
# Listener Feedback #140

**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-346.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-346-lq.mp3

---

SHOW TEASE: It's time for Security Now!. Steve Gibson's here with 10 great questions and answers, more on buffer bloat, security news, and a whole lot more. Yes, we'll even talk a little bit about coffee and the iPad. But just a little bit. Security Now! is next.

**Leo Laporte:** This is Security Now! with Steve Gibson, Episode 346, recorded Wednesday, March 28, 2012: Your questions, Steve's answers, #140.

It's time for Security Now! - time to protect yourself, your friends, your loved ones online; protect your privacy, too. And here he is, the Explainer in Chief, the man who makes it all happen, Mr. Steve "I've Got a Giant Mic and I'm Not Afraid to Use It" Gibson of GRC.com. Good day to you. How are you, Steve?

**Steve Gibson:** Great. Great to see you and talk to you and be connected to you. And we've got all kinds of stuff to talk about.

**Leo:** We sure do. It's Q&A episode.

**Steve:** And some of it, some is even about the Internet and security and computers. I'm sure we'll get to that.

**Leo:** Well, we have one Carbonite ad. We'll do that right after your SpinRite mention. But let's dig right - and I know we have questions and answers, but let's dig right into the security news off the top here.

**Steve:** Okay, now, first of all we should let our listeners know that, as a consequence of our buffer bloat episode, where I ran the analyzer last week...

**Leo:** Love that Netalyzr.

**Steve:** Oh, yeah, that's been a win for so many people.

**Leo:** I got an email from them saying thank you.

**Steve:** No kidding?

**Leo:** Yeah.

**Steve:** Oh, neat.

**Leo:** They said we got more traffic from you guys than we've ever had before, and it's really helped us with our research, so thank you.

**Steve:** Oh, fantastic. Well, as we know, we learned that my own connection had arguably a little more buffering than I wanted. But more significant for the podcast is, because I have a pair of T1s, and packets can be routed down either one, they were coming out at the other end with much more out of sequence than is normal for a connection. We've talked about this often. There's nothing that guarantees packets will arrive in the same sequence they're sent. TCP handles that by putting serial numbers in all of the TCP packets, and both ends do some buffering of their own. This is not the evil router buffering that is subject to bloat. This is after they arrive at the endpoint or until they are acknowledged having been sent, each endpoint keeps them, and that allows them to be reassembled in order.

Well, the problem is, for something that really needs to be real-time, like VoIP, you don't have time to hold things because that would introduce too much roundtrip delay. It would be like you and I were on a satellite connection instead of a real-time connection. So the decision is made, if packets come in out of sequence, if one appears to be missing, if we get one that comes along, we'll just send the audio in that packet and not worry about what's been missed. Now, there are algorithms for trying to fill in gaps, to sort of continue the audio from the last one you got and sort of hopefully bridge into the next one. And that's somewhat useful.

Anyway, the point is that, for this podcast, I have shut down one of the two T1s. So we are now running over one T1, and we'll just sort of see as the podcast goes along, and maybe assess it at the end, whether we seem to have had an improvement. And if so, I did get a tweet from someone who said that I could use Cisco access lists to route UDP over only one.

**Leo:** Oh, of course, through only one.

**Steve:** Now, I knew that there was a mechanism where I could use more of a hashing approach so that the hash of the source and destination IPs and maybe even ports was used so that you would always choose a consistent T1. The problem is that that would effectively cut my bandwidth down in half when, for example, I'm web surfing and things, because I really don't want TCP only to use one. But I would love, I have no problem if UDP protocol only uses one because DNS uses UDP. That's probably the only other thing I'm using UDP for other than VoIP with you and Skype, so that would be fine with me. So I have to see if I can actually do that with a Cisco access because I didn't know that I was able to specify not to share them. So we'll see how that works. But so that's the connection that we're talking over.

**Leo:** I guess the bottom line that may be counterintuitive is it just doesn't mean automatically more - and this is the buffer bloat problem, too. More bandwidth does not automatically mean better results on a UDP product like Skype. And simply having a strong consistent connection without dropped packets is more important than having the combined 3Mb that the two T1s provide you.

**Steve:** Yeah, you want low jitter. Okay, and jitter is a variation on buffer bloat. If packets are queuing with other packets in a buffer, then the receiving end perceives that as jitter. It doesn't see the packets coming in at a uniform rate. It doesn't know why. But it knows that somewhere along the way the packets got slowed down so that they were no longer coming in uniformly. And what VoIP wants is just as uniform packet timing as possible. So you want low jitter. And of course you certainly want low loss. And as we may be learning, you also really want them to come in in proper sequence. But I turned on - I went to TWiT Live this morning and happened to catch a replay of yesterday's MacBreak Weekly, where I heard you going off on Google is now evil. And so I thought, okay, wait a minute.

**Leo:** You have to understand, sometimes I get upset about things. And I overstate them.

**Steve:** We love you. We understand that. But I just thought I would let our Security Now! listeners know what it is that Google did to trigger this.

**Leo:** Well, it stemmed from - and I didn't quote these articles in our conversation on MacBreak Weekly, but I'll quote them now. It started with an article by Mat Honan, who is a really great writer on Gizmodo, which is a less-than-great gadget blog, but Mat is good, so I'm going to give Mat credit: "Google's Broken Promise: The End of 'Don't Be Evil.'" And in it he talks about where this "Don't Be Evil" mission statement came from. It's an interesting story, actually. Every company has a mission statement, and usually namby-pamby crap like...

**Steve:** We will serve our customers to the best of our ability.

**Leo:** We'll do better at - yeah, exactly. And Paul Buchheit - who was an early, I think Employee 23 or 24 at Google, later went on to write Gmail, and is now at Facebook, I think, but a brilliant guy, and he's a venture capitalist - was at one of these

corporate meetings at Google early on, and they said we need a mission statement. And he said, I want to do something that's not namby-pamby corporate. How about just "Don't Be Evil"? And they adopted it, and it was kind of famous that Google's kind of plan was not to be evil. Now, when that...

**Steve:** I loved it in sort of a hacker T-shirt sort of way. I mean, it's just like, it's a great ethic.

**Leo:** But it doesn't mean much. In fact, first of all, it's something not to do, not what do they do, but what they don't do. But it makes sense. And remember, you have to remember the context. This was in a time when all the other search engines were selling paid results into the search results. They were giving you search results that were tainted by advertising. And so in that context I think it's very likely that that's what he meant was let's give people clean, good search results and not be so focused on the bottom line that we do things that are bad for the customers, but good for us. Now, that's my own gloss on what he said.

So I think really for me where Google really has started to go wrong is not the privacy and tracking issue, although you might think that, and certainly that hasn't helped their reputation. Doing things like putting an invisible form in iOS browsers in order to end-around the third-party cookie prohibition is clearly evil. I mean, there's no question about that. But I will continue to defend - I think the word "tracking" when it comes to tracking cookies is such an anthropomorphic term that it really scares people more than it ought to. So I will defend the idea of targeted advertising and tracking cookies to that extent. But that's not - so I don't think that's really what I'm talking about.

I think where Google started to go wrong was with Search Plus My World, where they truthfully did start to modify their search results to benefit Google's bottom line. Now, they have lots of excuses for it. But really what happens is Google Plus is highlighted in search results, as is Yahoo!, also a Google property. And now if you go to Gmail or anywhere on Google, and you'll see your Google Bar, your black bar at the top of all Google pages now has a big ad for Google Play, which is essentially - here, I'll show you. So they put this bar on all Google properties. That's not necessarily evil. But they're starting to use it for promotional value.

Now, Google Play is of complete no use to anybody who isn't using an Android phone or Google's music. This is a service, a paid service that Google provides. And the fact that they're highlighting this so high just bugs the heck out of me. You know, the Google page used to be famous. You'd go to Google.com and...

**Steve:** So clean.

**Leo:** ...and it would just be Google.com, search box, and that's it. And they've more and more I think subverted what used to be a very good business model, by the way, to promote their own vehicles. And that to me - I don't think Google's more evil than Amazon, Apple, or any other company, or Facebook. They're all doing this. But I think this does, this runs counter to their original mission statement, "Don't Be Evil." And it comes from this Mat - I recommend people read the Mat Honan article

on Gizmodo because that's a much more eloquent and, I think, effective statement of what I'm talking about.

**Steve:** Well, there was some good discussion that you and Andy and the group had on this week's MacBreak Weekly. So I just wanted to understand that. And also, if our listeners are looking for another podcast that they're not already listening to, I can vouch for the fact that that was a fun dialogue that you guys had.

**Leo:** Yeah, thank you. I appreciate it. Yeah, I think that's a good show. More interesting to Mac and iOS owners, although nowadays I think there's nobody doesn't own something. Speaking of which...

**Steve:** Well, but you were also talking about Amazon and Google as...

**Leo:** Right, I mean, they all do this; right? I mean, this is not...

**Steve:** Or Apple, rather, Amazon and Apple.

**Leo:** Yeah, yeah. So, by the way, speaking of Apple, my Liquipeled phone came. And I'll be doing a review on this later, probably on Before You Buy, our product review show. But I had my iPhone dipped in an invisible clear shield. This is in lieu of a case. Wow, you can't even tell. You can't even tell.

**Steve:** And do they do it for you, or...

**Leo:** Oh, yeah, they have to.

**Steve:** ...do you attach a string to it and dip it in?

**Leo:** No, no, no, no. It says they don't recommend the device come in contact with any liquids. However...

**Steve:** Should it happen...

**Leo:** Should it happen, it applies a protective water repellant coating to your personal electronics. Boy, you couldn't even tell that it's been done. All right. So I'll do a review. I guess I'll have to dunk this in a bucket of water.

**Steve:** Well, I guess it was - was it at CES we saw them with phones underwater, and they were working.

**Leo:** And it was because you and I talked about it, and others, that they've contacted us, said, well, give us something, we'll do it. Okay.

**Steve:** And you hand over your phone.

**Leo:** I gave them my phone. I didn't even wipe the data. That's how trusting I am.

**Steve:** Well, in security news...

**Leo:** Yes.

**Steve:** We will cover coffee at the end of our catch-up.

**Leo:** By the way, I don't know if this is in your - I haven't gone through your rundown yet, but did you see there's a YouTube video about how law enforcement can completely get around that four-digit passcode on your iPhone?

**Steve:** Yeah, I did. And I don't have it in the notes here, although - we don't have enough information about what it is that they're doing. And it's a commercial product, where they're selling it to law enforcement. Just for the people who haven't seen it, they show running a piece of software, connecting the MAC that's running the software to an iPhone. You power off the iPhone, then you power it on while holding down the Home button at the same time as it boots. And...

**Leo:** Well, now, that's a flaw that's been fixed, I think, in many phones, in many iPhones.

**Steve:** And so, yeah, so that's one of my questions.

**Leo:** I thought that was fixed.

**Steve:** Is this still the case? Has it been fixed? They talk about on their site how they have all these engineers who are continually needing to find new holes in firmware of phones that are being updated. So they're using some sort of a hack in this video to get around that. And the question this raises is, if you had the "10 strikes and you're wiped" option turned on...

**Leo:** Which I do, of course...

**Steve:** Absolutely. I do, too, because we keep our devices backed up through iTunes on our machines, or even iCloud now. So who cares if it gets wiped out? I definitely want it

wiped out if some guy's trying to hack into it. So, and I don't use four digits, of course. I have the full alpha keyboard turned on, so...

**Leo:** Well, I use a pattern. Is that better or worse?

**Steve:** Eh, just don't tell anybody that's what you do. Oops, never mind.

**Leo:** Dvorak - well, no, it shows - no, on the iPhone I use a number. I use a pattern on Android.

**Steve:** I know. You're able to look at the screen.

**Leo:** Yeah, Dvorak's always looking at the smudges and saying, I think I can figure this out.

**Steve:** Oh, yeah, I think I know what your number is. I've got your number.

**Leo:** He's tricksy.

**Steve:** Yeah. So anyway, it was an interesting video. It's not clear that that means anything. As you said, we remember that hold the Home button down problem, and that's apparently what they're doing. Maybe they have a different means around that. Or maybe this is just old. I mean, for example, one of the stories for this week is from a year ago, essentially, and that's this - it just sort of resurfaced because the paper that is going to be published about it is soon to be published. But this was work done on problems with single sign-on systems which Google and Facebook and others are using, where some researchers were able to, by reverse engineering the protocol that they could see passing through their browser, they were able to subvert the single sign-on system.

And we've talked, of course, we did a whole podcast on exactly how OpenID works. And we'll remember that the way it operates, when you go to a site that gives you the option of signing onto that site using a different site that knows you - for example, sign on using Twitter, sign on using Facebook. When you click the button, you are taken to a page on the site you're still visiting, which then gives your browser a redirection with a bunch of special headers to the site that is going to be the identity provider. It sees this special request coming in, and it has all the required fancy crypto and authentication bells and whistles. And again, we covered this in detail on a full podcast to that effect. The identity provider then provides a response which your browser redirects back to the so-called "relying party," which is the one that is offering you the option of logging in through this identity provider. It then receives that packet, which it verifies and authenticates and again does all the crypto things and says, okay, fine, I'll accept that credential from that identity provider site as yours. And that's the way it works.

Well, it turns out, naturally, that we're in first-generation of these things. And we're often using downloadable kits where it's just, oh, grab this Java-based solution and plug it in, and you'll be up and running with multifactor, single sign-on technology. Well, there are

bugs in these things. And there are problems with implementations. And in fact there are problems with implementers in some cases. For example, what the researchers found was that it was possible to alter the data going by which the recipient of that data assumed was correct and had been signed but never checked the signature. So they broke the crypto signature, but the recipient of that just said, oh, well, we'll take the data. We're not going to verify it. So...

**Leo:** So that's an implementation error.

**Steve:** Precisely. The problem is that, for example, Computerworld's headline was "Study finds major flaws in single sign-on systems." And Ars Technica even said "Flawed sign-in services from Google and Facebook imperil user accounts." So, quote, "The researchers also found weaknesses in OpenID, a popular open standard that the researchers said Google, PayPal and 9 million other sites use" - nine million other sites, that's great - "to grant access to more than 1 billion accounts. The OpenID foundation has since addressed those bugs, as well."

So what I want our listeners to understand who saw this, and I got a bunch of tweets from people who wanted to make sure that I was aware of it, so thank you for those, is the protocols are solid. Without exception, we understand how to do this well enough that that part we got right, exactly as you summed it up, Leo. It was just some implementation mistakes. It was first-generation kits and assumptions being made by the users of the kits. Like, for example, they didn't have to make a call to verify the signature. The kit would do it for them. So in some cases it was just sort of a communications error in not understanding whose job it was to perform the verification stuff.

And, okay, and this was all last year, around this time, like April/May of 2011. And all of this has been fixed a year ago. It was fixed immediately because the researchers told everybody whose systems they had been able to hack what they had done, and they were fixed immediately. So it's like, oh, okay, thanks very much. So, I mean, and as far as everyone knows, no one has been exploited by this. So this was - this made the news because the paper will be presented in a couple of months, and but it's already been fixed. And, more importantly, there's nothing wrong with the concept. Certainly we have to implement it correctly. But that's always the lesson that we're encountering with security stuff is that, yeah, not only can the theory be right, but the implementation has to be there, too.

**Leo:** We're going to give you a new title, the Debunker in Chief. Explainer and Debunker.

**Steve:** There was another story that made a lot of news. And this was actually Ars Technica covered it initially on March 16th, and then it was updated with some additional information because what Ars reported really upset people. And this was that iPhones and iPads were leaking their past MAC addresses. A security researcher by the name of Mark Wuergler worked with Ars and an Ars reporter after some research he had done. He has a penetration services company and has written some apps that are sort of very much sort of Firesheep resembling, where you can go to an open access point, and he gathers all this information over unsecured WiFi and presents it in a nice fashion. So the best thing to do is just to read a little bit of this verbatim from Ars's page, just the top of the story, to give you a sense for what this is. It sets it up perfectly:

"As a security professional who gets paid to hack into high-value networks, Mark Wuergler often gets a boost when his targets use smartphones, especially when the device happens to be an iPhone that regularly connects to WiFi networks. That's because the iPhone" - and by the way, this is true, this is verified, so keep that in mind. "That's because the iPhone is the only smartphone he knows of that transmits to anyone within range the unique identifiers of the past three wireless access points..."

**Leo:** Past three?

**Steve:** The past three "the user has logged into."

**Leo:** Criminy.

**Steve:** "He can then use off-the-shelf hardware to passively retrieve the routers' MAC ... addresses and look them up in databases such as Google's Location Services..."

**Leo:** Wow, so use that to track you.

**Steve:** Uh-huh. Well, okay, get this. If you were at a Starbucks somewhere, and you were capturing the MAC addresses of the customers that walk in who have iPhones with their WiFi turned on, as most people would, and the phone is looking for an access point to access, and you were able to critically cross-reference those MAC addresses with their location, then you know they're here at Starbucks and not home, and you know where they live.

**Leo:** Oh, wow.

**Steve:** So, yeah.

**Leo:** You could say, hey, let's go. This'd be a good time to go visit his house.

**Steve:** Precisely.

**Leo:** He's going to be drinking coffee for a while. Let's go surprise him.

**Steve:** Yeah.

**Leo:** Wow.

**Steve:** Not good.

**Leo:** Not good.

**Steve:** So continuing, "By allowing him to pinpoint the precise location of the wireless network, iPhones give him a quick leg-up" - so to speak - "when performing reconnaissance on prospective marks. 'This is interesting on a security level because I'll know where you work, I'll know where you live, I'll know where you frequent,' Wuergler, who is a senior security researcher for Miami-based Immunity, Inc. told Ars." Continuing the quote, "'If the last access point you connected to was your home, for example, I'll know right where to go to get to you later or to get to your data.'" Or to visit while you're at Starbucks.

**Leo:** You're kind of cute. Maybe I should visit you at home later. Geez Louise.

**Steve:** "'If I'm an attacker that wants'" - I know, and this is true. "'If I'm an attacker that wants to break into your company, this becomes a disclosure that an attacker isn't going to pass up.'

"The exposure of MAC addresses extends not only to iPhones, but to all Apple devices with WiFi capabilities, he said. It means that whenever the wireless features are enabled and not connected to a network - for instance, during a brief encounter at a Starbucks - they broadcast the unique identifiers, and it's trivial for anyone nearby to record them. Wuergler speculates the behavior is a feature designed to automate configuration for networks users regularly access."

**Leo:** Wow.

**Steve:** Okay. So this generated so much kerfuffle that Ars asked Robert Graham to verify. Now, Robert Graham is a person we've spoken of often. He was the main networking techie behind BlackICE. And he's still involved. Errata Security is his firm. So Robert knows what he's talking about 100 percent. He fired up Wireshark, walked into Starbucks, turned on his iPad 3, and saw his home router MAC address sent off into the air and verified that this is in fact happening.

So our listeners probably know enough, having listened to us discuss how the Ethernet works in our networking fundamentals stuff, to even guess what's happening. This is an ARP query. This is the Apple products wanting to ease reconnection to a previously connected network. So unlike standard Ethernet devices - and actually, Leo, this may be also a function of the fact that they sort of stay on all the time. You know, they're not actually being shut down and rebooted, which may flush the ARP cache in the device.

But the fact that Android phones don't do it, BlackBerrys don't do it, so it may be something Apple is deliberately doing to make their users' experience more seamless because, for example, if you were to come into your house, what your phone is doing is it's querying the last several access points, WiFi points, that it has visited to see if it is again in the presence of one of those. Unfortunately, in order to do that, it's divulging what access point it was connected to because this is a - ARP is a non-encrypted, pre-connection, low-level - it stands for Address Resolution Protocol, the way devices talk over Ethernet. And as we know, WiFi is just a wireless version of the Ethernet. So it's got all the same protocols.

So I don't know how Apple will respond to this because people are not happy, as you can imagine, that their devices are, when in the presence of a WiFi network and connecting up, trying to connect to access points they have connected to before using their MAC address. And once upon a time that wouldn't have been a problem, except that Google has been roaming the streets of the world, literally, acquiring all the MAC addresses and noting where they are.

**Leo:** Well, not just Google. There's a company called Skyhook that also does this.

**Steve:** Yes, yes.

**Leo:** And predates Google doing it.

**Steve:** Now, Google did increase the difficulty of making ad hoc queries by requiring you to provide two MAC addresses that are physically near each other. You can't just say give me one. You have to have two. And that's a nice workaround. It allows the fundamental underlying technology still to work, but you would have to actually be near one of those in order to have a MAC address of another access point nearby. You ask them both of Google. Google verifies that they are physically near each other, and so it's not that you've just been able to capture one at Starbucks. So that raises the bar a little bit in a nice way. And I think it's a clever solution to the problem. But this is still, I mean, this is arguably a problem. I imagine Apple...

**Leo:** I would say inarguably a problem. I think, Houston, we have a problem, yes.

**Steve:** I think Apple will probably have to back off and change their networking stack so that it no longer does this. Users, until we get an update, can simply turn off WiFi and not have the benefit of seamless WiFi connectivity when they walk into Starbucks and other similar open access point locations. And that might be a good idea, if this sort of thing upsets you, because it is true that the world probably knows exactly where your router is geographically, and it's certainly possible to get that information one way or another. So probably not good.

**Leo:** Unh-unh. No, not good at all.

**Steve:** Microsoft in the news, this was just in the news today, is getting involved in the HTTP 2.0 effort. A couple weeks ago we talked about SPDY and covered the way that protocol works in detail. Microsoft - immediately when I saw this I sort of closed my eyes, I went, oh, no. Here we go.

**Leo:** Here we go again.

**Steve:** The standards wars. Microsoft is saying that they're okay with SPDY.

**Leo:** But...

**Steve:** That they think it's a nice effort. But...

**Leo:** They got something better.

**Steve:** They, well, they say they want things that are more oriented toward mobile devices.

**Leo:** Well, that's true.

**Steve:** But they haven't said what that means. And SPDY's developer at Google said, wait a minute, there's nothing un-mobile about SPDY.

**Leo:** Right, it's any web browser; right?

**Steve:** What it does sound like maybe Microsoft wants, which I can see, is something a little bit more like the web sockets API which is in HTML5. And that is to say, more application-level features. What Google has done is they've sort of made a transparent improvement so that the browser still browses and the server still serves. But behind the scenes it's doing that as our listeners know from that podcast in a much more efficient fashion in many ways. But there really aren't any application-level features that web applications could use in order to leverage that. And so Microsoft I think has a point, that the web sockets API does provide that sort of feature, although it's layered right now on top of HTTP, so it still has the underlying carrier technology, the transport layer of HTTP. Integrating all that more tightly makes sense.

So the good news is we're moving towards a next-generation HTTP one way or the other which looks like it's going to incorporate the best of all of these efforts. It'll probably be - it'll be slow and take a while. Someone, it was Simon Zerafa, made a comment about Twitter no longer using SPDY. I just saw his Tweet earlier today, and I haven't had a chance to verify it. I do have - I've got my little SPDY gizmo displayer in Chrome. Let me go to Twitter.com.

**Leo:** Why would they - they must have found something better, if they're not using SPDY. You know, what's interesting is that Twitter has a development toolkit which is now being widely adopted by people as a framework. And if it had SPDY built into it, and I'm not sure it doesn't, that would be a really cool way to spread SPDY to websites easily.

**Steve:** Well, I am at Twitter.com over an HTTPS secured connection. And my little indicator is off.

**Leo:** Oh, man.

**Steve:** And it was on earlier.

**Leo:** Oh, man. Well...

**Steve:** Well, they may have found a problem, which they will get around to fixing.

**Leo:** Certainly, if somebody would find a problem, it would be Twitter. So, yeah, maybe there is a solution. They have a framework called Bootstrap which a lot of people are using now to create sites.

**Steve:** Oh, cool.

**Leo:** Yeah, in fact Gina Trapani just did a site with Bootstrap. Let me just see if Bootstrap supports SPDY. It's S-P-D-Y; right?

**Steve:** Yes.

**Leo:** Responsive design tools, no, it doesn't look like it's built in. That's too bad because that would be great. It's CSS-focused. I guess it's not.

**Steve:** Yeah, okay, so it's not low enough level to...

**Leo:** Not low enough level, yeah.

**Steve:** So it would be compatible with it, but wouldn't require it. Brian Krebs, our intrepid security reporter who spends a lot of time digging around down in the dark recesses of the Internet, warned recently that the latest Java flaw is being actively exploited, and successfully. He reported that the Java flaw that was fixed middle of last month, which is v6 update 31, or v7 update 3, has been added now to the very popular exploit kits, which many of the latest exploits are being built on, and as a consequence has dramatically increased their penetration success rates, which says that many people, not our listeners I'm sure, do have Java installed and aren't paying attention to it and aren't keeping it current.

Now, I mention that, of course, because last week's tool for measuring network buffer bloat required Java. And I have to say that I was looking favorably at it from a standpoint of, wow, if they can do all of that, I could definitely have some fun doing low-level network things in a way that was platform-independent. Yes, it requires Java, but I'm afraid the future probably does.

**Leo:** Well, the good news is all browsers now, by default, don't turn on Java. I think; right? I mean, I know Chrome says, do you want me to allow Java?

**Steve:** Right.

**Leo:** Isn't that default now in most cases?

**Steve:** Well, you can definitely disable it.

**Leo:** Yeah. But that would be foolish.

**Steve:** I think you're right that you have to enable it. What Brian suggested, I liked. And that is, if you find yourself often needing Java, you could go the multibrowser approach. For example, it's easy to disable in Firefox under add-ons. You just say turn that off. And then, for example, you might just run Java under Chrome and use Chrome when you need Java, but don't use it when you're just out cruising around the Internet, or vice versa, or any combination of those.

So I did want to remind people, you just go to Java.com, and then there's a link, "Do I Have Java?" You can click that, and it will run a test to tell you if you have it and, if you do, if it's current. And I just did mine this morning to make sure, although I was sure that I had done it a month ago. And it said, yes, you've got v6, update 31. So it's like, ah, okay, good.

**Leo:** I always thought Java was sandboxed, and so it was less vulnerable, the applets were less - I know there are problems, but it seems to me that, if you combine the fact that by default it's opted out, it automatically updates, and it's more secure, say, than JavaScript, it seems like it's not a bad choice, let's put it that way.

**Steve:** Well, it's not a bad choice. And what you're thinking of is that it has always been an interpreted byte token style language. It was built, as we all remember, back in the day, what is it, Bill Joy, for set top boxes, and so meant to be platform independent, processor independent. You have a Java runtime which interprets the byte code. If the runtime were perfect, then you could argue nothing you could do in terms of giving code to the runtime could represent a problem. So it has the potential of being very secure.

The problem is there's just this huge tendency to sacrifice security for speed. Even Google, with the Chromium project, there is a move afoot to run native code in Chrome, not just JavaScript but actual Intel executable. And Google says, oh, we have a way to corral it and make it safe. And it's like, okay. I mean, they really want that ultimate performance. And it'd be great if we could have everything.

**Leo:** I'll have to look because when Java - for a long time Java had this sandbox model where, if it was untrusted code, unless you had a certificate, and you explicitly

trusted it, it couldn't access the file system, it couldn't access the networking, it couldn't access browser internals, it was very much protected. And maybe over time - as you say, people want convenience - that's eroded. But it was, for a long time, it was deemed mostly secure. I mean, there are malicious applets. But you have to trust them explicitly, I think.

**Steve:** So a little quick iPad follow-up. The news came out this week that it may be the LED lighting that has been the source of heat. And I can confirm from my own experience that, when I have turned the backlighting up all the way, that does really seem to increase the heat. Nothing that I do on my iPad 3 is processor intensive. So I'm not doing gaming with lots of animation or movement or anything. But it was when I had the light turned up. And in fact because of this next-generation, super-high-resolution screen, the lighting is 2.5 times brighter, that is, there is 2.5 times more backlighting, and that generates heat because it is pulling power from the battery. So that could certainly be part of it. Then there was just in the last day or two some controversy about the iPad charging, that it saying it was 100 percent charged, but it wasn't really, that some investigators discovered that...

**Leo:** Yeah. By the way? Bogus.

**Steve:** I know.

**Leo:** Apple's responded. Completely fooled the "investigators." Which is a shame because this is the guy from DisplayMate. I mean, I've always trusted him in the past. He just kind of got fooled.

**Steve:** Well, he...

**Leo:** He noted some behavior.

**Steve:** Yeah. He wasn't wrong...

**Leo:** Right.

**Steve:** ...in what it was doing. And frankly, I'm glad to know they're doing this because I have worried, for example, when I have a laptop that is charged up, but it's just sitting there on the adapter, I have wondered if the battery isn't over time just draining without any attention being given by the laptop. And so sometimes I'll, like, disconnect it deliberately and reconnect it so that the laptop checks it again, and hey, sure enough, it'll go into charging mode and bring the battery back up because it had drifted away from a full charge over time.

**Leo:** And as it turns out, that's exactly what Apple's doing mechanically.

**Steve:** Yes. Apple is taking active responsibility for it. And so it's bringing the battery down and then bringing it back up again, and bringing it down and bringing it back up, in order to keep it - to have knowledge that it's keeping it up at 100 percent. And unfortunately, with our current technology, there's so much battery-charge monitoring on the cells themselves that there is some self-discharge of lithium ions. So you do need to keep an eye on it, which is what Apple's doing. So for anyone who was wondering. And then I can now tell you, Leo, that I still prefer reading on my large Kindle.

**Leo:** Ah, the DX, the eInk version.

**Steve:** Yes. The eInk, even over the iPad 3. The iPad 3 is amazingly gorgeous. And I have one with no antiglare film and one with antiglare film. And I can't decide what I want because the antiglare that I have does knock down that super-high retina resolution. It sort of gives it sort of a twinkly sparkly effect, sort of a specular feeling, where when I compare it, and I've left the other pad with no antiglare so that I can do some A/B comparison, there's no question that that glossy screen is even more important when you've got this kind of pixel density, if you really want to see the quality and the clarity of the LCD. On the other hand, it is a problem when there's light behind you, or you're outside, or you're in a restaurant with some bright lights above, and you've got them shining into your pad. So I like them both sort of in a different way. But more than either of them, just for reading, I still think a reflective display is a lot easier on the eyes. I love my Kindle DX.

**Leo:** Yeah, I don't think it's a resolution issue. I think it's having backlit versus reflective. I think that's the...

**Steve:** Oh, yeah, unfortunately I was confusing about nine different issues there in one.

**Leo:** No, no. And I know what you're talking about. And that was really what people always said is, well, I don't care, the quality of the screen isn't the issue, it's that the screen itself is emissive.

**Steve:** Yes.

**Leo:** And that's not as easy on my eyes as reflective. And some of it's conditioning; right? Because we grew up reading reflective materials.

**Steve:** Yeah, I don't know. I'm able to be reconditioned. Except when it comes to cold brew coffee.

**Leo:** Oh. How do you - okay. So you bought it. You bought the Toddy.

**Steve:** Of course I have all this stuff, Leo. I'm going to find out what's going on. So I tweeted earlier this week, I said "Bite Me!" because I like bite.

**Leo:** Yeah. There's no bite.

**Steve:** I like my coffee - there is no bite.

**Leo:** It's real smooth. There's no acid

**Steve:** It is bizarre. Now, I also have the stomach of a billy goat. I mean, I'm sure...

**Leo:** Right. Because a lot of us, especially as we get older, my stomach gets upset. I just had a cup of acidic coffee, and I feel it.

**Steve:** Yeah. I love it.

**Leo:** So this comes from our discussion last week of FunraniumLabs.com, the Black Blood of the Earth, cold-brewed coffee. And, now, the good news is that Toddy, that brewer that you got and I have here, too, is only $30.

**Steve:** Right. It was an inexpensive experiment. I lost a pound of Starbucks Espresso.

**Leo:** You really don't like it.

**Steve:** No, I just - it's just - it doesn't taste like coffee. It just sort of freaks me out.

**Leo:** I should have warned you because the Funranium is just like that, that Black Blood of the Earth. It's just too smooth.

**Steve:** Yeah, it is ultra smooth and - I don't know. Now, so the good news is, if there are listeners who have a problem with acidity, this solves it, baby. I mean, it is smooth. But it just isn't coffee. I want to be bitten.

**Leo:** Well, you won't like this, then. Don't watch. Because I've purchased another device. Let me show you real quickly. This is - it's just pretty. That's why I bought it. It's a wooden rack...

**Steve:** I saw that. I thought, oh, no, Steve.

**Leo:** It's like a distiller.

**Steve:** Don't go there.

**Leo:** Yeah, yeah. It's a little pricey. And but I thought it'd be kind of cool in my house to have this on the counter. And [laughing], oh, lord.

**Steve:** It's gorgeous looking.

**Leo:** It's, well, that's kind - it's more like it's just cool than anything else. This is called - I'll give you the name so that those of you who want to find out more, it is on Amazon.

**Steve:** It looks like a coffee guillotine.

**Leo:** Yeah. It really looks like a device; right? And it's tall.

**Steve:** Yeah, looks like a chem lab sort of...

**Leo:** Yeah, it's called a Yama, Y-A-M-A, cold drip coffee and tea maker from Northwest Glass. And let me see how tall it is. It's two feet tall. So the idea...

**Steve:** Oh, it looks like it's seven feet tall.

**Leo:** I know. It does look like it goes a long way. It looks a little bit like a distillery. It's got a lot of glass. That's one of the reasons I got it because the Toddy - this is a little more expensive than the Toddy, it's 200 bucks - but the Toddy is plastic. They say it's not - there's no BPA in it, and so it's safe, et cetera, et cetera. But it is a plastic brewing device. And it took all day, right, 12 hours. So you put the coffee in here...

**Steve:** Oh, I gave mine 36.

**Leo:** You let it just go.

**Steve:** I let it go for 36. Mine said - I think I got a different version or a different brand.

**Leo:** You strong-brewed it.

**Steve:** Mine was Filtron, I think, was the...

**Leo:** Ah, okay.

**Steve:** Yeah. And so you basically ground - I used the nice burr grinder that we have and ground the entire pound of Starbucks, dumped it into this thing, and then it drizzled the amount of water in slowly in the top, and it filled up over the course of about an hour. And then I let it sit there for a day and a half, 36 hours...

**Leo:** They say longer the better, or the longer the richer.

**Steve:** Yeah. And then I drained it all. In fact, this said that some people who want to stretch their coffee dollar will then use half the amount of water a second time to, like, run it through the grinds or grounds...

**Leo:** We're not stretching our coffee dollar here.

**Steve:** I wasn't stretching. And in my case the result is this syrup, this super-concentrated coffee syrup.

**Leo:** It's got flavor, doesn't it? I mean, it's not that it doesn't have flavor. It just doesn't have that bite.

**Steve:** Yeah, you're right. It definitely has flavor. And I just thought, no.

**Leo:** I'm with you. It tastes - it feels watery. Because it doesn't have that acidity, it feels like there's nothing going on. It's just kind of tea.

**Steve:** Yeah, it's just - it's different. It's very flat to me. So...

**Leo:** Anyway, we'll just have to see.

**Steve:** Anyway, so now I know.

**Leo:** Hey, this Yama is, like, 200 bucks. So I'm glad you didn't go that far.

**Steve:** No.

**Leo:** But it will be - I will look like I am making meth in my house. Oh, lordy.

**Steve:** Well, I have a short note, a very note, from a SpinRite listener who is anonymous. He said, "I am a computer maintenance freak. I had been experiencing a problem which turned out to be a software glitch. However, I was at the time afraid my drives were going to go. I learned about SpinRite while reading up on SmartComputing." He has those capitalized, so maybe that's a site. And he said, "Double-checked with my

office computer guy, who highly recommended SpinRite. Purchased and downloaded it today. It took a few hours to run through my drives. Seems like things are running better and faster than ever. So thanks. This was a great investment, and I will add SpinRite to my maintenance schedule." So he's got the right idea. Get it before your drives die, and they probably never will.

**Leo:** Yeah, yeah. All right. Question #1, an anonymous listener dropped by to make a quick, however important, point, suggesting that doubling key size is security theater. Remember we had a question two weeks ago, somebody says, well, why don't we just - we went from 1024 to 2048. Why don't we just go to 5096 or something? If your key is large enough, he says, or she says, to make a brute force attack infeasible, a longer key doesn't add security. Beyond that point, a determined bad guy will try to exploit a weaker link, and there are plenty of those, like buggy software, spear phishing, social engineering to get a keylogger installed. Bad guys know the old story: If you're hiking in the woods, you don't need to outrun the bear as long as you can outrun the other hikers [laughing].

**Steve:** I thought this was a good point.

**Leo:** I love that analogy.

**Steve:** So it's the weakest link. And at some point we know - in fact, true crypto failure is almost unknown. I'm trying to, I mean, almost, I want to say, because there have been, historically, there have been some flaws found in older technology. We no longer use MD4. But even RC4 that was the crypto used in the very first WiFi, the WEP WiFi, it wasn't the fault of the crypto, it was the fault of, again, the implementation wrapper that the crypto contained. And so that we continue to see, just like we were talking about OpenID and other things. It was like, hey, all of this is all super security and signed and keys and all that, but then they forgot to check, see if the signature was valid.

**Leo:** Right.

**Steve:** Like, whoops.

**Leo:** Whoo, yeah, all right.

**Steve:** So but the one thing I did want to remind us of, sort of that is a counterpoint to this notion that doubling key size is security theater, is the notion of future-proofing because that's something to keep in mind. There's this spectre of quantum computers hovering out there that are sort of going to just instantaneously try all possible keys at once. We're a long way from there. And lord knows what happens when those exist because the end of the world as we know it.

But for now, it is significant, I think, that that NSA facility is not attempting to crack things today. They're going back and going to crack things that they've been recording for the last 50 years, back when the underlying security technology was strong enough for then, but not for now. So there is this notion of the future. At the same time, 128 bits

is plenty for connection-oriented things, like Carbonite, for example, is using 128 encryption. That's a session key used on a point-to-point link which is regenerated and changed every time you reconnect. And also sometimes on the fly you're able to renegotiate a key on a running basis on these connections.

So 256 bits is plenty for data at rest as opposed to data in motion. So you want to choose key lengths properly. But the anonymous listener who wrote this question is certainly correct that, once you are future-proof, then all you're doing is wasting space and time and processor cycles. And heating up your iPad 3 needlessly.

**Leo:** By the way, it is not that hot. It's just nicely warm.

**Steve:** Yeah, it is.

**Leo:** Think of it as a hand warmer, a sheet warmer.

**Steve:** I love it. I don't regret the iPad 3 at all.

**Leo:** Oh, man, I love it, yeah. They're just...

**Steve:** It's a nice device.

**Leo:** A lot of people saying, oh, it's just a mere incremental upgrade. I don't know how they could say that when they look at the screen. It is more than an incremental upgrade. And the camera.

**Steve:** Amazing.

**Leo:** Yeah, yeah. Al Kraybill in Arlington Heights, Illinois, found some buffer bloat. SN-345 equals fantastic. I ran the test and got "Network buffer measurements" - he's talking about the Netalyzr test. In fact, bit.ly/sn345 will take you to that Netalyzr test. Don't all do it at once. We broke it last time. But like I said, they wrote, and they said thank you, we appreciate it, we're getting a lot of data for our - because it's a study they're doing.

**Steve:** Neat.

**Leo:** And I gave it to Russell, our IT guy, said this is a great thing to have in your toolkit because we learned a lot. So this guy had an uplink of - I guess an uplink latency of 490ms, downlink of 2,000ms. Yikes. That's what we call a bloated buffer. So what can I do about it? Is there any way to tell where the bloated buffer is? My router, a D-Link DI-602, is about eight years old. Could that have too much buffer?

**Steve:** And this is the problem, that I saw some amazing measurements from our listeners. One guy was at 7.5, wait, not minutes, 7.5 seconds. So, I mean, 7,500ms in one direction. And the problem is that we're in that awkward place where something is getting a lot of attention, yet as the stickers say on the back of our televisions, there are no user-serviceable parts inside. Once upon a time you had tubes, and you'd take the back off the TV set, and remember you'd not want to use one of those cheater cables that allowed you to keep the thing fired up with the back off because you wouldn't want to electrocute yourself. And you'd pull the tubes out and take them down to the drug store and run them through the tube tester. I'm sure you remember those days, Leo.

And now we've just got boxes that are closed. And at the moment, while this issue is still so new, there just isn't anything for us to tune. There is, like, the newer version, the newest version of Linux, 3.3, is beginning to address this. Hopefully that will make it into some router firmware, like the Tomato or the DD-WRT stuff, where we'll begin to get this addressed. At the moment, right now, I don't think there's much that anyone can do. And, I mean, this is where I guess I'm glad that I'm as busy and backlogged as I am with existing projects because, I mean, I could just go off on this and never return.

**Leo:** Yeah, it's fascinating. It's fascinating.

**Steve:** Oh, I would love to do a utility that would tell you where in your link the problem was, and it's possible. But no, don't worry, I'm not going to let myself get distracted by that. So Al, I just, unfortunately, it's useful to know we have the problem. There's not much we can do except to work to minimize the buffering, which is to say, if you know you've got a - when something is saturating your bandwidth, when you know that in addition to saturation you also have delay, then the only thing you can do, if you're unable to find the delay and remove it, is work on whatever it takes not to allow that buffer to get full. Which, for example, means being careful not to be uploading a big file when other people in the household are trying to be interactive on their computers, do that some other time. And so at least now we understand what's going on, which is a big step forward, although it also creates some frustration in people who want to fix it.

**Leo:** Yeah, yeah. Well, I'm sure, I think we'll see fixes in time. I mean, we'll see something, I hope. Steve Coakley in Phoenix, Arizona found router buffer excessive buffering. Oh, really. Another one. I ran the Netalyzr utility you mentioned on Security Now! - again, bit.ly/sn345. And besides excessive buffering, I got a lot of strange errors about DNS not working correctly. It doesn't look like I can change the DNS server in my Qwest Actiontec Q1000 DSL modem/router. It's set to - and he gives some...

**Steve:** IPs.

**Leo:** ...IPs. He changed it to the 4.2.2.2, which is Verizon, right? Or is that Level 1? Anyway...

**Steve:** Actually, it used to be Level 3.

**Leo:** 3, I mean, yeah.

**Steve:** And I'm not sure whose name is on it. I think Level 3 still.

**Leo:** And ran the test again. This time all the odd DNS errors went away, and it only found two problems. The first one, network packet buffering may be excessive. We estimate your uplink as having 5,700ms of buffering. Yeah. 5,700ms of buffering. And we estimate your downlink as having 450ms of buffering. Wow. 5.7 seconds is a long time. Can anything be done about that? Can I even tell where it's happening? This is kind of like the previous question. The second problem, DNS resolver properties lookup latency was 340ms. That doesn't seem so bad.

**Steve:** Okay. So I did want to mention that that test tests a lot of other things. And many users found, just as Steve Coakley did, found other problems with their network that they were unaware of. My sense is that 340ms is a little slow for DNS lookups. I don't know why it would be so slow. Maybe it's just the DSL connection that he's got. I wanted to remind people that my own, GRC's DNS Benchmark, exists, and that that might be a good thing to use. There may be some solid, publicly available DNS servers other than the Level 3 servers, although those generally do perform up near the top of the list.

But the DNS Benchmark from GRC, you just - in fact, I think you can just put "DNS Benchmark" into Google, and I pretty much claim that territory now because the Benchmark is a good one. It is Windows-only, but it is friendly with Linux under WINE, and you can run it on MAC with WINE, as well. And again, we've discovered huge buffers. One of the other problems that we have is that it could be ISPs buffering in their routers, so those buffers are completely inaccessible. And also we know that later model network adapters have large ring buffers in the kernel, so that's introducing delay. And we may never have access to that.

So again, the best thing we can do is, like, ask everybody, make noise, jump up and down. Hopefully this is a problem which just sort of, well, we know that it just sort of crept up on us. Nobody was really paying attention to it. Now a lot of attention is coming to it because people are downloading large content, not just being interactive. This was never a problem when everything was just web surfing, clicking on links and being interactive. This because a problem when some member of the family wants to watch TV over their Internet connection, which was crazy five years ago.

**Leo:** Crazy. Crazy talk.

**Steve:** Now it's what people do while other people want to...

**Leo:** It is, it's what we do.

**Steve:** It is, it's just - it's amazing.

**Leo:** In fact, often three or four people in the house are watching different channels on their iPad, their TV, their Netflix, their Roku. I see that in our house all the time. And then somebody wants to Skype, it's just - what a world.

**Steve:** It's amazing. It's amazing it works.

**Leo:** Bandwidth consumption is not going to go down anytime soon. I think we've realized that. I think Comcast realized it three years ago. That's why they put caps on. Magic Johnson - or, no, just Magic John in Colorado, USA, needed to comment about our server security conversation: I listened with horror, Steve, as you agree with Leo's supposed expert. You did not agree, by the way. I want to make this clear. You sat there and nodded, but you didn't necessarily agree.

I maintain websites with in excess of 14 million unique visitors a day. We have never been compromised, yet we see hundreds of attempts per hour. It's not PHP that is the problem. It is the code that is written in PHP - well, okay, thank you, master of the obvious - and the willingness of a system administrator not to correctly set file and directory permissions. And that one I might agree with. Bad code can be written in any language, as can good code. The difference is bad coders are tempted toward the use of toy languages such as PHP. I don't think he understands what I was talking about.

There's no excuse for the injections that have happened and the placement of code on Leo's system. I was especially horrified by your acquiescence, which again, you did not do, to my comments that in the good old days we had a cgi-bin directory. We still have a cgi-bin if we want, he says.

**Steve:** Well, I wanted just to come back to this briefly because I had a nice exchange with Bear.

**Leo:** Bear, who is a pretty good expert.

**Steve:** And what happened was...

**Leo:** Runs systems for Mozilla, among others. Okay.

**Steve:** There was some, yeah, similar sentiment that I tripped over in the GRC newsgroups.

**Leo:** Oh, I more than tripped over it. I got it nonstop.

**Steve:** And so I went over, and I said, you know, guys, I don't - we don't have all the facts. We don't know what's going on. I'm unwilling to pile on someone who I don't know, and it just doesn't seem fair to me. And I said, so we just don't have enough information. And Bear, someone must have said hey, you know, this is being talked

about over at GRC. So he, to his immense credit, came over and said, hey, it's me, I have a thick skin, so let's talk about this.

And I what I had posited was that TWiT was in transition, and this is from things that you had said, Leo. I mean, you were quite literally a cottage industry for quite a while, just down the street. And Bear commented in the newsgroup, for example, that he had found the problem and turned it off, but somebody else turned it back on, and that that was really the way that this came to notice and was the problem that it was. And what that really said was that you guys are growing, and that there's a need for policy.

**Leo:** But let's put that aside because of course that's a bad mistake. But the larger question is, and this was the thing that I really would love to get to the bottom of, is that as long as a website is being changed in any way, you're going to have security flaws, and that breaches are not - this is my real question. Are breaches uncommon or common?

**Steve:** And I'm not an expert. I cannot say.

**Leo:** Bear is of the opinion and knows - runs a very big site and knows others who run other well-known sites, and he's of the opinion that, as we have become more and more high-profile, we are certainly getting more attacks; and that it is very, very difficult, if nigh impossible, to prevent breaches of some kind. The question is really how quickly you see them and modify them. But on the other hand, people like our commentator here do raise the point, well, Magic John says you shouldn't allow file systems to be written to and shouldn't be able to do - but I think that that's the point of an exploit is that it somehow allows access, higher level. I don't know. I'm not an expert. I don't know.

**Steve:** Well, and we don't know, none of us know except whoever is your, I mean, the gurus of your web server, for example...

**Leo:** There's the guy, he's the guy who runs our server, so...

**Steve:** Are all of the directories read-only except, like, very carefully tuned so that the equivalent of cgi-bin, I mean, for example, at GRC there's only one location where anything can be run. Every...

**Leo:** And this was my problem with PHP. PHP, unlike cgi-bin, can be put in any directory and run from any directory. So, I mean, admittedly, all directories should be write-proof except that you have to have some directories that are going to be written to; right?

**Steve:** Well, okay. So what you have with PHP is similar to what you have with ASP, Microsoft's Active Server Pages, the idea being that it's - and our listeners are tired of hearing me talk about JavaScript. JavaScript is scripting run by the browser. So PHP...

**Leo:** Right. This is all server-side.

**Steve:** Exactly. PHP, like Active Server Pages and other technologies, there is server-side JavaScript also. Anyway, those are scripts being run by the server. So there the server is not delivering static, prewritten HTML. It's actually running the PHP code to create these pages on the fly. And so that's, I mean, it is a powerful technology, but it's also one that you have to be far more careful in deploying. So what I think we can definitively say is that it is, from everything we've seen, all the evidence, and it's not like TWiT is the only site on the planet being hacked, I mean, we're talking about hacks all the time. I mean, like, RSA, the security company, is getting breached by exactly these sorts of things.

So what we can definitively say is it is really hard. There are old curmudgeons out there who want to say, oh, no, this is really easy. It's not easy. I mean, my site is easy because I don't have any server-side scripting stuff like that. I haven't had to deal with this. I do have static HTML. And then my own stuff generates dynamic pages like the ShieldsUP! page and so forth. So that's why I really can't speak to the challenges, because I haven't faced them myself. But we know that security is difficult. And so I think, I would imagine, that the lesson that TWiT has learned is it's time to really focus on security. I mean, there probably shouldn't be a situation where Bear could disable something which is now a known problem, and somebody else could turn it back on again. So...

**Leo:** Well, yeah. And that was a miscommunication, and that certainly, you know, that's not going to happen again.

**Steve:** Yeah. And so there needs to be a single point of responsibility, somebody who really has that job, and sort of like moving...

**Leo:** Well, that's a problem because we have web developers that are working on the site. Now, I think a big problem was, with this previous TWiT.tv development, we didn't have a production and a development site. We were doing stuff - code went live live. And that was a big mistake. We're not going to do that anymore.

**Steve:** So also there's some live and learn.

**Leo:** Yeah. But I don't know. I don't know. I don't understand it.

**Steve:** It isn't easy. I mean, we talk about it...

**Leo:** I don't want to have to hire a full-time security sysadmin, and I can't afford it. So I don't know what the answer is. I don't know what the answer is.

**Steve:** Yeah, well, I mean, you're wanting to have a fancy, feature-rich site, and the truth is that's hard to do. This technology is fundamentally flawed from a security standpoint. This technology, as I've said before, if you have a system where you can

upload an SQL command that the server will execute, I mean, that's just crazy. It's easy to implement, but it's a disaster from a security standpoint. And frankly, the idea of pages being PHP so that all of your server needs to be executable, that's horrifying, too. I mean, it didn't have to be that way. But, boy, is it simple.

**Leo:** And I would say in my defense that Richard Clarke, who used to be the counterterrorism czar in the U.S. administration, said every major U.S company has already been hacked by the Chinese government. So, I mean, if it were really easy - I don't know. He says it's pretty clear the U.S. government did the Stuxnet attack. I thought that was interesting. I think there was some minor Israeli role in it. So we wrote Stuxnet, according to Richard Clarke. Israel might have provided a test bed, for example. But I think that the U.S. government did the attack. I think this is fascinating. And we talked a lot about Stuxnet. You of course immediately said, no, it's governmental. It's just not clear which government.

And I think the attack proved what I was saying in the book, which came out before the attack was known, which is you can cause real devices, real hardware in the world, in real space, not cyberspace, to blow up. He's talking about cyberwarfare. He says the war's already begun, and we're losing. China has basically hacked every single major company in the U.S., says Richard Clarke. And a few small podcast networks.

**Steve:** I guess my - when we hear that, it's easy to think, oh, my god, unplug everything. But the fact is you're getting far more benefit from your website than it costs you.

**Leo:** Right. And you're not - nobody's getting credit card numbers by hacking our website. They're not getting anything, you know...

**Steve:** Right.

**Leo:** They're getting our website. They're getting us. But they're not getting any credit card numbers or anything. "'I'm contributing to FUD,' says Hogan." I'm just quoting Richard Clarke, who seems like he has a dog in this hunt. But I don't know. See, I can't say anything. I'm just going to shut up.

Question 5, G. Sveddin in Southern California about Spidey - or SPDY. I call it "Spidey" because I'm a Spiderman fan. With respect to SPDY, I'm surprised to hear your excitement about the server push mechanism. Seems to me a server that can push content without a request will waste bandwidth on unwanted content. For example, of NoScript is installed and functioning, why should I download the huge JavaScript framework libraries on many sites now? Another example is turning off images when mobile browsing. With such bandwidth caps and costs, I wouldn't want to download content I don't want. And this actually is an interesting question. I kind of..

**Steve:** Yeah.

**Leo:** It came to me, too. I understand why there's a scaled-down version of server push which hints to the client what they should be asking for next. But that appears to hinder the speeding-up of the bandwidth as the client would still have individual requests. I would think the parallelization of requests and content returns would have a better payoff. Let the browser request all it wants, after filters like NoScript or image block, and fill the bandwidth with only desired content. All in all, SPDY sounds good. But from the little I've heard and read it seems focused on delivering all content, a business perspective, rather than just desired content, a user perspective. I'd love to see a more middle-of-the-road perspective. I hypothesize adoption would be much faster on both sides. Thanks, G. Sveddin. What do you think? Is it anti-user and pro-business?

**Steve:** Well, it certainly changes the model from a client-side request to one which does offer some server push. Now, in fairness, the server push side is regarded as an advanced feature. It isn't part of the base SPDY spec. Both of those things, the server hints and the server push, are sort of more on the experimental fringe. It's like, well, this is in the spec. We didn't want to, like, not design it in so that we wish we had it later because maybe it would be a good thing. My sense is it isn't something which is being actively used and deployed at this point, probably for much the same reason. And I really do, I mean, as a proponent of only getting what we ask for and of things like NoScript, he's right. Some of these scripting libraries are just big blobs which someone gives you the URL to it, and your browser's going to suck it down and take all the time to do that, which is crazy if you've got scripting turned off.

So, no, I think his point is well taken. For what it's worth, my sense is it isn't happening now. And I would argue that the business perspective is giving the user the most responsive page possible, and that you could, for example, give them images. Well, of course, here he was saying he wants to be able to turn images off. So that's a problem. Anyway, so I guess it's certainly a tradeoff, and I thought he raised a good point, which is why I wanted to include it.

**Leo:** Brian M. in Edmonton, Canada wonders about SPDY and CDNs: I've been catching up on the discussion about SPDY. I saw a major problem. Many websites, my own included, use Content Distribution Networks (CDNs) that serve static assets - images, JavaScript, et cetera - from geographically distributed servers. Many CDNs use anycast, a topic you discussed a few weeks ago. With SPDY, wouldn't all of my assets need to be pumped through the same servers that handle my main dynamic content? That could be catastrophic for many sites. It would slow down my app servers. It would be more expensive because the price to serve static assets from app servers is quite a bit more costly for a few different reasons - processing, bandwidth, et cetera. Or am I missing something? Thanks for the great podcast. Brian.

**Steve:** Okay. So what he's assuming is that this - we talked about the notion of a single connection being made between the browser and the server, and sort of had to get over the idea that that could be faster than multiple separate connections. And as long as we've got a single connection highly used, a fully utilized single connection, which is what SPDY for the first time allows because we can overlap things, that single connection can be overall much more efficient than making separate connections. But that absolutely doesn't keep you from establishing additional SPDY connections to entirely different domains.

So the idea would be you would have one connection per domain or per server from the browser out to that remote asset, and a different connection to the CDN. So you're still going to get the benefit of SPDY. In fact, the CDN might not support SPDY, so the browser would seamlessly use a non-SPDY connection to the CDN, yet a SPDY connection to the server in order to pull all of those assets from one place. So there definitely is no assumption or presumption on the part of SPDY that it will only be making a single connection to a remote location. It'll make a single connection per server. But if you're pulling content from 36 different places, it'll set up 36 connections, and those that are SPDY can run faster than they would otherwise.

**Leo:** So no problem.

**Steve:** Compatible with all of that, yeah.

**Leo:** Jon in Kentucky wonders about the future readability of computer media. Oh, don't we all.

**Steve:** Ohhhh.

**Leo:** Anybody with a Zip drive knows, or is it disk, I should say. If you have the drive, you're okay. Mr. Gibson, sir. My father's church is getting ready to celebrate its 100th anniversary pretty soon. They plan on opening the time capsule under the cornerstone - that's neat - and adding a CD with photos on it. It hasn't been opened since the '60s. I was thinking about this. What would be better for them to use, figuring the technology still exists to read them, a CD/DVD or a thumb drive? Thanks for your time. Jon.

**Steve:** You know, this is a great question.

**Leo:** It is.

**Steve:** Because, for example, I have around here some MFM hard drives with data on them, yet no MFM controllers. Or if I have an MFM controller, it's got an ISA, the original bus from the PC, and I have no motherboards with ISA buses anymore. It is a really good point that we think in terms of the technology we have and are using, but this whole issue of will we be able to read it in the future, not just will the medium itself hold up its integrity, I mean, that's a question, will the writable CD be readable even if you had the technology? I mean, well, if I had an eight-track tape from back in the day, I don't have any eight-track tape players. Of course vinyl has come back into vogue, so there's an exception to the problem of needing a turntable to play the disks that you and I were listening to in college, Leo.

**Leo:** Well, you know what you should do, you should print prints. Because those will be good.

**Steve:** That's exactly right.

**Leo:** And better yet, put it on papyrus. It's the only thing we know lasts thousands of years.

**Steve:** Put it on acid-free paper.

**Leo:** Yeah, exactly.

**Steve:** You want it on acid-free paper so it will not yellow. All of my DEC manuals from the early 1970s are just...

**Leo:** They're crispy, yeah.

**Steve:** ...incredibly yellow because they were not on acid-free paper. I printed all of the "Passion for Technology" books that I published on art-grade, acid-free paper, not that I particularly thought they'd be in any time capsules for any reason, but I just thought, well, I want them to look the same way a hundred years from now.

**Leo:** But music, if you're talking about music, a vinyl record is going to be very simple to reverse-engineer because you can look at it, and you say, oh, I see, these are wave forms. I just need something to read these wave forms. Maybe, it's possible a CD will be so easy to reverse-engineer. Certainly there'll have been a lot of CDs around, and presumably any future archeologists a hundred years from now will certainly know how to read CDs. Hard drives, a little more opaque, if you ask me.

**Steve:** Yeah, I was thinking maybe put the CD and a USB CD drive in. That is...

**Leo:** Ah. A reader. Put a reader in.

**Steve:** Put a reader in. Now, the problem, of course, is the USB interface. We're already moving...

**Leo:** That's gone.

**Steve:** ...to USB III. And so a hundred years from now we're not going to have, well, I don't know when they're going to open this again. But 50 years from now we're not going to have USB. We'll be onto some Twilight Zone technology. I mean, Intel's already pushing stuff that's, like, can it even work, it is so fast. It's like, okay.

**Leo:** Make prints. The human eye has not changed its interface in tens of thousands of years.

**Steve:** That really is the answer is print these things out. Instead of doing a CD, print them out. That's what you need.

**Leo:** But this is actually a very big topic. And what I find fascinating is it is now the province of librarians because librarians have become information specialists and archivists. And they are at the forefront of this. It's fascinating stuff. It's something I talk about on the radio show a lot because this is something real people do want to think about and really don't know what to do about.

**Steve:** Well, especially photos that are sort of, by definition, for the future.

**Leo:** Right.

**Steve:** But if the future can't look at them, then you've sort of defeated your purpose.

**Leo:** Jared in Western Australia has been wondering about spare sectors. That was not a good Australian accent. That came out all wrong. Listening to your discussions on the show about how SpinRite works and how it shows the drive's bad sectors and induces the drive to map out weakened sectors before they become terminal is all well and good. But I've never heard you discuss what happens if the drive runs out of spare sectors and cannot map anymore bad sectors? Oh, you're in trouble if that happens [laughing]. You said that sectors are mapped out as bad at the factory. So when the drive is new, its capability to map out additional sectors would already be somewhat compromised. Is there any way for us to know where the drive stands? Can you run out of places to put stuff?

**Steve:** Oh, yeah. You can.

**Leo:** It's more the drive allocation table, I would guess, than, I mean, the drive you can find spare sectors. It's the drive; right?

**Steve:** Well, what happens is that drives generally store extra sectors at the end of their track. And I've talked about mapping sectors. Well, the way that actually happens is they will move all of the sectors from the bad spot to the end of the track down by one. They shift the whole, all of that block of data down so that the drive can continue to read at normal speed. It just sort of reads past the bad sector and finds the one it was looking for, that it expected to see in the bad spot. Now it finds it a little bit further down. So they actually just shuffle the balance of the track downwards toward the end of the track.

Well, there is a limit to how far they can go. And this is the one place that that SMART, S.M.A.R.T., the Self-Management Analysis And Reporting Technology, SMART is the acronym, and it will show you - there are different parameters that SMART has. And one

is relocated sectors. And what you don't want is for that to get down to zero because essentially it doesn't - the problem with SMART is that manufacturers never wanted to tell us anything about what was going on inside the drive. They wanted it to be a black box that we buy, and we're happy with.

But Compaq, back in the old days, said no, we insist - Compaq was like a major IBM clone manufacturer, for those who don't remember the name. They insisted, and they had enough purchasing power to force drive manufacturers, the big ones at the time - Western Digital and Seagate, Micropolis and Maxtor - to force them to give them a means of asking the drive how it feels. And Compaq actually famously used SpinRite on their dock where they were accepting drives. They would over-order drives and use SpinRite to prequalify them before putting them into their machines, and would send back the drives that SpinRite said were weaker, less good, than the majority of the drives. And manufacturers didn't like that they were doing that.

So they said, okay, fine, we'll - and that's where this whole SMART system came from. It is a sadly weak specification. It's not something that anyone should be proud of. Manufacturers were compelled to do it, or they would lose a major vendor in the form of Compaq. And so they added this. The point is that it doesn't give you a lot of data. It sort of gives you a happiness indication. And when that runs to zero, then you're really in trouble. So that's one of the things that SpinRite monitors while it's doing its work. There's a screen there that monitors the SMART parameters on the fly and also is able to show you the rate at which error correction is occurring, which allows you to get some sense for the relative health of the drives.

So they are black boxes. The manufacturers don't want us to see what's really going on. There is no specification for asking a drive for its bad block tables, for how many spare sectors it has remaining, for which tracks are almost out of spares. There's no interface like that, that is available to the outside world. So we do the best we can. And of course running SpinRite on the drive from time to time allows the drive to at least know what's going on with it, for what good that does.

**Leo:** Next we got Bob Lindner in La Crosse, Wisconsin. He found news of Astaro and, brace yourself, Stochastic Fair Queuing: Guys, I thought it worth a mention, I've been playing with the Astaro product, the home ISO download. Good for you. I think this is a really great product. A few weeks ago I enabled the QoS (Quality of Service) settings. The Astaro Security Gateway (ASG) allows you to enable an Upload Optimizer and a Download Equalizer. They are described as follows by Astaro:

The download equalizer: If enabled, Stochastic Fairness Queuing (SFQ) - this is becoming the acronym winner of the week - and Random Early Detection (RED) queuing algorithms will avoid network congestion. In case the configured downlink speed is reached, packets from the most downlink-consuming stream will be dropped. That's the download equalizer. The upload optimizer, if enabled, will automatically prioritize outgoing TCP connection establishments, that is, TCP packets with the SYN flag set; acknowledgment packets of TCP connections, that's the ACK flag set and a packet length between 40 and 60 bytes; and DNS lookups, UDP packets on port 53. I thought you might find this interesting. Thanks for the great podcast. Bob. Talk about dumping this in your lap. So there. What is that?

**Steve:** What's really interesting, okay, we talked about the first part, the download equalizer that uses this Stochastic Fairness Queuing and Random Early Detection, the idea being that, as a buffer is filling, the router will begin discarding packets statistically

more often to prevent the buffer from ever getting completely full. And that means that, if a particular stream is hogging the bandwidth, the chances are its packets will get dropped, which will send its endpoints the message that they need to back off and slow down. So that's this RED, Random Early Detection, is the most often used Active Queue Management - there's another acronym, AQM - which is what we're going to be dealing with now for the next decade, is smarter queue, active queue management.

But the upload optimizer is really interesting. This is a move to the front algorithm. It says, if enabled, this option will automatically prioritize outgoing TCP connection establishments, that is, TCP packets with a SYN flag, as in synchronized set; acknowledgment packets of TCP connections with the ACK flag set and short packets, that is, between 40 and 60 bytes; and DNS lookups. So this is neat because it means that the little tiny packets which we need in order to keep our connections running move to the front of the queue. They don't have to wait behind a long delay of blob that's being uploaded or downloaded. The Astaro technology gets them out right away. That doesn't delay the blob because these are necessarily very small packets. They're going to be 40 bytes rather than 1,500 bytes. So in terms of the packet delivery time, there isn't much cost, but they do allow the system to act as if there is no bloat in the buffer. So bravo, Astaro. They clearly gave this some thought, like before the rest of the industry had.

**Leo:** Wow, that's pretty smart.

**Steve:** Yeah.

**Leo:** Amazing. Our last question, Mr. G., from Kyle D. in B.C. [laughing]. How did I do that? He wonders about private email. Steve, I have a question about private email. I currently store way too much information on Google, and I'm trying to move some services away. Do you know of any good email services with a better privacy policy than Gmail, Hotmail, or Yahoo!? I normally use PGP for anything sensitive. I would just like slightly better terms of service/privacy policies when dealing with email. Thanks, great show. Kyle D.

**Steve:** So that's really one for you, Leo. I have my own servers at GRC, so I've never needed to think about the repository. But for people who, like with Gmail, have all of their mail living somewhere, Kyle is becoming a little nervous about that. And so I thought - I wondered if you had any suggestions for probably smaller solutions who maybe are a little more honorable.

**Leo:** I do. But I would say, first and foremost, that any service that you're going to use is - you're trusting them. And maybe we've all lost trust in Google, Yahoo!, and AOL; maybe not. But most services have similar access to your content and, if they provide antispam services, are doing similar scanning of the content. So, I mean, that's just the nature of antispam. They're looking at keywords in the same way that Google does. Google does it for both antispam and for advertising, so maybe you don't like that.

Since he uses PGP, I'm going to recommend a Canadian company called HushMail, which is an encrypted service. Now, I should point out that HushMail, if presented with, like anybody, I think, presented with a governmental subpoena from law

enforcement, will disclose. So the reason they can do that is because I guess they have the key. Anyway, I do think Hushmail was created with exactly this point of view. They have a privacy policy. They do not do spam filtering because they don't look at your mail. They talk about in their privacy policy web logs and cookies. They do log IP addresses. Cookies are not used for marketing purposes, however can be used to track your settings. It is not possible for users to view and update their personal information, but this feature will be available in the near future. Hush must be able to authenticate the user, and they do that. They require a securely sent Hushmail message. But that's the only information they have, basically. Pretty much this is a good solution, and it is encrypted using PGP. Phil Zimmerman worked with them.

Another solution that you might want to look into, I think I'm moving my email to them, is a U.S.-based company called Island Email. I know about them because they support MailRoute, and so they have very good antispam filtering, but that means that MailRoute is scanning your email. Your Internet service provider is probably similar to one of these services in the sense that they also may do some scanning of your mailbox. Unless you host your own email server, I think that's the only way to for sure know that only you control that content.

Finally, I'll tell you what I am currently using for my IMAP. They were recently purchased by Opera, so that may be a nonstarter. It's called FastMail. I like FastMail a lot because they are, I think, the most sophisticated IMAP server out there. And Opera has not changed them in any way. In fact, the only thing Opera has done, as far as I can tell, is give them more resources.

So if you're looking for hosted email, understand that hosted email is always going to have some of this problem. I guess HushMail is probably the best choice in hosted email because of PGP encryption built in. And they have a very aggressive privacy policy.

**Steve:** And they've been around for a long time. And I know that a lot of people trust them. So, yeah, I think HushMail is a great thought, Leo.

**Leo:** Yeah. "In some countries government-sponsored projects have been set up to collect" - this is on HushMail. They have a good article, "How HushMail Can Protect You." But there was a recent news story, you should probably read it, that says HushMail does give up some information. "HushMail does not put you above the law. We're committed to the privacy of our users and will absolutely not release user data without an order that's legally enforceable under the laws of British Columbia, Canada, which is the jurisdiction of our servers. However, if we do receive an order, we're required to do everything in our power to comply with the law."

I think what they're saying here is, well, this is the next one. I thought data was always encrypted. "When one HushMail user sends an email to another HushMail user, the body and attachments of that email are kept on our server in encrypted form, and under normal circumstances we would have no access to that data. We cannot pick an arbitrary encrypted email message off the server and read it. However, since HushMail is a web-based service, the software that performs the encryption either resides on or is delivered by our servers. That means that there is no guarantee that we will not be compelled under an order enforceable under the

laws of British Columbia, Canada, to treat a user named in an order differently and compromise that user's privacy."

What they're saying is we could be forced to compromise our own encryption software so that law enforcement could read your email. But I don't think you could do it after the fact. I would suspect that that would mean in an ongoing conversation. But anyway, read it. And the other thing to do, which is not a bad idea, is to use PGP encryption on your email. And I do that.

**Steve:** Yes. If you use your own encryption, then you're just transferring a blob from one place to the other. So you're doing end-to-end encryption, or as we have called it before, pre-Internet encryption, where you take responsibility for encrypting. Then nobody can read it.

**Leo:** Right. Unless the law enforcement comes to your house, then says give us the keys. And now the courts are going back and forth on this one. The most recent court decision says they can demand keys if they have other evidence that there is illegal activity going on. They can't do it as a fishing expedition. So there you go. I think HushMail is probably the best choice in all of that.

**Steve:** Yeah, great idea.

**Leo:** But it's web-based, and he wanted - but he did want web-based. Steve, we're out of time. But you know what, it's perfect. In 30 seconds we'll begin This Week in Google.

**Steve:** Yay.

**Leo:** You couldn't have done it better.

**Steve:** Nice podcast. Lots of coverage of all kinds of interesting things. So I think we did a good one. I have no idea what's in store for next week, but I can promise something interesting.

**Leo:** That's good. Tune in every Wednesday, 11:00 a.m. Pacific, 2:00 p.m. Eastern time, 1800 UTC. You can watch us live. But if you miss it, we have audio and video available. Video available from us exclusively at TWiT.tv. Audio we have a variety of formats, but Steve's the one with the low-bandwidth one, the 16Kb version. And you can get that directly from GRC.com. That's where SpinRite lives, all his free stuff.

**Steve:** And I have even lower bandwidth versions in print.

**Leo:** Yes. You don't get to hear our voice, but you can see every single word we

speak, thanks to Steve and Elaine, at GRC.com. If you want to ask a question, GRC.com/feedback is the place. And if you want SpinRite, the world's best hard drive maintenance utility, absolutely GRC is the place to go and get that. Steve, thanks so much. We'll see you next week

**Steve:** Thanks, Leo.