so it would translate from the backup encryption to the local file system encryption, and there'd be a lot of encryption there.

Leo: Yeah.

Steve: I think it sure seems like you'd be safe. But I wasn't able to confirm that, so I didn't want to leave people with the idea that they might not be.

Leo: Unknown. Josh in Michigan comments about the security of the LAMP stack, the Linux Apache MySQL PHP stack. That's how most web servers run, they run LAMP stacks, including ours. Regarding PHP and MySQL being insecure by nature: As a web developer myself, I profit more from finishing a project quickly, and therefore I am not inclined to salt and hash passwords, verify that file access scripts don't traverse file systems, parse uploaded data for cross-site scripting, check variables against respective types, determine if TLS and modify session cookies accordingly, use prepared SQL statements, and so forth. But I do each of those, and many more, anyway. The security of the code is simply a side effect of how much time a project environment-aware and security-conscious development team is allowed to design and focus on the project. By thinking like a hacker, we work to prevent the successfully manipulation of PHP's environment.

While I'm at it, I'd like to mention that MySQL allows for prepared statements forcing all input parameters to explicitly behave as strictly data, even if the parameters contain MySQL injection code. It's just more difficult, and it takes longer to write that way and test, so most people don't bother. I thought I'd mention that at least the capability is there, even if it's rarely used, because it seemed to really trouble Steve that SQL statements can be modified by user input. Thanks for another terrific Security Now!.

Steve: I'm glad that Josh is putting all that effort into the security of the sites that he creates. As he enumerates all the things that he has to explicitly do, which he knows to do, it brings to mind the question, well, what about a web developer who is less security conscious, who doesn't know that all of those things have to happen because they all

represent tried and true and previously exploited approaches or exploitable approaches to securing a site. I've seen some comments from people yelling at me that PHP is no worse than anything else, and I completely agree. It's not PHP, it's the environment that we've created. And I liked Josh's question because it enumerates nicely how difficult it is, but how necessary it is to do all of that in order to lock down any website. And he mentions, yes, it's a function of how much time we're given. And so I would tell any managers of developers, please, give them as much time as they need to make it secure.

Leo: Well, it's also - it may be necessary, but there also is the question, is it sufficient? The presumption that he's making is, well, if I do all these things, we're secure. And as we well know from history, it's not possible to be a hundred percent secure. So it may be - he's actually sounding a little cocky to me, like, well, I do all these things, so I know my sites are safe. Boy, that sounds like a very dangerous attitude to take. It's necessary. Is it sufficient? I don't know.

Steve: Yeah.

Leo: Nathan Long in Charlotte, North Carolina wonders: Doesn't coding in assembly limit SpinRite? Hey. Hey, Steve. What are you doing? I'm a programmer at the opposite end of the spectrum. You work in assembler and mentioned that using JavaScript is really high-level for you. I work in Ruby and have just started dipping my toes in C, which seems very low-level to me. So it may be my question's a bit nave.

I'm wondering whether coding SpinRite in assembly placed constraints on what systems it can be used with. Doesn't assembly refer to chip-level details? Can SpinRite be used on any x86 system? Can it work with RISC processors? Putting myself in an assembly programmer's shoes, it seems like the ability to write C and compile for lots of different machines would be an amazing advance. And it was seen that way in 1979. I added that. That was me editorializing. Thanks for entertaining my question and giving any insight into your tools of choice.

Steve: Well, Nathan, let me tell you. I'll put it this way. When I heard that Mac was dropping the PowerPC for the Intel x86 platform, I was delighted.

Leo: Yes.

Steve: Because it meant that SpinRite could move largely unchanged over to the Mac. It didn't happen the next day. Obviously it still hasn't. But it means that it is entirely feasible. I need to just deal with the program's interaction with its environment rather than the program itself. So you're completely right. I like coding in assembly language, and that is absolutely tied to the processor which the code runs on. You have no processor independence, as it's called, which is really exactly why C was created. And you're right, it is very low level. The developers of C, Ritchie and...

Leo: Kernighan.

Steve: Kernighan, yeah, Kernighan and Ritchie. Those guys set out to create the smallest layer above assembly language because they had coded the first UNIX in assembly code. I mean, it was written in assembler. And they said, okay, wait a minute. Now, if we do that, we're tied to the chip. And they didn't want UNIX to be tied to the chip. They wanted to be able to more easily move it around to other architectures. So what's the smallest thing we can do to make us independent of the assembly language, the machine language underneath? And that was C. There was a predecessor, BCPL, which was the language before. And so C came after B, and that's the one that stuck.

So SpinRite will never run, I think it's safe to say, on a non-x86 system. The good news is, Intel won that battle. The non-x86 platforms are the mobile portable devices that are running the ARM architecture. And we talked about the Advanced RISC Machine, ARM architecture, many podcasts ago. But they don't really have a need for SpinRite today. So I'm glad that our desktop systems have ended up being x86 based, and SpinRite can run there. Whew.

Leo: Whooo!

Steve: Yeah.

Leo: Well, it's kind of ironic because of course one of the reasons you work in assembler is so that you can work to the bare bones of the machine. And it is CPU-dependent for that reason. It's the double-edged sword of it. But you, when you wrote SpinRite, actually created a dependency on code in BIOS, as we talked about earlier. And that's - I'm sure that's what's holding you back because everything else...

Steve: Yes.

Leo: It's a routine in BIOS that doesn't exist in EFI. I guess. I guess it doesn't exist. Well, it must exist, it's just not interrupt-driven. Right?

Steve: Yeah, it's just a different approach. You ask it for entry vectors, and it gives them to you, and so you call them instead of having - it's sort of a different way of doing it. And it makes it...

Leo: The capabilities of INT 13 are still there. INT 13 is the interrupt that accesses the drive.

Steve: Well, and actually SpinRite is still using some BIOS things just because they've always been there.

Leo: They're there.

Steve: I'm already making very little use of INT 13. And so what I'll be doing is I'll be

eliminating my use of INT 13 completely, and that will then bring me up to - actually, more portability will be a side effect of that.

Leo: Sure, exactly. And the whole world's x86 now. It's funny because Intel was moving away from x86. Intel was going to abandon it with the Prescott and all of that. I can't remember what they called the new stuff. But they were going to get rid of it. And then they realized that was a terrible idea.

Steve: Was there Itanium? Was that that...

Leo: Itanium, yeah. And I can't remember what it was called, the replacement. But IA64, that's what it was called.

Steve: Yeah, I've been coding a lot in the last few, well, I've been coding...

Leo: Have you.

Steve: ...all year, this longest repeated string thing that we'll be talking about here before long. And I just - it's such a pleasure coding in Intel assembly language. I just breathe it. So...

Leo: Yeah, you know it. It's your native tongue.

Steve: It is my native tongue.

Leo: Yeah. It's actually a crappy assembly language compared to something like 68000, which was...

Steve: Oh, god, do I wish they had not won. Oh. Oh.

Leo: I know. With the segmented - you still have to deal with that, right, the segmented memory?

Steve: No, that's gone.

Leo: Oh, thank god. You have a flat...

Steve: You're dealing with a legacy of few registers. And it's an evolved architecture, and evolution never generates as elegant a solution as starting from scratch and designing something beautiful. The Motorola 68000, National Semiconductor had a 32000 instruction set that is, oh, it's just...

Leo: Gorgeous.

Steve: ...sublime. And Intel...

Leo: That's what I - I coded in assembler in 68000 because I was writing for the early Macs. And I looked at x86. Actually it was i386 or something. Or was 8086, actually, that's what it was. And I went, ahhh. I played with it because at that time you couldn't access all of memory. It was a segmented memory architecture because the registers were too small.

Steve: Yeah.

Leo: And so you had to load a page, and then load an address within that. It was crazy.

Steve: Yup.

Leo: And I look at people like you, and I just go, whoa.

Steve: Well, and the biggest annoyance is there's a notion in an instruction set of something called orthogonality. If you have an orthogonal instruction set, then the idea is that, for example, all the registers can be used with all the opcodes. That is, it doesn't care. But the Intel architecture is anything but orthogonal. You have A, B, C, and D registers, and then something called ESI or ES and DS, and then BP and SP, the stack pointer. But all of them have different characteristics, like these can be combined in this way, and those can't. CX is an auto increment. Only BX, ESI, and EDI could be used for this. And so it was like, oh, it's just - and that really creates inefficient compiler code because the compiler has to know all of that. It's much easier to write a compiler for an orthogonal instruction set because it's able to freely juggle registers around and not have to be as, like, understand all of the minutiae of the instruction set.

Obviously this problem has been solved because lots of compilers exist for the x86. But boy, it's just - it's really sad that Intel won this war. But I'm sure glad only one person did because I only have to have one assembly language.

Leo: Yeah. That's true, too. And you know what? The new Intel stuff's great. No complaints.

Steve: Boy, and we have a lot of power.

Leo: So much power. More than we need now. I mean, it's just done. That whole thing is done. Tyler Larson in Scottsdale, Arizona offers an opinion on buffer bloat, why fixing your router usually won't help: You mentioned how a DD-WRT variant

attempts to solve the buffer bloat problem by limiting queue lengths and protocol adaptations such as RED. Actually, it won't help. The reason is your home router isn't the choke point in your network. The cable modem or DSL modem is. Your router has 100Mb or 1Gb on both ends, both in and out. Your cable modem takes in 100Mb or 1Gb from your LAN, but can only put out 1Mb, 10Mb, 50Mb, et cetera, to the WAN, depending on how fast your Internet service provider is. So the buffers that get filled aren't in the router, they're in the modem. The router's buffers are consistently empty. Same story for your computer. Changing your OS buffering behavior won't help, either.

The only thing you can really do to affect your buffering behavior is artificially limiting your bandwidth at your router. If your modem has 1Mb upstream bandwidth, you need to limit the bandwidth from your router to something like 800Kb, below the megabit, to keep from overloading the modem. This obviously decreases your overall speed, but can improve your latency under load. It's not perfect, but there you go.

And I'm going to give you a side question that we were debating in the chatroom last week. Does more bandwidth overcome buffer bloat? In other words, if you have tons of bandwidth, do you have to worry about buffer bloat? So these are kind of related, I think.

Steve: Okay, yes.

Leo: He says the cable modem or the DSL modem is where the buffer bloat is happening. That's not what others have told me, but is he right?

Steve: Well, and that's a question because he's right if the modems have buffers, that is, big buffers. It's not clear to me, because a modem is not a router...

Leo: Yeah, some modems come with a router. So maybe that's what he means.

Steve: Some are hybridized, right. But I don't know that the modem itself has a problem. We ought to go right into Question 10.

Leo: We'll do this all together.

Steve: Yeah, tie 9 and 10 together because Steve came up with a solution.

Leo: Steve "Snuffy" Sims in Hedley, Texas solved his buffer bloat problem: Steve, a very belated thank you for the many years of assistance you have given me from the early days of Windows 98 NETBIOS problems to now. I have followed you and Leo since the days of Screen Savers. I thought I might give a possible Band-Aid for the buffer bloat problem:

I ran the ICSI Netalyzr - which is such a great tool, such a great tool - with a download time of 200ms, but with an upload buffer of 2200ms. In other words, giant buffer, 2.2 seconds. Then I ran my own test, per your suggestion, of a large FTP upload on one computer while running Ping Plotter on another computer on the same network. Ping time to a totally different server went from 57ms to over 500ms due to the saturation. I am running a D-Link 825 router and went into the QoS settings - Quality of Service settings. My actual upload speed is 650Kb. I have a place there to limit upload speed and set it back to 600Kb, which apparently was enough to prevent the buffer from filling up. With this setting, the Ping Plotter ping time only increased from 57 to 80ms while the large upload was in progress on another computer. This didn't help Netalyzr results, but made a big difference in actual real world use. Once again, a very hearty thank you to you and Leo. Snuffy. So he's saying by constraining the upload bandwidth, he kept from saturating his network. But we've known about that for a long time.

Steve: Well, yes. In fact, that is, well, that is the effect of the buffer bloat. If he sees that pushing too much bandwidth delays the data in getting out of his system, that is, there is a buffer somewhere, and we don't know if it's in his router or in his modem or where, but somewhere he's able to induce 2.2 seconds of delay, that is, the buffer's that big, that if he just pumps as much data out as he can, then he'll fill that up. And that means that other traffic that's not part of this huge upload, for example, that he's doing, it's waiting its turn in the buffer also.

Leo: But that was always a problem because, if you saturate your upstream, and somebody else is surfing, the ACKs and the SYNs aren't going to come out, and so your upstream - what he's done, I don't know if it has to do with buffer bloat. He's just saying I won't let one computer saturate upstream, and that way my other computer will continue to have some access. But that has not - that's not buffer bloat, that's just how networks work. If you saturate your upstream with one computer, of course nothing else is going to work very well.

Steve: Except that all the computers on the network have equal access to the bandwidth. That is, they're all able to put packets onto the network that'll go out through the router at the same time. So...

Leo: I see. So they have - so he can't saturate it. He can only take his turn.

Steve: Correct. Exactly. There is no - there's really no notion of saturation. We've come to think of it that way because of buffer bloat.

Leo: Ah. I get it.

Steve: Yeah. So what he's done is, by recognizing that he cannot push, in his case, in Steve's case, more than 650Kb out, he's deliberately limiting himself at his router to 600Kb. And now he notices that he's never getting that long delay. So what that means is that he can do big uploads, and everybody else in the family can stay interactive because he's not letting that buffer get too deep, which would be delaying everybody

else who's also trying to use the system. So that is exactly what you want to do.

So that really does, that ties in with our prior question, Tyler, who was saying he believes the buffer is in the modem. Well, we don't really care where it is. But the recognized solution, even from the original videos that were demonstrating buffer bloat a couple months ago, was throttle your upstream yourself, and that keeps everybody else interactive. It prevents one person, see, it's not that one person's hogging the bandwidth, it's one person is forcing the buffer to be filled, which then hurts everybody. So one guy can really slow down the whole family.

Leo: So he fixed it by saying - by limiting his upstream to something below his capability.

Steve: Correct. And it doesn't have to be much below. Just enough below...

Leo: 50Kb, yeah.

Steve: Yeah, because if you're just a little bit below, then you'll have zero buffering. Really, think about it, if it can leave just a little faster than it's coming in, it'll never fill. If it's a little bit above, then it'll slowly build up because it won't be able to get out as fast as it's coming in. And you really lose nothing because you're either going to have a - the buffer's going to fill up, and it's going to be really bad for everybody, or if you limit yourself to just a little bit less than your upstream bandwidth, you really don't lose any speed. You prevent this buffering. But what we really wish is that there weren't huge buffers, that the buffers were only 15 packets deep, and they were just getting thrown away. Then all of our TCP systems would throttle themselves, and again everything would work. And this is the point is that buffers are not a good thing to have in a packet-routed network. They're just not good.

Leo: Right. So what about my contention that having more bandwidth doesn't necessarily fix buffer bloat?

Steve: Well, having more bandwidth means that it's harder to fill the buffer.

Leo: Okay, so it does.

Steve: Yes. So you're right, Leo. The idea is that, if more bandwidth means that the buffer is being emptied out the other end more quickly, so you have to really - you have to work harder. You have to run ahead of the bandwidth in order to fill the buffer. So if you've got more bandwidth, it's harder to run ahead of it.

Leo: Right. Okay. So it does help.

Steve: Yes.

Leo: I was wrong. We're going to go to Question 11, Steve Coakley in Phoenix with more Netalyzr test results: After looking into the long times given for DNS resolver lookup latency last time, it seems to be due to running the test in the evening when I had long ping times. I have 20Mb DSL service that's fast in the daytime but starting to slow down a lot in the evening when everyone is watching Netflix. This seems to be true for everybody. Normally I get about 42ms ping times to 4.2.2.2, but during the evening pings go up to 200ms.

Anyway, I found this list of fast DNS servers at TheOS.in and tried testing them all. Now, of course you have a product that does this, as well, Steve. They all gave me fast ping times and pretty low lookup latency of around 110 to 150ms, except for Google. It returned 229ms. However, they all had lots of DNS problems like returning names that don't exist, except for Google and Verizon which didn't have any problems. So it looks like GTEI/Verizon is still the best to use, 4.2.2.1, 4.2.2.2, et cetera.

Steve: This was a really good observation that I wanted to bring up. I don't think I've ever mentioned it before. First of all, these wacky IPs that you and I are familiar with, for anyone who isn't, those are - there's a set of six DNS servers, 4.2.2.1 through 4.2.2.6, which I've been with Level 3 for - actually I was with Verio in the old days, and those used to be Verio's servers.

Leo: Right, right, they were Verio, that's right.

Steve: Yeah. And they've always - I don't know where they are or what the story is with them. But, boy, I mean, they're, within the sort of the intelligentsia of Internetness, those are what people use. Now, I of course wrote a DNS Benchmark that I finished late last year.

Leo: Which is excellent. Highly recommend.

Steve: Which works great. And I do have, when I looked at that link that Steve provided, that TheOS.in, I remembered going there and making sure that I had all of those DNS servers in my master list also. So the Benchmark knows about those. The point was that time of day really does matter. We saw that vividly during all of the beta testing and pre-release testing of the DNS Benchmark. And I say in some of the web pages, don't just try this once. Try it and make a note of what the results are, but try it deliberately at different times of day because a DNS server that may be lickety-split when you try it is just really dragging. So the point is that DNS servers themselves come under varying degrees of load, and they are very load-dependent. So you really want to find the best one, you do need to try it in the morning, in the afternoon, in the evening, late at night, and make sure that your choices - and there's no way that my Benchmark can take the responsibility for that. It's got to be the user who does that.

Leo: You've got to do it by hand, yeah, yeah.

Steve: Yeah. And so it's just like, okay, try it at different times to make sure you don't,

by mistake, choose one that's really fast the one time you tried it, but is painfully slow when you actually want to rely on it.

Leo: Finally, our last question of the day, Mr. G., comes from...

Steve: Right on time.

Leo: ...James Lorenzen in Joplin, Missouri, and he wonders about the difference between 128-bit SSL and 2048-bit server keys. Quick question: You've been talking a lot lately about the number of bits we should use when creating server keys and certificates, or the fact that 768 bits has been compromised, and that 1024 bits should be good enough, but the recommended standard currently is 2048 bits. My question is, when visiting sites that use SSL, they say, hey, safe and secure, we're using 128-bit SSL. What's the difference?

Steve: So the short answer to this, I realized I have seen other questions...

Leo: We've talked about this before, but...

Steve: Yeah. But I think maybe I get too complicated or detailed or wander off the track. But so here's the short answer: SSL uses both. There's two types of encryption or crypto in an SSL connection. There is the server keys, where we end up with these big numbers, the 1024 and 2048, and someday we'll be talking about 4096. And then there's the connection keys, which is where the 128-bit SSL comes in. So it would be more proper for websites to advertise both, that is, the size of their asymmetric public keys - which we'd hope would be 2048, but if they're 1024 that's okay, too - and also their 128-bit SSL, but everybody has that. When you have SSL, you're going to have that level of security.

So again, the short answer is there are, because there's two different types of cryptography, there's public key and private key, or also known as asymmetric key, where you have different keys for encrypting and decrypting, and symmetric key, where you have the same, the asymmetric or public keys are much bigger because they need to be much bigger to provide an equal level of security due to the way their technology works, which is different than symmetric keys, which can be much shorter to provide an equivalent amount of strength. So that's why. It's like the websites aren't really telling you the whole story. They're sort of using what everybody else says. All connections use both types, one long one and one short one, and that gives us the security that we need.

Leo: It's just that simple. So in other words, don't worry. It's okay. It's meant to be that way.

Steve: It's good. It's all good.

Leo: Steve Gibson is the man in charge at GRC.com. That means you can go to

GRC.com for lots of things, including of course the world's best hard drive maintenance and recovery utility, SpinRite. Now, Steve, if you make a big change and make it Mac compatible, how much is that upgrade going to cost? You going to charge people for upgrading? Do you know?

Steve: Don't know. I've got a bunch of things that I've got to get done. And then I want to then, as I have mentioned, I mean, it's not even on the horizon yet, but I know what I want to do for what I'm calling 6.1, and it is to free it from the BIOS.

Leo: Free it. Free at last.

Steve: Free it from the BIOS. And I want to update it for a number of things that have happened. Western Digital has 4K sector drives. They're these hybrid drives that use both some EEPROM and some hard drive storage. There's secure erase capability. I want to give it stronger Serial ATA support. It works now, but sometimes users have to go in and reconfigure things in the BIOS for SpinRite to see it. I just want to make things easier and better. That's my short-term goal. And then the longer term goal is to take a look at Mac and the EFI and all of that. But so it's not happening...

Leo: It's not going to happen tomorrow.

Steve: It's not going to happen tomorrow. And but we will absolutely, as I always have, protect people who purchase, and then there'll never be any regrets. We still allow people to upgrade SpinRite 1.0 from 20-plus years ago and get a discount on SpinRite 6. So 6.1 will definitely be free, so there's no reason not to get it now. When we do 6.1, that'll just be transparent. Everyone will be able to upgrade, for sure, no charge.

Leo: So go to GRC.com, get SpinRite, get it now, with confidence. You should also, while you're there, there's lots of free stuff. And browse around. You'll be amazed. You won't believe the variety there, including this show. He's got show notes there. He's got transcriptions, text, as well as 16Kb versions. On our site, TWiT.tv, we've got the high-quality audio, the video and that. And of course you can watch the show live, it's always fun. We usually have some pre-show conversation about coffee, food, vitamins, the stuff that we try to keep out of the podcast. So if you like that stuff, tune in a little early, 11:00 a.m. Pacific, 2:00 p.m. Eastern at TWiT.tv, that's 1800 UTC, every Wednesday unless we move them around. But you can always find the calendar - by the way, I don't know if we say this enough - on TWiT.tv. That's where you'll find out where all our shows are. I posted on Google+ the schedule for today. We have, like, I mean, we do a ton of shows.

Steve: We may not have you next week; right?

Leo: Next week, I'm not sure. Eileen just - I don't know. I'm in Vegas. We do our NAB coverage. Here comes Eileen, running, running, running. We're going to do this show on Friday, Steve. So expect an email. We're moving it to Friday at 9:00 a.m.

Steve: Ooh, I love it.

Leo: Aren't I lucky. I picked it, apparently. But we didn't tell Steve. So now he knows. So next week it'll be Friday, 9:00 a.m. Pacific, 12:00 noon Eastern time, and we'll do the show then.

Steve: Perfect.

Leo: Thanks, Steve.

Steve: I'll be ready.

Leo: Consider that your notice. He's also on Twitter, @SGgrc, and of course - where else? Oh, if you have questions for Steve - I knew there was something else - and you want to be on the next feedback episode, which is in two weeks, GRC.com/feedback. There you go. There you go. Somebody's saying, "Which Friday? What are you talking about?" That would be April 20th. So instead of April 18th, we'll be on April 20th because our coverage, wall-to-wall coverage of the National Association of Broadcasters show starts Monday, goes through Thursday of next week. We're all going down to Vegas.

Steve: Perfect.

Leo: Thank you, Steve.

Steve: Thanks, Leo.

Leo: Thanks for being here, and we'll see you all next time on Security Now!.