SECURITY NOW!

Transcript of Episode #362

Listener Feedback #148

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-362.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-362-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We're going to take a look at the security news of the week, maybe throw in a little sci-fi here and there, and then he's going to answer some great questions from you, the members of the audience. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 362, recorded July 25th, 2012: Your questions, Steve's answers, #148.

It's time for Security Now!, the show that protects you, your loved ones, their privacy, their security online. And we couldn't do it without this guy, the Explainer in Chief, Mr. Steven Gibson. Here, let me just move the painkillers out of the way.

Steve Gibson: I didn't ask you before we began, officially. You are recording this; right?

Leo: I am recording this, sir.

Steve: Normally I ask before.

Leo: No, no, no. You need not ask. I have trained professionals now who push the buttons, if I'm incapable of doing so.

Steve: When was it that you recaptured the feed from Justin?

Leo: Oh, gosh, yeah. We haven't done that in a while, though. I don't know if we've done that since we moved here. This is now - well, actually, maybe we have.

Steve: Speaking of which, it's coming up on a year, isn't it.

Leo: Yesterday was one year. And Dick DeBartolo flew out. We had a little parade, recap, a party, sheet cake. It was so fun. It has been one year. Isn't that amazing?

Steve: Wow, yeah.

Leo: And I'm so happy here. I'm so happy here. We really are.

Steve: Well, my god, you've built yourself a little kingdom. It's fantastic.

Leo: Yeah, and I'm the Napoleon of this little kingdom. No, it is so great. And the real joy of this - and it sounds insincere, but I'm deeply sincere - the real joy in this is the people I get to work with, like you. I get to talk to you every week. If it weren't for this, I probably wouldn't.

Steve: No, because we're both busy people.

Leo: The people in the studio that I get to, I mean, I just love the team. And so that's a real blessing, to be able to come to work. Because I know, I've been in jobs where I had a stomach ache every day of going to work. And so it's such a joy to come in here.

Steve: I had a job once, I was in my mid-20s, I had to put alarm clocks down the hall from the bedroom to the bathroom, just to drag me along.

Leo: You've worked for yourself, though, for almost your entire life; right?

Steve: Ever since that job, yes.

Leo: You see, unlike me, you're smart. You learned the lesson. I had to wait till I was 50 before I figured that part out. But you learned the lesson early. And this is, I think, a really important lesson. And I'm trying to get this into our schools, certainly the high schools...

Steve: That everybody else is a moron?

Leo: No, well, shh, that's the subtext. But the real lesson is you don't learn to get a job, you learn to make a job. And not everybody can do this, and certainly I have 25 employees, thank god they're employees. But if you can, if you can learn to be the person who creates their own job, you will, I think, often be happier. You may not be more wealthy, but you will be happier.

Steve: You can work a lot more and have periods of stress, but at least it's yours. It may fall, but it's yours.

Leo: Nothing worse than, in fact, I remember I saw a study years ago of what people are happiest in their jobs. And it was traffic cops and orchestra conductors were way up there. And it was because they were calling the shots. The thing that makes this more stressful than anything else is having somebody else tell you - at least it is for me, and I think this is true in general - what to do. That's stressful.

Steve: Well, and that's why I have a small company, and I'm just a really crappy manager, is I don't want to tell other people what to do. I don't like looking over people's shoulders and being a manager. I hired people back in the day and just wanted them to go launch and...

Leo: Do it. Go. Just do it.

Steve: Yeah. And it would work. It was sort of a parabolic launch. It would go up and reach its crest, and then it would begin coming back down, and we'd have a crash landing about four or five months later because they just realized, oh, look, if I don't come back from lunch, Steve apparently doesn't notice.

Leo: Yeah, we have that problem here, too. But, see, I did a really smart thing. I hired Lisa. And she is a good manager. She knows how to manage teams. And I let her do that. So this is the other lesson. Create your own job. And then, if you can, hire people to do the parts of it you don't want to do.

Steve: And that's what I have. It's Sue and Greg, famously, handle customer support and all of the books and bookkeeping and finance and stuff. So I get to do the fun stuff, and they're happy that I give them total freedom with scheduling to do what they need. So it works.

Leo: Right, exactly. That's how TWiT happened, by the way. I wanted to do the shows, but my dream - and I thought this was pie in the sky - was to do a show, like do this show, and then get up and walk away and have everybody do the stuff I didn't want to do - produce it, edit it, push it, all that other stuff. And I have it now. It's so cool.

Steve: Yes. As I said, kingdom.

Leo: Kingdom. But I get to do what I'm good at, and I avoid what I'm bad at. Boy, was I bad at doing taxes. Boy, was I bad at doing payroll.

Steve: I'd be in prison. The IRS...

Leo: I would be in prison, too.

Steve: Not because I meant to do anything wrong. I would have been delighted to write them a check. I just would have kind of forgotten about it.

Leo: Exactly. Anyway, we have security news. And this is a Q&A episode, 148.

Steve: It is? Yes, it is. No, that wasn't a question.

Leo: Not how many questions. This is the 148th question-and-answer episode.

Steve: Yes, and otherwise we would never finish this podcast.

Leo: Only 10 questions.

Steve: Actually, we haven't started it yet, so perhaps it's fair that we wouldn't finish it. We had a bizarre event since I last spoke to you, which was probably the single most tweeted observation from the people who follow me and wanted to make sure that I knew about the news of unbreakable crypto which allows you to store a 30-character password in your brain subconsciously so that no one can torture it out of you.

Leo: What? So, like, you don't even know it?

Steve: You don't know it. Now, the bad news is it doesn't do that.

Leo: Oh.

Steve: It's not 30 characters. It's a little better than 30 bits. So unfortunately the headline - and this was in ExtremeTech. And for what it's worth, I'm seeing, whatever ExtremeTech talks about, I get tweeted. So we have a huge following of ExtremeTech watchers.

Leo: It's a good site. It was started by Bill Machrone. You remember Bill; right?

Steve: Oh, of course I know Bill. Yeah, yeah.

Leo: He was the former editor-in-chief of PC Magazine for years. I don't know who owns ExtremeTech these days, but his idea was let's - PC Magazine was getting its lunch handed to it by sites like Tom's Hardware and AnandTech.

Steve: Right.

Leo: He said, well, we should do an enthusiasts site. And that's the result is ExtremeTech. So this is the article here. But they got it wrong, unfortunately.

Steve: Well, yeah. The headline is wrong. But I want to talk about it because what it does, I mean, it is serious neuroscience. It's being presented at this upcoming USENIX conference. And it really does work. So they were a little wrong in this 30-character password storage. It doesn't do that. It actually stores something with an entropy of about 38 bits, which is more like 6.5 characters. But...

Leo: That's not very strong, is it.

Steve: Well, and it's authentication only. Anyway, we'll get into it. I imagine you've got a sponsor you want to talk about, and then we'll get into our show, and we'll do that, and a bunch of questions.

Leo: You are so good. And what a great tease. Steve Gibson, the Explainer in Chief. And we're going to show you how you can store this in your subconscious; right?

Steve: Yes, actually, there is an online version of this that's a little Flash app that people can play with. I would imagine...

Leo: It would take me an hour to do this, so I'm not going to do it on the air. But maybe tonight.

Steve: Yeah. I would imagine that people want to understand what it's about. That's what I'm going to explain. And then they can go look at it, if they want to. It's been referred to as sort of like Guitar Hero and the way that works.

Leo: I play the game a little bit. I'll play it, yeah.

Steve: And we'll do our user-directed episode of Security Now! this week.

Leo: Yay. Steve Gibson, time to plant something subliminal in my mind.

Steve: Okay. So this is very cool and sort of tangentially security related. The idea is that neuroscientists have long known that it's possible to teach us things, sort of like muscle memory, in the way that, like when you're learning to play tennis, you have to be very conscious about everything you do, where your feet are, how you move, where your balance is, how you're gripping your racket. It's very conscious. And if you're serious about the game, you'll get a pro to work with you and look at you and sort of get you started down the right path. But what happens is that, after some length of time, that becomes unconscious, or subconscious, that is, you're directing yourself at a higher level, and it's often called muscle memory, where it's...

Leo: Hey, there's no muscle in my brain.

Steve: It's not actually your muscles that are remembering. But those sort of subroutines are established, and then you're calling them, rather than running them by hand, to use a computer analogy. So this was "Neuroscience Meets Cryptography." And the title of the paper is a little worrisome. It's, I'm not kidding you, "Neuroscience Meets Cryptography: Designing Crypto Primitives Secure Against Rubber Hose Attacks."

Leo: Oh, dear.

Steve: So and of course a "rubber hose attack" is the slang for hitting somebody.

Leo: Torture, yeah.

Steve: Exactly, hitting somebody over the head or somewhere to get them to tell you what secrets they have. So what these guys designed is a means for training people on a sequence of events such that they're unaware of the sequence, but on a subconscious level they become better at following one sequence than another. And so this, for example, you don't have the ability to simply sit down and type out a password that you're not consciously aware of. So the way the system would work is you would first authenticate yourself normally. You would use a username and password, things that you know, maybe even something that you have. But then this next level of authentication would - it would know who you're claiming to be, and it would know the sequence that you had previously been trained on.

So this is all kind of Jason Bourne sort of stuff. And the idea is it would give you a sequence of events that you respond to, knowing what you're supposed to be better at, that is, that you have been previously trained to be better at, than somebody trying to impersonate you. And what they found was it works. So what's online, and our listeners can play with it, it's BrainAuth.com/training.html. And if you go there, you skip around a couple of initial screens. If you just go to BrainAuth.com, there's this strange thing about how you're going to get paid $5.

Leo: Oh, well, there, now you've convinced me.

Steve: But there's a disclaimer saying, oh, whoa, we're not doing that anymore, so ignore that. But it's like, okay. Or just go to /training.

Leo: Oh, whoa, whoa, whoa, it's going so fast, I can't keep up. It's like Guitar Hero.

Steve: What they were doing was, initially, they were soliciting people to train up on these sequences for their research, and when they finished it, they would get a coupon that was redeemable for $5. And so they were offering five bucks, probably for some students at Northwestern. This is a guy at Stanford, some people at Northwestern, and SRI, the Stanford Research Institute, put this whole thing together. So what you'll see there is six columns...

Leo: I'm terrible at this, as I am at Guitar Hero.

Steve: Yeah, and the circles drop, and the columns are ADF and JKL, which are two groups of three keys on the keyboard. And so the idea is it's choosing what drops in what sequence, and you're supposed to be pressing the keys at the right time to get these things to drop into the holes that are down at the bottom. So with six symbols, what they were trying to do is they're trying to build associations, temporal associations between pairs and triplets and quadruplets.

Leo: It's too hard. I can't do it. I can't do it.

Steve: So where this 30-character thing came from is actually what's known as the "Euler walk." If you take six things, and you draw interconnecting, bidirectional arrows between each one and every other one in some sort of big star formation, then the Euler walk traverses each path through that exactly once. And so that does result in a 30-object sequence. And so what happens is this thing, when you're using it, it arbitrarily chooses an Euler walk through this six-character alphabet, essentially.

Leo: "Euler" is E-u-l-e-r, like the Euler Constant?

Steve: Exactly.

Leo: Okay. He's a famous mathematician.

Steve: Way old and long dead. But he...

Leo: He would hate this game.

[Laughter]

Leo: [Geezer voice] What are you doing?

Steve: They explored all these things a long time ago. So anyway, so what happens is, without you being aware - and this also throws in non-training sequence events, so you're not just training up on exactly that Euler walk, it sort of throws - it deliberately throws in other things. And the point is you don't know, thus, what parts of what is happening on the screen is part of the secret sequence you're being trained and what isn't. So you're never given it. You're never explicitly told, here is your 30-character sequence. Instead you just play this game several times for 30 to 45 minutes, they say in their paper, and they run through it a number of times. Then they wait a week. They wait one week, and they test you again, and two weeks and test you again. And what they found was there's some loss of retention after the first week, but then it sort of flattens out, and you don't lose nearly as much in the second week. And it's sticky enough that they're claiming it could be used for authentication. And so again...

Leo: But how, I mean, I don't know it consciously, the keystrokes, but somehow I can trigger it?

Steve: Well, no. The idea would be that you're actually getting better. Even though you're not thinking, Leo, that you're very good at that right now...

Leo: I'm terrible at it, yeah.

Steve: You did spend...

Leo: I bet a pianist would be very good at it.

Steve: If you did spend an hour, there would be things it would be deliberately doing, like SKD.

Leo: No, I'm recognizing some patterns, actually. I do see some things repeat, yeah.

Steve: Right. And so you wouldn't be able to speak it, necessarily, but your body and like those lower levels of your autonomic stuff...

Leo: Autonomic nervous system.

Steve: Yes, it would be sort of acquiring a bias. And that's really what this is.

Leo: It is. I have a bias towards doing it right, that's right, yeah.

Steve: Yeah, it's giving you a bias towards things that it has deliberately caused you to expect. And so at some time in the future it could test you for that bias, and you would have a particular bias.

Leo: Oh, I see. So it would give you the same game again, basically.

Steve: Yes, exactly. And that's why it couldn't recognize you from a crowd. But if you're claiming to be you, then it could test...

Leo: I should be better at this than somebody off the street.

Steve: Exactly.

Leo: In this particular sequence that I get again and again.

Steve: Yes. You should be better at the one particular pattern that you had been previously exposed to.

Leo: Right.

Steve: Thus the Jason Bourne, sort of.

Leo: Yeah. He doesn't know he's Jason Bourne. But he is.

Steve: He triggers, yeah.

Leo: This is going to drive me insane, and I can't seem to stop.

Steve: So SISL, Serial Interception Sequence Learning. Anyway, that's what that is. Many people sent the link from ExtremeTech to me, and so I wanted to thank them and acknowledge it and explain. And unfortunately it's not storing a 30-character password in your brain, but it's creating a something unique in you that nobody else would have, that could at a later date be checked for.

Leo: That makes perfect sense. That makes perfect sense. And the NSA couldn't suddenly come to you and say, okay, what's your sequence, because you have no idea.

Steve: Exactly.

Leo: It makes sense. And it is muscle memory because I would guess it's probably as much muscle as it is brain. I don't know.

Steve: I think at some point after about 30 minutes you probably zone out, and you just...

Leo: You do, yeah.

Steve: You just sort of start doing this...

Leo: Exactly.

Steve: ...and you're thinking about other things while you're continuing to try to press the right keys. And it probably actually does sink into your subconscious. There's a 17-page PDF, if anyone really wants to go into this. If you just put in, probably "rubber hose crypto"...

Leo: That's pretty much a unique phrase.

Steve: Yeah, I think that Google will find it for you. And it's interesting. You pretty much know all you're going to find out from having just listened to this from me. But they said: "A cross-disciplinary team of U.S. neuroscientists and cryptographers have developed a password/passkey system that removes the weakest link in any security system: the human user. It's ingenious. The system still requires that you enter a password, but at no point do you actually remember the password." Now, here again I'm quoting from ExtremeTech, where they don't quite understand...

Leo: They thought because it was 30 characters that it was somehow 30 characters.

Steve: Yes. And you're not entering a password. You are demonstrating a bias, sort of a statistical bias which only you have. So you first have to tell it who you are. Then it's able to challenge that assertion with sort of an additional step of authentication. So anyway, it's very cool.

Leo: Yeah. If you were really Leo Laporte, you would perform 10 percent, it's probably a small amount, 10 percent better at this than...

Steve: Yeah, they've got charts and graphs. And it's enough better that it's useful. You could imagine some really over-the-top, super-secret...

Leo: It is about 10 percent. This is interesting. 8.6 percent difference.

Steve: Yes.

Leo: It's that close. It's very interesting. And they used Mechanical Turk to do this. Did you know that?

Steve: Yeah, on Amazon.

Leo: On Amazon. That's really interesting. What a good way to get research subjects. Wow, I love it.

Steve: So a little sci-fi update.

Leo: By the way, in their bibliography, among other things, this is where the rubber hose comes from, an article on CNET. "Turkish police may have beaten encryption key out of TJ Maxx suspect." So maybe they named it appropriately.

Steve: Yeah, I'm afraid maybe they did.

Leo: Wow.

Steve: I wanted to give our listeners a bit of a sci-fi update. I did read those last two, the most recent two "Lost Fleet: Beyond the Frontier" novels. I mentioned after having only read - I think I was maybe two-thirds through the first one, which was "Dreadnaught," that it was really sort of disappointing me. I wanted to mention, just for those who had read the first six, that seven and eight - seven is a little slow, but eight really picks it up. And I'm not sure why I'm not as enthusiastic about it as I was the initial ones. I don't want to lead anyone one. But it was good. And so if you're looking for something to read, and you like the Lost Fleet series, seven and eight are useful. And "Invincible," which is the last one, was better. And I am now in the process of "Kill Decision," which is...

Leo: Ooh, how do you like that, yeah.

Steve: Yeah, I'm just starting. I'm back on my stair climber again. I'm working myself back into shape. I really had some substantial keto adaption period. I was having to drink a lot of bouillon in order to keep from just dehydrating and cramping.

Leo: You were feeling weak.

Steve: And I'm past that phase now. I've weaned myself from the bouillon, and I'm fine. So I thought, okay, now I'm going to start getting back into shape. While this was going on I didn't want to really tweak myself too hard with my workout, so I backed off on that. But so Daniel Suarez is - so far it's really interesting. I just like his writing. It's good stuff. It is not a trilogy going from the first two books.

Leo: It's unrelated.

Steve: Yeah, it's unrelated. Although, again, it's good cyber stuff. It's good tech. It's good military. So it's a nice read. I am enjoying it. And I did want to mention that Jenny and I saw "The Amazing Spider-Man" the other day. And it's, I thought, the best 3D that I've seen so far. I was more impressed with it, for some reason, even than "Avatar" and "Prometheus." I saw both in 3D, and I was not that blown away by them. But "The Amazing Spider-Man" I liked. I've liked all the Spider-Man movies, Tobey's movies. And this one was good, too. So if anyone is curious.

And I did get a nice note from a John Newcomb, who's a listener of ours. He said, "Dear Steve, just wanted to tell you how much I appreciate all the great info I get from your podcast with Leo Laporte. You guys are a great team. I recently bought a copy of SpinRite that I've added to my bag of tricks. I'm a computer tech and work for a company who services the dental industry. SpinRite has already saved me a lot of time. I was in an office the other day, working on a machine that would not fully boot Windows. I ran SpinRite, and it recovered several bad sectors, which then allowed me to image the drive and install a new one. Thank you for making my job so much easier. You software works so well. In using it and observing all the little details, I think it's clear how much care you put into its design and development. Sincerely, John Newcomb." So thank you, John. I appreciate the feedback and letting our listeners know.

Leo: Well done. You know, we don't have another ad, so let's get right to the questions. Are you ready, sir?

Steve: I be ready.

Leo: I be ready. But I'm not. So wait a minute, now, hold on.

Steve: Oh, I did want to mention, while you're getting it ready, Leo, that many people, many of our listeners have had tremendous physical benefits from experimenting with very low carb stuff. I had a bunch of tweets that I was considering sharing, but I thought, oh, I don't want to clutter up Security Now! with that. But remember that SGvlc is my Twitter feed. If anyone is interested to hear real listeners talk about many 10s, 20s, 30s, 50s, pounds of weight that they've lost, blood pressure normal for the first time in their adult life and all kinds of other good stuff, do a search on SGvlc and take a look at those.

Leo: Yeah, I kind of surprised my doctor today. He said, "Wow, you're looking good." So thank you very much.

Steve: You are, actually. I've noticed, too.

Leo: I feel good. Question No. 1 for Mr. Gibson from Brian Finn in New Hampshire: Hi, Steve. After thinking about all the precautions you've discussed in the past, and the best way to protect ourselves, it occurs to me that much protection is going to be wiped out by the switch to apps. Which is really ongoing, isn't it, not only on mobile, but also on the desktop.

Using the iPad negates LastPass. It's a nuisance, but doable in Safari. However, it's impossible to use LastPass in an app. While it may be possible to craft clever passwords, I believe the inevitable result will be to go back to our old habits, using the same password everywhere. And I have to say, I think he's probably right. Maybe I'm paranoid, but this looks like the next big thing, apps. What do you and Leo think?

Steve: Yeah. And, I mean, I'm having the same experience. It's fabulous in a browser mode, where you're logging into sites. I am so happy having LastPass there. But my iPad, that is probably the app environment where I spend most of my time when I'm off at Starbucks in the morning reading and poking around and doing research. You can do sort of - and I tried it for a while. Actually it was on the iPad 2 where I used some JavaScript snippets in order to invoke LastPass and try to - but it was, like, so awkward that I didn't even bother doing it when I switched to the iPad 3. It's like, okay, I just, you know.

And they have been moving, LastPass has been moving their own browser forward. The LastPass Tab, it's called, which is a nice iPad browser. So of course it brings LastPass to a web environment. But as you said, or as Brian said, we're generally more app-centric. One of the things that I find annoying, frankly, because I've already got so many apps, is that when I'm surfing with the iPad, now the sites I go to are all pumping their own app. It's like, oh, get the Slate app. Oh, get the...

Leo: I know, I hate that. I hate that.

Steve: I do, too. It's like, I don't want an app for a site.

Leo: No. I'm here in my browser. Give me the page.

Steve: Yes. And so sometimes you close it, and it comes back, or you change pages and it comes back again.

Leo: Oh, that's so frustrating.

Steve: It's like, oh, please. It's funny, too, because there was a comment in the mailbag about somebody - actually it was Jared, who sends me a lot of stuff - asking about spoofing user agent. And I was thinking, boy, I'd like to be spoofing my user agent because that's...

Leo: I'll spoof your user agent. What does that mean, "spoofing your user agent"?

Steve: Well, the user agent is part of the request header that browsers send out to tell the server what it is. It used to be in the old days they would disclose the screen resolution. Now they'll say I'm iOS and Safari on an iPad, blah blah blah. And so that's how these sites know that you're in mobile mode. And in fact, one of the things I find annoying is I'm unable to authenticate with PayPal when I'm mobile and I want to get something through eBay on PayPal because the mobile version of PayPal gets all tangled up because I've got multiple one-time password things.

I still have the football from the old days, and I've got the VIP, the VeriSign Identity Protection, running on my BlackBerry. So I've got mobile six-digit changing, time-base changing code. But that requires an extra step in authenticating because you've got to tell it which one of the devices you want to authenticate with, and the mobile version doesn't understand that. And once, for a while, they had a button to go to the non-mobile site. It's like, oh, thank you. That's gone now. They upgraded, and they took it away. So I'd like to be able to tell them, no, no, this is not - I'm not on an iPad. I'm on IE on Windows. And that's information that comes through the user agent, which is where the browser declares what it is. So it would be nice to be able to spoof that. And in fact, I think LastPass, the LastPass Tab browser does offer you that option.

Leo: Oh, that's nice. That's a separate browser for LastPass Pro users.

Steve: Yes, exactly. And worth taking a look at if you're really a committed LastPass person and you're an iPad user because it does give you access to LastPass, your LastPass database, synched through the cloud, secure as we know it is, with websites. But I still like - I just sort of like to use Safari. It's there.

Leo: Well, Apple kind of pushes you that way because you can't change the default browser. So when you click a link, you're going to get Safari. You know, Chrome has, and this is one of the nice reasons to download Chrome on iOS, it has a menu item, "Request Desktop Version." So you can, in Chrome, on iPad, and I think on iPhone, too, say give me the desktop. I don't want the mobile version, give me the desktop.

Steve: Wait. Download Chrome on your iPad as opposed to Safari?

Leo: It's available for the iPad.

Steve: Oh, Chrome, Chrome, Chrome.

Leo: Google Chrome, yeah.

Steve: Ahh.

Leo: So that is a nice feature of Google Chrome.

Steve: But Brian's right. I mean, as we are being pushed away from a web-centric mode in our mobile devices...

Leo: Well, we've got to push back. We can't let people - this is annoying.

Steve: Yeah.

Leo: For other reasons. I think a free and open web is important.

Steve: Yeah. And I like the idea, the model that Google has initiated and other people are following, of the web browser being your portal. There's a lot about that that works.

Leo: Yup, that's how it should be. But you can't control the advertising as well. That's the issue, of course. Jay Atkinson in Sydney, Australia, was left with a "coddling" question: Gentlemen, I have a problem. I find the information you provide intensely interesting - yes, that is a problem, but there's a cure. No, I'm sorry. I'm being facetious - and I often find myself wanting to talk through the content of the netcast to confirm that I understood what's been discussed. That's a good idea. Get a Security Now! buddy, listen together, and then afterwards say, okay, this is what I understood. And then he could say, well, this is what I understood.

My problem is following along with 357 I realized the very limited number of friends I have with whom I could discuss the netcast without them glazing over and sending them into an information-induced coma, so I'm very cautious to control my enthusiasm and choose my audience carefully when discussing anything Security Now! related. You just need better friends, Jay. Any mention of "Uncle Stevie Gibson" or Leo in my household already has my six year old rolling her eyes. She'll come around. But I know that this is the safe forum for posing such questions. So - I love this, Steve. I don't care what his question is. This is great.

Can you provide an explanation as to how they came up with the hard limits of "5ms in 100ms" in relation to the buffer coddling algorithm - oh, I'm glad he didn't ask me that - as on the face of it, it seems an arbitrary timing. Do you see any reason these timings would need to be reviewed in future? Thanks, gents. Leo, hope to catch up when you are down this way later in the year. I will, Jay, be in Sydney on November 7th, and I think Brisbane - is it Brisbane? - I think Brisbane on November 9th. Jay Atkinson (Magooligum), Sydney, Australia. Your response.

Steve: Yeah. So, great question. And I know, Jay, many of our listeners can sympathize. Relative to the 5ms in 100ms, we'll remember that what that was in the coddling algorithm was their criteria for whether or not to start dropping packets at an accelerated pace. If sometime during 100ms the delay through the buffer ever got down to 5ms, then the algorithm was happy with leaving things alone. But if at no time during 100ms was the delay through the buffer ever at or lower than 5ms, then the algorithm would worry that it needed to start sending some messages back to the protocols that were sending data through the buffer, that they needed to back themselves off a little bit. So this allowed bursts to occur, but not long-term overuse of the buffer. Those timings were arrived at empirically just through lots of modeling and testing.

But the 5ms in 100 actually comes from our real-world experience. It is an amount of time which, when multiplied by the number of hops packets need to traverse, still makes the Internet feel responsive. So the idea is we want both to be able to be downloading blocks of data, big video streams and things, where we really don't need interactivity. We don't need real-time round-trip interactivity. But we also want to be able to be playing games with shared servers, and we want to be able to click on a link and have the page immediately start loading and populating.

So we have differing priorities. But for us humans, if we take 5ms times a number of hops, we'll keep the sense that the Internet is working, that it's fast enough, that something hasn't broken. And that's where that came from is just, I mean, it's we as humans, and what do we ask? If it were all just big, non-interactive, huge streams of data moving autonomously, no one would care if buffers got deep and fat. That wouldn't be a problem. But we want the Internet to still feel reactive to us. And so that's the key is to keep those buffers small so that our intentions don't take long to get somewhere and the responses to get back.

Leo: So his question actually is legit because that might change when bandwidth situations change and so forth. Yes or no?

Steve: Well, no. And that's what's so cool about this, is this is bandwidth independent. This is based on time. So as bandwidth increased, then the buffers could be deeper, and they would automatically deepen, but the time factor would stay the same. And that's the genius of this is that it's based, not on bandwidth, and not on particular buffer packet size or length, but just on time. And time, the interactivity of our experience, is really what we're trying to preserve, and this does that.

Leo: Really interesting stuff. I love it when big minds solve problems and really think about this stuff.

Steve: And this thing is decades old, this problem. I mean, this is...

Leo: It's about time we fixed it.

Steve: This has confounded the best minds for quite a while. And then these guys finally said, okay, we really need to sit down and figure this out. And they did.

Leo: Two questions in a row here, so hang on. First from Carlos Alfaro in Rockaway Beach, New York, about iOS and Android security: Steve and Leo, love the show, been listening since Episode 1, own SpinRite, et cetera, et cetera. I know that both the iPhone and Android have their own implementation of sandboxing and memory protection to prevent applications from getting to the information they shouldn't have access to. But looking over the security - uh-oh. Let me make this a little smaller.

Steve: We've got a page boundary.

Leo: Yeah. Looking over the security - ah, here it is. It does this to me every time - model - and then it reformats. Okay, let's see. Let's view this as a single continuous scroll. Thank you. Just have to tell preview how to do it.

Looking over the security model for both platforms, I wonder whether they are really secure. An example of my concern would be using a banking application to access a bank or credit card institution. How would I know that another application is not getting my user name and password? If you read the permissions some applications ask for, it's out of control. Some apps "require" permission to read the state of the phone and see the number you're dialing. They can see your contact list. This is more so, I would say, on - you get these granular permissions on Android, not so much on iOS. We don't really know what they're asking for on iOS.

They can see your contact list, maybe even copy it. They can record sound, even phone conversations. A scary one is they can turn on the camera and take both pictures and video even when you're sleeping. Which used to be the case with IMDB on Android. The Internet Movie Database asked for permission so the application could turn on the camera and take pictures. They use the GPS to determine locations, coarse or fine setting, and others. Now, I know those permissions are not needed by some of the apps requesting them, and I know it does not mean that just because they have permission, they will use those permissions. I guess they look at all that information to monetize it? It's more complicated. I can explain why they have to ask for those permissions.

So finally, my question is, can I trust either Android or iOS to allow me secure access to my bank and other secure sources of information? Because I am not sure I have never used my Android phone for that purpose - I carry a USB stick with Ubuntu Linux all the time just for that - I've removed all applications that ask for permissions other that what I think is necessary for them to fulfill the purpose for which I installed them. Thanks again for looking at my question, and I apologize for such a long email. Carlos Alfaro, Rockaway Beach, New York.

And Brian Tannahill of Overland Park, Kansas, same kind of thing: I heard the news item a couple of weeks ago about Microsoft telling us to abandon and disable Desktop Gadgets and the Sidebar because they're insecure. Five years later. I'm amazed at how everything is riddled with security vulnerabilities, and I want to know, is it feasible to produce a reasonably secure operating system? Or is this nonsense going to continue forever? That's Brian Tannahill.

Steve: Okay. So, yeah. My sense is, and I'm sure you're seeing this and probably feel it, and our listeners I'm sure do, too, we're still as an industry trying to work these things out. You were just talking about, I think it's before we began, the new Gatekeeper in iOS X or Mac OS X...

Leo: Mountain Lion, yeah.

Steve: ...Mountain Lion, and how it's causing some problems for users who aren't used to it because it's additional protections that have been built in. The mobile platform is in general somewhat frightening. We've covered throughout the years problems with apps misbehaving, with permissions not being handled correctly, so on. So there's the problem of the platform itself not properly restricting controls, with mistakes being made there.

Then there's, as our first questioner asked, there's the problem of apps overly broad asking for permission.

My pet peeve that we've discussed a couple times is the way the authentication platform OAuth, where for example you, quote, "Sign in using your Facebook credentials," where you bounce over to another site and then come back, I'm annoyed that sometimes apps, all you want to do is allow them - or, for example, Twitter is another example where you want to sign in using your Twitter account. They'll say, oh, well, the site you're coming from, asking for authentication on Twitter, wants to be able to post to your Twitter stream. It's like, no. That's not what I want to give it permission for. But you have no control. You can't say, yes, authenticate me, but don't allow it to do those things. It's not an option on the screen.

So I do think it's going to take, as you said, Leo, it's going to take user pushback to some degree. It's going to take applications. The other thing that I see is sort of just an egocentricity on the part of developers who all feel that their app is God's gift to the platform, and their users are going to want to give it all these permissions so that it can do all these amazing things. There was a story the other day about how Google, I think it was a Google R&D project, would be looking out of your camera phone to sort of see where you were and what was doing on, and it would be able to recognize if you were at the beach and start feeding you beach ads. And, I mean, it would, like, know if there were baby strollers around and then give you ads relative to those. And it's just - some of these things get a little bit creepy feeling.

Leo: Yeah. And these guys have got to be aware of this. And I agree, more granularity. Part of the reason that apps sometimes ask for overly broad permissions is because that's how it's set up by the companies doing the APIs is that...

Steve: Right, those are the only things that are given.

Leo: ...that's your choice.

Steve: Yes.

Leo: Right. So one thing would be for Google and Apple to have more granular APIs where they could say, okay, you could do this. If you want to do this, ask for this. You don't have to ask for all of that and so forth.

Steve: And I guess then, of course, the back pressure on that, because arguably at some level, even if everything was done right, if everything was bug free, if there were no problems in the platform, if there were no malicious applications that were going to be abusing this, then you've still got the issue of users being annoyed by being asked anything. Some users are just like, don't ask me anything, just do, just go. Whereas there are a lot...

Leo: They're going to have to get over that.

Steve: Yes.

Leo: Sorry. You can't have both. You can't have apps that don't bug users and have privacy and security. You've got to have one or the other. And I think there's more of us who care about privacy and security than people who say, oh, just do it.

Steve: And then the other thing you would like is that, instead of an app saying you must agree to all these things or we won't work, you'd like - it'd be nice if the apps would say, here's what we want, and here's the features that are associated with those needs, and you can turn off the ones you don't want to provide and still use everything else.

Leo: Right. And I think that Android's moving in that direction. I think Google has made some noise saying, yeah, we want to give, not only more granularity, but give people the chance to turn stuff down and just the app just can't do that particular thing. But it doesn't mean you don't get the app. You just get a more limited app.

Steve: Well, I have to say that I am really impressed with the $200 Nexus 7 little tablet. I mean, it's a beautiful tablet. I was unimpressed with the Fire, but this is the first Android tablet that I thought, wow, this thing, I mean, it is nice. And I do like the idea, much as I am an iOS and iPad fanboy - I mean, I acknowledge it, I love what Apple has done. But we know from having seen so much history here that Apple needs some competition. And so I love the idea that Google with the Android platform could be establishing some standards of behavior that Apple, even though Apple has a reputation for ignoring everybody else, if there is push, then Apple will respond. So it would be good to...

Leo: Yeah, well, I think they are. I think the rumors are strong they're going to do a tablet of roughly this size.

Steve: Yeah, although I'm not a 16x9 person. I really like 4x3. I just don't think...

Leo: Well, that's what Apple's going to do. You're in luck.

Steve: I know. That's why I bring it up is it's 1024x768 in a 7" form factor.

Leo: We should say this is all rumor and speculation.

Steve: Yes.

Leo: Apple has not said anything that it's going to do. I don't mind 16x9 because it's a little thinner and taller, right, so it can slip into pockets and so forth.

Steve: Yeah, true. And it is more oriented toward media. And my bias is more toward reading.

Leo: Yeah. Well, but this is good for a book, too, isn't it? It's very much like the Kindle.

Steve: Pages, yeah, well, no, no, the Kindle is much more square.

Leo: Is it?

Steve: The Fire is also 16x9, and it's like reading in a long column.

Leo: Ah.

Steve: That doesn't work as well for me.

Leo: Really. Because this seems like roughly the page of a paperback would be. Maybe it's too tall.

Steve: Too tall.

Leo: Too tall for you, huh?

Steve: Yeah, well, compared to - 4x3 I think is a nice compromise because then you can still do media, you just lose some pixels on the top and bottom.

Leo: Yeah. I don't know. I think this is pretty doable. I think one thing that all of these have over traditional books is you can change the font size to make it just the right comfort level for you.

Steve: Oh, I'm an eBook reader. You were talking about my Palm Pilots that are in the refrigerator before we began recording.

Leo: You're never going to use those.

Steve: I was reading books on those.

Leo: But isn't this better than a Palm Pilot? Come on.

Steve: Oh, yeah. Oh, yeah.

Leo: I have to say I do believe that that transition is happening now. For a long time I thought, oh, no, paper books, my generation will never give them up. My generation is giving them up. I see eReaders everywhere now, everywhere I go.

Steve: Yup.

Leo: People love the convenience. They love the lightweight ability to carry to dozens of books. Magazines, too, I think that that's turned out - I don't read magazines. But for people who do, that's turned out to be a big part of this platform is reading eMagazines.

Steve: Oh, yes. Avoiding that throwaway paper makes so much sense.

Leo: Oh, I love that, yeah.

Steve: So much sense.

Leo: I'm so guilty, I'm a guilty New Yorker subscriber. I just want to read it. I don't need the paper. I don't need to make a coffee table item.

Steve: And I think there are people also who, like myself, we enjoy - I like holding the thing. I like the technology. I mean, I was jonesing for these things when Captain Kirk and Spock had them, and Uhura would bring the tablet over, and Kirk would sign off on the commissary or whatever it was.

Leo: They hadn't figured out that styluses are dead. Oh, yes, Uhura, let me just approve that. There we go.

Steve: He was always signing something.

Leo: He was. He was always signing a tablet. Brandon Hamilton, Rockford, Illinois writes: I'm worried about password security if the length is known. Oh, interesting. When it comes to brute-forcing a password, we know the longer the password, the longer it takes to crack it. But what happens if somebody knows the password is X number of characters long? For example, somebody wants to hack an account on a site that has a password requirement of at least eight characters. They know the password is at least eight characters long, so they don't bother brute-forcing five, six, seven-character passwords, reducing the amount of time needed to crack it.

The same holds true, for instance, if a person or entity learned a TrueCrypt password was 30 characters long. They wouldn't bother forcing all the possibilities from one to 29. Wouldn't that significantly reduce the time needed to brute-force a password? But how would one go about calculating this to determine if a password is still safe from centuries of brute-forcing, or if it would now be cracked in a few years? Thanks for the time you two put into producing a great show. I put no time into it. It's all Steve. What's the answer, Steve?

Steve: Ah, but you show up every week.

Leo: I show up roughly every week at 11:00.

Steve: So there's no mystery to this. And, in fact, one of the things that I appreciate is when I see applications and websites that deliberately hide the true length of your password. Sometimes they'll put up just a bunch of dots to sort of represent the fact that you need to enter your password, rather than leaving the field empty. But they deliberately don't even show the length because it is understood sort of implicitly that, oh, if you knew how long it was, that would give you a leg up on cracking it.

I immediately thought, when I was reading this, of the Password Haystacks page because it's all about the size of the dictionary and the length of the password. And what I did when calculating the length of time required to crack a password, and from that or before that the total number of combinations, is I did the math of the number of combinations with a single character, plus the number of combinations with two characters, plus the number of combinations with three characters and so on. So I did the sum of all the numbers of possible passwords of all shorter size, up to and including the one that the user entered. And because the theorem or the thesis that I was working on was that a true brute-forcer would start with A and then try B and then C, run through that and then go AA, AB, AC, AD and so forth, in order to go all the way up.

But Brandon's right. If you knew that a site said passwords must be 12 characters long minimum, then you would - no brute-forcer would try passwords of one through 11 characters. They would just start at 12.

Leo: The real question is how much time does that save really? Compared to what solving that 30-character password's going to take?

Steve: Right. And it actually turns out not to save much time because every character you add multiplies the length of time by the size of the character set, which is typically like 96, so it's 96 times 96 times 96. And so even if you know the minimum, you are still having to go from there up to whatever their maximum is. And Brandon, if you're curious, you can use the Haystacks page to easily see that. So, for example, put in a large character set 11-character password and look at the number of combinations that the Haystack page reports. Then type one more character and look at that, and subtract the first one from the second one. And that will give you the number just for that, for example, if you put in 11 characters and then added one to make it 12, well, subtract the value, the number calculated for 11, from the value calculated for 12, and you'll see that it's going to be, what, like on the order of 1 percent, something like that, a little more than 1 percent is what you're going to be losing.

Leo: The bottom line is it's such an astronomically large number to solve long passwords, we can give up a little bit, and it's still impossible. Not impossible. Time-consuming to the point of absurdity.

Scott Broschell in Ottawa, Canada wonders about - I like this - Elliptic Curve Crypto: I had a question I was hoping your expertise could help me with. I'm a software developer investigating some licensing software, and I came across the term "elliptic curve cryptography." Suspicious of any cryptographic algorithm I've never heard of, I did some research; and, somewhat surprisingly, what I read seemed to make sense on a security level as this cryptography works the same way as standard public key crypto with a hard math problem, often the integer factorization problem, but instead uses the discrete logarithm of elliptic curves. Ever heard of this? Is it secure?

Steve: Okay, so, yes.

Leo: Clever.

Steve: It is actually, in the crypto community, very well known. It is the currently most actively researched by the academic guys public crypto system that we have. What's interesting about it is that the computational complexity is lower, yet the apparent and well-tested and believed security is higher. So that's really important. For example, Microsoft's Phone has used it. BlackBerry has been using it for years.

Leo: Oh, interesting.

Steve: And it's a well-known technology. In fact, DNSCurve, which is the encrypted DNS technology, it's all ECC based, Elliptic Curve Crypto based. So to give you a sense, we were recently talking about how 1024-bit keys are being, well, they're still in active use. But, for example, they're regarded as a new RSA public key ought to be 2048. I had to, for example, use 2048-bit keys when I got my extended validation certificate. EV certs have to be 2048. And Microsoft, as our listeners may remember from a recent news blurb, is in, I think it's next month, they will be formally removing support for RSA crypto shorter than 1024. There are some 768-bit RSA things still around. They're removing that from XP SP3 and on in August. So 2048 bits is where we're moving to under RSA. But get this. 256-bit elliptic curve provides comparable security to 3072-bit RSA.

Leo: Oh, interesting.

Steve: Yeah. So only 256 bits of ECC public key provides about the same security as 1.5 times the bit length of 2048, which is 3072. And the reason Windows Phone and BlackBerry and other people are moving toward ECC is, again, those are lower powered systems, and they need the benefit of that greater gain. Essentially, you get more protection for less cycles with elliptic curve crypto than you do with RSA, partly because you're able to get more security with a shorter key, and key length is directly proportional to complexity, as we've talked about. When you go from 1024 to 2048, you get a large scaling of computational complexity, so you'd like to keep the public key as short as possible, yet not sacrifice security.

So ECC, it's well known, I mean, it's been around for many, many years. These very conservative crypto technologies take a while to happen. RSA's patents, for example, on the RSA technology, those patents expired in the year 2000, that was done so long ago. So RSA is well known. Elliptic curve is less well known, but it is present in the various crypto toolkits now. So it's available, and anyone can use it without concern. As far as we know, it's absolutely solid.

Leo: Sometime that might be a good subject for...

Steve: To go into detail?

Leo: Yeah, how that works.

Steve: Yeah, that's exactly how it works.

Leo: Little bit of math.

Steve: Yup.

Leo: Abhi Beckert in Cairns, Australia waxes thoughtful about browser session cookies: I'm a web developer, and I've noticed Chrome/Firefox remember cookies more than IE/Safari. I don't know what that means. But maybe you can explain. But my question is, how long should a session cookie last? If you're reading your email and have three tabs open in Safari, and if RAM is low, iOS and sometimes OS X will suspend Safari to disk, then relaunch Safari again with the same memory. If you restart your iPad or iMac, they are suspended to disk and re-opened as is. In OS X, Safari forgets the session cookies over a restart. It tells the kernel it's not capable of being closed/reopened in low RAM situations. I bet this will change soon, he says. But Chrome on a Mac and Safari on iOS can be terminated without losing cookies. I get it. So these are the session cookies that are, well, we've talked about them before. I'll let you explain.

Steve: Exactly.

Leo: In a world where an app cannot - [announcing dramatically] in a world where an app cannot be expected to be running at all times, perhaps that browser window is just a screenshot, and the app - I wouldn't be surprised, really - and the app has been quietly killed until you click it. When should session cookies be erased? And what about the reverse problem of users who never close their browser? I might leave a tab open on my iPad for two years, never closing it or the iPad. Should that cookie be revoked, or never be revoked?

My conclusion is servers should simply expire cookies after inactivity. Period. That is what I recently did when refactoring how our servers handle session cookies. Two hours without reading a cookie, it gets deleted from the server. Clients can keep them for years if they want to, but the server will give them a new one if they try use it. Abhi. Interesting question.

Steve: Yeah. Now, this follows from our discussion about my recent discovery that only IE honors what we have traditionally thought of as session cookies.

Leo: Right.

Steve: When you close IE - and that was following on from a user in a Q&A, like probably two back, saying, hey, Microsoft didn't impress me with their support when they said "Close IE, and that'll flush your existing login sessions." And it turns out that Firefox silently changed the behavior of what we've always thought of as session cookies. I mean, I guess I'm - I've been programming HTML for more than a decade, and I'm old school. Session cookie means session. It means it's never written to the disk. By definition it is never written to the disk.

Well, browsers kind of went off the reservation. They all started cheating except IE. Chrome preserves them. Safari preserves them. Firefox preserves them. Unless you, in the case of Firefox, as we talked about I think it was last week, go into about:config and manually override and push it back to its behavior before v4, back to good old v3.6 that many of us stayed with for a long time, was honoring session cookies. But what happened was so many web-centric systems are now using session cookies for their logon authentication, that users were being inconvenienced. And as we talked about, back when we were discussing this in Firefox, Leo, you'll remember Firefox tends to crash. And so when it restarts it was losing all of its logons, which was upsetting people more than having them be more sticky. So the behavior got changed.

So as it happens, I have come to the same conclusion that Abhi has. In my own use of cookies, for example when my employees are roaming and need to log into our backend database in order to perform customer service functions, the cookies that I'm continually providing them contain a timestamp. And if the server ever receives one that is aged more than an hour, in my case, because of the kind of work that they're doing, they're either doing things or they're not, if the server makes a request containing an expired cookie, then I require them to reauthenticate. I just tell them they have to log in again, essentially. And so as long as they're actively using the application, they get to stay there. If at any period of time they don't log out, but they stop using it, then after whatever reasonable amount of time the developer chooses, we just decide, okay, there's enough of a chance that somebody else has come along that you'd like them to reprove who they are. So we're going to be fighting authentication issues probably for the rest of our lives in this crazy Internet world. But I thought that this was a useful observation and comment from Abhi. I think the idea of allowing people to explicitly log out or logging them out - and essentially, session cookies are no longer being treated as what we used to think of them as, as session cookies.

Leo: That's what's really happened; right?

Steve: Yeah, it has. And so now it's up to the web developer on the server side to recognize that they could be getting a cookie from a year ago, a session cookie from a year ago, so they need to take responsibility for assuming that they are sticky.

Leo: A year ago. It's true, though, yeah.

Steve: It's funny, when he mentioned he might have a tab open in iOS, I'm thinking, yeah, I've got a couple stale tabs, yeah.

Leo: We never reboot anymore; right? That's all ancient history.

Steve: No, no. Yeah.

Leo: Bill Barnes, Charlotte, North Carolina wonders what's all that jive about Java? Steve, every other episode or more you talk about the evils of scripting. But you also mention "good" scripts and "bad" scripts, especially Java. So this has all left me confused. Could you enumerate in a single place what are the "good" scripts? You say remove Java unless I need it, but Java's okay because it runs in a safe sandbox. I know there are two computer products called "Java"; right? But I don't know how to tell the difference. If I uninstall the program in "Add & Remove Programs," will I still be able to edit my blog? Thanks, Bill. Oh, I'll let you do this one.

Steve: So, okay.

Leo: He's a little tangled in here.

Steve: Yeah. Nobody knows why Netscape named their client-side...

Leo: Oh, I know.

Steve: You do?

Leo: Yeah. But go ahead.

Steve: Well, nobody but Leo...

Leo: I'll tell you why. I know. They were literally - there's a whole story about it. But go ahead.

Steve: Okay. For some reason, which Leo will...

Leo: I'll enumerate later.

Steve: ...shortly explain, Netscape named a completely unrelated client-side web browser scripting language JavaScript, though it has nothing to do with Java, which is what Sun Computers' Scott McNealy famously named their interpreted object-oriented set-top box language when they were developing it. So what we have is two completely different entities. They run differently; they're treated differently; they're completely different languages; and, sadly, they both have "Java" in their name. One is Java. The other is JavaScript, though JavaScript is not a scripted version of Java. So there's the first issue of...

Leo: Problem No. 1.

Steve: ...deobfuscation, yes.

Leo: And by the way, it's no longer - did you know that it's no longer JavaScript? It's ECMAScript?

Steve: ECMAScript, yes.

Leo: Yes, ECMAScript. So there. So it was originally called Mocha, and then later it was called LiveScript. But coincidentally, when Netscape...

Steve: Wait, wait, wait. You mean in Netscape it was called Mocha?

Leo: The original code name was Mocha.

Steve: Okay, so there was a coffee flavor.

Leo: So there was a coffee flavor, but not an intentional confusion with Java. And then it was named LiveScript. That was the official name, I remember. A guy named Brendan Eich created it. But it was coincidentally at exactly the same time that Java was deployed in Netscape, in a joint release with Sun, that they announced that they were going to name its scripting language JavaScript. Some say, according, to Wikipedia, "The choice has been characterized by many as a marketing ploy by Netscape to give JavaScript the cachet of what was then the hot new web programming language. It's also been claimed" - now, that's what I believe, the former. But "It's also been claimed that the language's name is the result of a co-marketing deal between Netscape and Sun in exchange for Netscape bundling Sun's Java Runtime with its browser."

Steve: I believe it, too.

Leo: It happened at the same time. So everybody knew what they were doing, and it's confused people ever since. And I do believe that's why they've changed the name to ECMAScript. ECMA is a larger industry group.

Steve: Yeah. My feeling, my own reading on that is that it really is becoming a standard language. I mean, I like it.

Leo: You've become proficient in it.

Steve: Yeah, actually I've been coding a lot in the last few days. I'll be showing some people some stuff that I've been doing before long. And so my sense is it has outgrown the browser. And, for example, it's supported in Windows now, separately. So I'm really glad that it has become standardized; it is moving forward; it is maturing. And there's a lot of positive things to be said about it. Now, unfortunately, if Bill thought he was confused before...

Leo: We haven't helped. But it's a very important - and we say this a lot. Java does not equal JavaScript.

Steve: Right.

Leo: They're in no way related except for the fact that Sun and now Oracle owns both names.

Steve: So from a utility standpoint, one of the things that both languages do is allow cross-platform stuff. So all the browsers now support very much the same implementation of JavaScript. For some reason Windows and IE, still just dragging their heels behind, but they're catching up rapidly. So web servers don't have to care where they're serving the page to. Java is a plug-in. And that's a key concept. JavaScript is built into the browser. Java is a plug-in, kind of like a PDF reader is a plug-in for most browsers, although they're beginning to support PDFs natively also. But it's sort of like an add-on.

What Java provides is also cross-platform compatibility, where you can write the same code and run it on a Mac and on a Windows machine and under Linux and under Sun, different systems. However, one of the things that Java provides is much more power. You can do networking stuff. That famous buffer bloat packet delay meter application, you'll probably remember, the one that they designed that measures the amount of packet delay and buffer bloat that you've got in your network connection.

Leo: Netalyzr.

Steve: Netalyzr, yes, was written in Java because, I mean, it's very aggressive in what it's able to do because Java is a complete programming language, whereas JavaScript is deliberately constrained. For example, in JavaScript you cannot save something to your computer. They've deliberately kept it for doing browser visual things. But it's unable to write files to your PC. Java can do that.

And so while technically they're both sandboxed, inasmuch as they don't have lots of freedom, the Java plug-in sandbox is to keep its behavior constrained; yet, if the programmers want it to do aggressive things, like be a botnet, for example, it could completely do that. Whereas JavaScript doesn't have the ability to do those sorts of things. The Java plug-in could write files to your system, and often does; whereas JavaScript deliberately is constrained.

So to answer your question, Bill, could you uninstall the program in Add/Remove Programs? If Add/Remove Programs shows Java, then that's the plug-in, and you can definitely remove it and still be able to edit your blog, since your blog editing would certainly be done in JavaScript, using JavaScript, not the Java plug-in. And the Java plug-in has been a constant source of security problems in the past. Can you think of anything else I've left out, Leo?

Leo: No, I think you nailed it, baby. I'm sure you didn't help, or Bill has no idea what you're talking about. But that's okay. Sorry, Bill. Listen to this over and over again. Maybe the transcription, that will help.

Steve: Yeah, and unfortunately it is, I mean, this just demonstrates how...

Leo: It's intentionally confusing. They did it that way.

Steve: ...how messed up it is, yeah. And once upon a time it helped both of their interests.

Leo: It really was marketing.

Steve: And I don't think we're ever not going to call it JavaScript. I mean, that's a name that's going to stick.

Leo: Well, ECMA is the worst name ever. I mean, seriously, it sounds icky. It sounds like a skin disease. "I have ECMAScript." "Oh, I'm sorry. Maybe you can get a salve for that."

Steve: Yeah.

Leo: But I agree with you. And by the way, ECMAScript, I mean, we're going to get emails from people saying, well, technically ECMAScript is not JavaScript. Yeah, well, there's JavaScript, there's Jscript, there's ActionScript, there's a lot of these things. And ECMA I think going forward is going to be the standards body, and it's going to determine what it is, and that's frankly what we need as a unified standard.

Steve: Yes.

Leo: On the lighter side, Kevin Ghadyani in Overland Park, Kansas says: Mailbag? What is this mailbag of which you speak? If, as you and Leo stated, most of the people listening are school folk from assigned reading, wouldn't the term "mailbag" predate them by at least 10 years? He's saying it's like "dial the phone." When is the last time you saw a dial? Being older than most of them at age 25, I can say when you speak of the mailbag, I'm only familiar with one because of old TV shows and movies from when I was a kid, you know, when people used to actually hand-write and postal-mail letters to each other. I guess it's a bit like the way Leo is able to "read" a book with his eyes closed. So I'm curious, then: Why use the term "mailbag"? Kevin Ghadyani. By the way, he says, I'm aware the term "inbox" is also showing its age. Because of the methodology of email, this terminology still fits. I'm also aware I wrote this email in the form of a letter. Thank you, second-grade teacher.

You know, it's not just "mailbag." There's all sorts of anachronistic - "dial the phone" is one.

Steve: I used the term to a couple of Starbucks baristas who were in their teens. I used the term "asleep at the switch." And I thought, you know, they have no idea what that...

Leo: But they know what it means. They don't know where it comes from.

Steve: Right, right.

Leo: Look, have you ever said "the whole nine yards"? Do you know what that means? That's a complete anachronism, but everybody knows what it means. "I shot my wad." People think - no, it's not sexual. People think that's something - it's not. It has to do with hand-loaded muskets.

Steve: Yup, sticking gunpowder down and...

Leo: It goes back to the Revolutionary War. So I don't think it's unusual. I think that's how language evolves. And what's interesting is that in fact these anachronistic phrases, which have lost all meaning, at least in terms of...

Steve: Still maintain their context, or their...

Leo: Exactly. They've lost all context. They maintain meaning, yeah.

Steve: Right.

Leo: I think that that's fascinating. So I don't know about "mailbag." I don't know if that's one of them.

Steve: I remembered, when I was reading that, I immediately pictured a scene from "Andy of Mayberry." I think that's where I remember...

Leo: [Geezer voice] Hey, Opie, the mail's in.

Steve: And the bag slung over his shoulder. In his little gray mailman outfit.

Leo: So "the whole nine yards" of course means everything, right, the total thing. But no one actually is sure what the origin of that is.

Steve: I was going to say, Leo, I don't know where that comes from, "the whole nine yards." Because that's such a - it's not like the ninth hole. That's obviously a golfing reference, but I don't know if ninth hole means anything.

Leo: Some people attribute it to the length of a machine gun belt, nine yards long. So if you shot all nine yards of the machine gun belt - this is from World War II vintage aircraft - that that would be it. However, nobody said that until 40 years after the war ended, so it seems unlikely. The other argument is that nine yards is a cubic measure, refers to the volume of a cement mixer. But in fact that doesn't work, either.

Steve: Dump the whole nine yards.

Leo: Yeah, or the length of a bolt of cloth or a sari, the structure of certain sailing vessels, and American football, none of which make any sense at all.

Steve: So we're all saying it, and we have no idea.

Leo: No idea. And so that's what I think is going to happen to "mailbag." See, London Bob's in the chatroom saying, no, it's a naval term. Well, we don't know. Go read the Wikipedia article on "the whole nine yards," and you can see that no one really knows. They call it an "etymological mystery."

Steve: Ooh, nice.

Leo: "The most prominent etymological riddle of our time."

Steve: And you heard it here, folks.

Leo: And you heard it - but what's interesting, it wasn't used until the '60s.

Steve: Maybe someone just made it up. It doesn't really - it never has...

Leo: I think somebody like Dvorak made it up and said, "Heh heh. Let's see, heh, if they'll adopt it."

Steve: Exactly. Foisted on an unsuspecting public.

Leo: Now, you know "hoisted on your own petard," the Shakespearean line, a "petard" was an explosive device. And to by "hoisted by one's own petard" was to be blown up by one's own device. Now...

Steve: Hoisted.

Leo: Hoisted, because it lifted in the air. But I think, unfortunately, that is a term that has become an anachronism. I don't think I could say to a young person, "You have been hoisted upon your own petard," and they would have any idea.

Steve: No, I think at that point they would lock Grandpa up.

Leo: Grandpa, get off my lawn. One more, one more, one more, and then we'll wrap this thing up. From Scott Van't Land - I hope I'm saying that right, Van't Land - in Coalhurst, Alberta, Canada, a little password humor. We sure can use a little password humor.

Steve: Very little password humor, actually.

Leo: Hey, you included it.

Steve: I know.

Leo: [Geezer voice] In the mailbag. Steve, love the show, listen every week. In all the discussion about secure passwords, I saw this joke, thought you might find it amusing: A large company was doing a security audit of current passwords and found one that was unusually long. When they asked the employee why the password was "Mickey, Minnie, Goofy, Huey, Dewey, Louie, Donald, Sacramento." she explained, "Oh, I was told it had to be seven characters and a capital." That's pretty funny. Mickey, Minnie, Huey, Gooey, Dewey, Louie, Donald, and Sacramento.

Steve: Yeah. I have to...

Leo: It's good.

Steve: This lets everyone know that I read my Twitter feed. About three months ago I got just flooded with this. And I don't know, again, if it was XKCD or XCKD or whoever that guy is. I don't know where it appeared.

Leo: Love him, yeah.

Steve: But our listeners all thought this was really funny. Or at least thought I would think so. And I avoided mentioning it until now. And for some reason I thought, well, okay, I'm just in a mood today, so we'll share the seven characters. And actually, these don't strike me as Sleeping Beauty's seven dwarves.

Leo: No. There's Mickey Mouse and Minnie Mouse. There's Goofy.

Steve: And he wasn't one of the seven dwarves.

Leo: Huey, Dewey, and Louie are Donald Duck's nephews.

Steve: Well, and they were also the three robots on that wonderful...

Leo: Oh, you're right. Wow, I forgot about that. Yeah, with Bruce Dern, where he's the gardener in the - I forgot the name of that. I'm sure the chatroom will tell us in a moment.

Steve: The space gardener.

Leo: Yeah. And then Donald of course, is their - "Silent Running."

Steve: Yes, "Silent Running."

Leo: Thank you, [Name] in India. They're fast. And Sacramento, of course, the capital of the state of California.

Steve: Yeah, so seven characters and a capital.

Leo: Huey, Dewey, and Louie.

Steve: Actually pretty secure.

Leo: You know, it's a great password, especially if you capitalize, use commas.

That's excellent. Might put an ampersand in there just to really throw them.

Steve: Yeah, do @Sacramento, that'd be good.

Leo: @Sacramento.

Steve: There we go.

Leo: Steve Gibson's at GRC.com.

Steve: Actually, I think that probably pretty much describes the Senate that we have.

Leo: Huey, Dewey, and Louie? Manny, Mo, and Mike?

Steve: Goofy, Mickey, and Minnie?

Leo: Goofy? Now, we can't get political, Steve.

Steve: Oh, okay, okay.

Leo: But I share your sentiments. And actually I don't think there's anybody in the land who doesn't. We do this show every Wednesday, 11:00 a.m. Pacific, 2:00 p.m. Eastern time, 1800 UTC, on TWiT.tv. It's fun to watch live, if you want. But if you can't, man, we make it available in a lot of ways. And I'm not calling this a podcast, although it is. I'm saying "on demand," how about that? How about that for modern? On demand, as needed, when you want it, where you want it, in audio and video. In fact Steve, for the 16Kb folks, the folks who have limited bandwidth, makes a 16Kb version available at his site, GRC.com, and a text transcript, which is really lightweight. You can find that at GRC.com.

Steve: Wait, wait. What do you mean, "lightweight"?

Leo: Nothing lighter weight than text, 8-bit, 7-bit probably, ASCII.

Steve: Oh, and it compresses down.

Leo: Compresses, like, hmm, beautiful. How big - have you looked at that? I mean, really, after compression it's just a few kilobytes, probably.

Steve: Eh, it's nothing.

Leo: Nothing. Nothing.

Steve: You can read it with your eyes closed.

Leo: You can also get SpinRite, the world's best hard drive maintenance and recovery utility, there: GRC.com. And don't forget, if you've got a question, we do these every other episode, so you can put your questions on the form there at GRC.com/feedback. He's got information about sci-fi, about diets, about secure passwords. You mentioned the Password...

Steve: Haystack.

Leo: ...Haystack page. That's GRC.com/haystacks.htm.

Steve: Yup.

Leo: But it's all in the menu. It's easy to find. You can get audio, higher quality audio and video of the show as well at our site, TWiT.tv/sn. It is a podcast, so if you subscribe, you'll get it every week automatically in whatever form you choose. And never any charge, thanks to our fine sponsors. Next week do you know what we're going to do?

Steve: I don't, but I think we ought to follow this goofy episode up with something really seriously deep and propeller-winding.

Leo: Okay.

Steve: So I'm going to come up with a good one.

Leo: You've got it.

Steve: Okay.

Leo: Exciting. Thank you, Steve. Thanks, everybody, for joining us. We'll see you next time on Security Now!.

Steve: Thanks, Leo.

Copyright (c) 2012 by Steve Gibson and Leo Laporte. SOME RIGHTS RESERVED

This work is licensed for the good of the Internet Community under the Creative Commons License v2.5. See the following Web page for details: http://creativecommons.org/licenses/by-nc-sa/2.5/