SECURITY NOW!

Transcript of Episode #358

Listener Feedback #146

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-358.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-358-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson's here. He's got his review of "Prometheus"; some security news, of course; and we'll answer some questions from our audience, including a great tip for you LastPass users, something you might want to change. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 358, recorded June 20th, 2012: Your questions, Steve's answers, #146.

It's time for Security Now!, the show that protects yourself and your friends and your loved ones and your privacy online. Here he is, our Explainer in Chief, our Chief Privacy Officer, Security Officer...

Steve Gibson: CPO.

Leo: Yeah, CPO Steve Gibson of GRC.com. Hey, Steve. Good to see you again.

Steve: Actually, speaking of CPO, we do have a question that we're going to cover, since this is our 146th Q&A episode this week, a question from a listener about career opportunities in security. And I won't spoil my answer about that by talking about it any further, but...

Leo: Stay tuned.

Steve: ...we will be discussing that. I finally saw "Prometheus" on Monday, so we can talk about that a little bit. And I wanted to also notify our listeners who aren't following me on Twitter that the second season of "Falling Skies" has begun. But here I am, already starting, and I know that we've got a lot to cover.

Leo: I'll tell you what, let's do an ad, how about that. And we have questions and answers and all sorts of stuff.

Steve: And some errata. There was a mistake...

Leo: No.

Steve: ...that was conveyed last week I want to fix...

Leo: Never.

Steve: ...about that MySQL authentication bypass and the memcmp() function in C.

Leo: Oh. We went on and on about the memcmp() function, turns out that wasn't it. Okay, well...

Steve: No, it was. Well, kind of.

Leo: We'll explain.

Steve: We'll get there. We'll get there. I like to cram it all into one sentence.

Leo: And I know it's probably not in your lineup, but I've really got to know what you think of the Microsoft Surface at some point.

Steve: Okay. I know nothing about it. I haven't even - it's not even on my radar yet.

Leo: Oh, well, Microsoft made a big announcement about they're going to make a tablet called the "Surface." And I know you're into tablet computing. There'll be two form factors. There'll be an ARM form factor that uses Windows RT, basically the iPad-like version of Windows. And then there'll be one version that uses Windows, Windows 8, the professional version. And it's an interesting product. "Interesting" is the best I can say about it.

Steve: Yeah. I have my iPad. I think I'm done.

Leo: Yeah, I think that's the problem is you're not alone. Steve Gibson. "Prometheus." What did you think?

Steve: Yeah [sighing].

Leo: Yeah, me, too. That's exactly my reaction. Yeah [sighing].

Steve: Yeah. It could have been so much...

Leo: It was beautiful, wasn't - it could have been so good.

Steve: Yes, it could have been so much more.

Leo: No - now, we've got to say, we're not going to do any spoilers here, folks.

Steve: No, I'm not going to do any spoilers.

Leo: Because you should see it. Everybody should see it.

Steve: Yeah. And I've ordered the disc, the Blu-ray, back on June 3rd...

Leo: Yeah, it was gorgeous.

Steve: ...and I'm glad I have it because I will watch it multiple times in future years, as I have watched "Alien" and "Aliens" and "Alien vs. Predator" and both "Predator" movies. I guess there's a third one now.

Leo: First half is like "2001: A Space Odyssey." I'm thinking, this is going to be the greatest sci-fi movie ever. And the second half is more like "Alien."

Steve: Well, in explaining it to a friend of mine, I said, you know, it felt muddled. The beauty of the original "Alien" movie was that what it was about was crystal clear. You absolutely knew what was happening. And they just crammed too much random stuff into this. I mean, it was exciting and visual and interesting and sci-fi. I mean, it's got spaceships, and how can that be bad? But annoying things, like they just happened to fly over the landing site when it could have been anywhere on the planet, and it wasn't emitting any radiation or any signal or anything, so what are the chances of that? And then it just - anyway, I don't want to, as you said, we don't want to spoil it. But it got a 7 out of 10 on IMDB. I'll be glad to own it. But, boy, it felt like a real opportunity was lost.

Leo: There was such promise.

Steve: Huge.

Leo: And the trailer made it look so good.

Steve: Huge amount of money got spent.

Leo: Well, and it was gorgeous. You can't deny it was gorgeous. It was also gruesomely gory; so if you have any squeamishness about blood products, stay away.

Steve: And we had our new Sigourney.

Leo: She's good, Noomi Rapace. Is she not great?

Steve: She was wonderful.

Leo: She is a superstar. She was Salander, Lisbeth Salander in the original Swedish versions of "The Girl With the Dragon Tattoo" of the Millennium Saga.

Steve: Oh.

Leo: And she's, yeah, I didn't recognize her, I had to look her up, and I said, oh, her. She was great in that Swedish version. And she's a star. There's no question, she's a star.

Steve: Yeah. And also we were hoping for some clarification, and I think deliberately Scott added additional confusion, hoping that there will be a sequel. But all the reports I've had is that the theaters are not full. People are not rushing out to see this. So I don't know if it's going to make enough money to pull a second one.

Leo: I was shocked. I planned. I went opening night. I planned ahead. I bought tickets ahead. I had planned to be in line. No line. I got there 45 minutes early, sat in the front, there was no one in the theater. In fact, the kids in the theater kicked me out because they had to clean it. I went, oh, no, I'm going to lose my great seat. I went out, no problem, it was there when I came back. It's too bad. I really - I was ready for the science fiction adventure of a lifetime.

Steve: Well, we do have another "Spiderman" coming up, and we have a new Bond coming up, and we have another "Bourne" movie.

Leo: You forgot another "Batman" coming up. Sequel mania.

Steve: We'll have enough things this summer, I think. And speaking of this summer, I tweeted a couple days in advance of Sunday's premiere of the - it's a two-hour premiere, but it was just two one-hour episodes back to back of the second season of "Falling Skies" on TNT. And so I wanted to let people know that, if they missed it, it will be re-aired on this coming Sunday afternoon, June 24th, which ought to give our listeners time to listen to this podcast and queue it up.

"Falling Skies" is not to die for. Its lead actor is Noah Wyle, who of course made himself in the "ER" series that ran forever. It's okay. The first season starts after Earth has already been invaded and the planet pretty much decimated. And so we're the resistance humans fighting back against the aliens. There's some good computer graphics. It's interesting. It tries to be character-driven. Anyway, that gets a 6.9 out of 10 on IMDB, and I'll watch it.

The first two hours were good. It looks like they've got more budget. It did obviously get renewed for a second season, so that says something about it. So I would say, if people like sci-fi, you definitely need to see the first season. You can't really pick up here with Season 2 if you weren't watching it last year. But if you were, you probably want to continue. And I wanted to let everyone know that it has started again for the summer. As of course has "True Blood," but I'm a little disappointed in that. That seems to have lost its way.

Leo: [Laughing] I think we're just bored. It's fourth, fifth season, I guess, now.

Steve: It was so fresh and new and amazing...

Leo: Right, exactly.

Steve: ...the first year or two. Now it's like, okay, fine.

Leo: Eh, more vampires.

Steve: Yeah. So, errata. We gave the impression last week that the problem with that MySQL, what was called the "MySQL Authentication Bypass" may have been an error in the memcmp() function. My careful explanation made it very clear that that wasn't the case, but people came away feeling that we'd said that there was a bug there, not in MySQL. The bug is in MySQL. It's that the return value from the memcmp() function, the word in C is "coerced." It's being coerced from its integer value to a byte-size value. And that's why the...

Leo: Ah. So MySQL is ignoring the sign, not memcmp().

Steve: Well, it's ignoring the more significant bytes. And that's the key. The idea is, if you do a comparison, and you have a multi-byte comparison, you have to care about all of the bytes in the result, not just the least significant byte, because the other ones could be non-zero, and in fact while the least significant byte is zero, which will be the case one out of 256 of the times, which is what I explained.

But so anyway, what MySQL is doing is they're coercing the return value to a byte which discards the more significant bytes that are probably non-zero, if you don't have a password match. And so one out of 256 times you're going to get authenticated when you shouldn't. So I just wanted to correct the record to make sure people understood that it was not the memory compare function, which is just fine. It was the coders who for some reason coerced its return value into a byte.

Also people picked up on this 64-bit vulnerability, which I got a lot of people writing in and tweeting about that. First of all, it's already been fixed. What this was, was an incredibly subtle difference between the way AMD implements some deep voodoo hardware 64-bit stuff in their chipset versus the way Intel did it. Intel is where the vulnerability exists in the case that code was written to the AMD spec. Whereas Intel is saying, well, we're not going to do anything about this. Our chips are not buggy because our chips work the way our spec says. Except that what happened is people assumed the two chipsets, AMD and Intel, were identical in this really subtle way where they're not.

So it caught everybody off guard, but it's a local privilege escalation vulnerability that requires - it's like a theoretical problem. And all the OSes have already responded. It was already in last Tuesday's patch of this month by Microsoft. So I wanted to let everyone know. I'm not going to go into any great detail because it's impossible to explain without really digging into details of the way ring transitions work in the OS, and it's just a subtle difference that has already been resolved. So a tempest in a teapot at this point. And as far as we know, nobody ever - it's not being exploited. It never happened to anybody. It was just like, whoops, here's something that was found, and it's been fixed.

Unlike the Microsoft XML Core Services exploit that I talked about last week and urged everyone to follow the Fix it link. It is actively being exploited, and now its exploit code has been publicly released as a new module to the Metasploit exploit framework. So that's going to really ramp up its exploitation. If anyone didn't get around to going to the little Microsoft Fix it link, you can scroll back not very far in my Twitter stream because it'll still be up there near recent since I'm not tweeting a ton of stuff, and find it. You definitely want to do that because this is exploitable, not only through visiting a web page, but also targeted exploits through the earlier Office products, I think 2003 and 2008, if I remember, across platforms [2007 for PC]. So definitely something that you want to fix.

Firefox updated to - we're at v13 now. And we were at 13.0. We moved to .1 because there was a subtle problem that the latest version of Adobe's Flash had with an obscure conflict with something called RealPlayer Browser Record, which some people had installed in Firefox, and lord help them because you know how we feel about RealPlayer. I know Elaine, it's the only thing she can use for transcribing, so we understand, Elaine, you're still using it.

Leo: You're the one who's still using it, yeah.

Steve: Who needs it, who needs it, probably. Anyway, so Firefox. Adobe said you can move back to a prior version of Flash, but Firefox stepped up and fixed it. It was, again, a very subtle interaction with the sandboxing, the new Firefox sandboxing technology for

Flash which is hoping to contain future Flash problems and prevent them from turning into exploits. That interacted in some bizarre way with RealPlayer's Browser Record, and so Firefox has been updated to fix that.

I did just pick up a little note that I liked, and that was with the iOS 6 that was talked about and previewed in the recent Apple Developers Conference. The developer preview release notes mention that, under iOS 6, which will be the next major update, the OS will request explicit user permission when an application attempts to access contacts, calendars, reminders, and photos. Anyone who's using an iPhone now, or an iPad, any iOS-based device, has probably encountered the "can we have location" permission.

So at the moment it's only your location, your geographic location data, which Apple had been requiring you to specifically give applications permissions to have, yet we've talked about some of what people felt were privacy abuses where applications like, I think it was LinkedIn, was sending people's calendars off to the mothership. And they were saying, well, yeah, of course, because we want to allow you to look up the profiles of people who you're going to be meeting in the future. It's like, oh, okay. Would have been nice if you'd asked. So Apple is going to be enforcing that at the iOS level. So that'll be more user interaction, but I think that it's good, that it'll give people more control over what apps are able to do. So I think that's moving forward and is a good thing.

There was a bunch of news about the FBI making some noise, worrying about IPv6 because it sort of was as if they suddenly woke up to the fact...

Leo: Hey, what? You guys are what?

Steve: Well, now 4.3 billion IPs doesn't sound like many. We've got multiple terabytes of hard drive space, so 4.3 billion IPs? That's nothing. Now, with IPv6, of course, we're going to have 340 undecillion IPs, which is what we get with 128-bit addresses.

Leo: What's the problem? The FBI can't count that high? Oh, I'm going to be in trouble. I didn't say that. That was the Canadians. That wasn't...

Steve: What happens in practice is that it's up to ISPs to keep the whois database for the regions of the Internet they control up to date. So as ICANN and other disseminators of address space have been slowly and judiciously metering out the very limited IPv4 space, they've had the leverage to enforce ISPs to keep the "who are you giving these blocks that we're giving you to" records up to date. And as far back as when I was working with Verio, maybe 10 years ago, so more than a decade ago, I had to fill out a "how am I going to use the IPs you've given me," it was like an IP justification form, they called it.

Leo: Now they say, "Have all you want."

Steve: Well, that's the problem. Yes. If you've got 340 undecillion IPs, it's just, well, in fact, I am being offered, I was going to switch my...

Leo: They give you like a Class C, don't they?

Steve: No, no, a full Class A.

Leo: They give you the Class A.

Steve: Yeah. I'll have a full 32 bits of my own.

Leo: That's yours. Oh, my.

Steve: And so the problem is, ISPs will only ever be needing to come back to the trough for more maybe every 15 years or so, rather than constantly. And so now the FBI is saying, wait a minute, we have now realized that we depend upon being able to track back the owners of IPs because that's the traffic that we see. That's the Internet's addressing is the IP, the Internet Protocol address. And so it's going to be real problem if, with 128-bit IPs, they just sort of go off into a big block that isn't enumerated, which could very well happen because there really isn't anything other than existing protocol which has been inducing ISPs to do this, and the fact that they were, in order to get more IPs, they had to justify how they're being used now, and so that justification filtered down to the customer and then back up to the formal whois database.

So politically this is going to be - the FBI is saying, unless this is being done voluntarily, we may need some legislation to enforce this because...

Leo: Oh, no.

Steve: ...we need to know.

Leo: Oh, they're going to legislate against IPv6.

Steve: That's been said, yes.

Leo: See, people are so worried about, quote, "tracking" with cookies, tracking cookies. This is real tracking. This is law enforcement wanting to know exactly what you're doing, and they're upset that they can't. Okay. Enough said.

Steve: On the flipside, I understand, I mean, there are bad guys, and we would like the FBI to be able to go get the bad guys. We're seeing, for example, botnet operators and people who are doing denial of service attacks. They're being tracked down and caught. And that's good for the Internet.

Leo: But won't they know, if you get a Class C, or Class A, they'll know, I mean, they still have - you're in a range that you own, even if you're using - so they know who you are. This is not thinking.

Steve: Well, okay, no. But it's important to get this, that for my provider who is Level 3, or in the case of my T1s that I have here at home, Cogent, for Cogent to ask for more IPs, they need to say where the IPs they already received went. And if they've got just so bloody many of them, then they're only going to need to ask every 15 years or so. And so there just won't be - right now it's just protocol that induces ISPs to keep the whois database current because the providers of these very limited IPv4 IPs can say we need you to map out how you're using what you've already got before we're going to give you any more. Prove to us you need more. And so that's happening on a continual basis. Thus there is some political pressure for the ISPs to ask their customers how they're using the IPs and sort of...

Leo: This is the FBI saying we don't really want to have to go through really annoying court order subpoena process. We'd just like to have a database of what everybody's doing so we could just follow you around. This court order thing, it's just so slow. It slows us down. Who needs that? Because they could get that information. Right?

Steve: Yes.

Leo: Yeah, okay. It's just that they want to know ahead of time.

Steve: Yeah. They would have to generate a legal paperwork to induce the ISP to tell who they gave this block of IPs to.

Leo: Do the legal paperwork. Don't be lazy.

Steve: Many people picked up on the story of Fujitsu cracking 978-bit crypto. Well, so everyone's like, okay, is this bad? What does this mean? And a lot of people picked up on the report. It was supposed to take umpteen millions of years, and they did it in some 120-some hours. So this isn't a problem. Nothing about the crypto we're using today is affected. This is a completely different crypto technology known as pairing-based crypto. It is very much next-generation crypto. There are cryptographic libraries that implement it. It's still deep in academia. It's a cool technology which potentially solves the certificate authority trust problem, so it's got everybody interested. It offers something called "identity-based encryption," which allows you to use things about yourself as your key in a secure way. It's incredibly complicated. We'll talk about it someday, if it ever happens.

But there were theoretical beliefs about the strength of it. And so what Fujitsu did, and this is very good for it, was they showed it wasn't as strong as people thought. And a perfect analogy is the factorial problem. The reason, and we're talked about this before, the reason we need 2048-bit keys for asymmetric encryption is the difficulty we believe there is in factoring an integer that large. The reason we only need 256-bit or 128-bit symmetric keys is that it's an entirely different problem to crack it.

So cracking symmetric encryption is entirely different from cracking prime factor-based asymmetric encryption which requires factorization. So similarly, this pairing-based crypto is yet again an entirely different means of encrypting, and so it's got unknown key length requirements. We know what the key lengths are for symmetric. We've settled on what they should be for traditional asymmetric that requires factorization in order to crack it. Now we're looking at a third type. And so it's still in academia.

So what happened is there were assumptions about how long it would take to crack a 978-bit key. And that's a weird number all by itself. And it turns out Fujitsu used, like, hundreds of cores, operators had 200-some, 248 cores cranking away. And in what was a surprisingly short time for the academic researchers of this next-generation potential, next-generation technology, Fujitsu had a breakthrough.

So that's good. That means, oops, 978 bits is not enough. We'll just add some more. So we're determining the required strength for this key of pairing-based crypto. Interesting, but doesn't affect us in any way today. Someday we'll be talking about it, and I'll explain more about how it works because it offers some really, really interesting, cool things.

And then in the wacky story of the week, we have the news that an Australian online retailer, Kogan.com, has begun taxing people who purchase from their retail website, Kogan.com. They're putting an IE7 tax on their shoppers.

Leo: [Laughing] I love it. You mean if you use IE7, you pay more?

Steve: Yes. Yes. So this is just one of - 6.8 percent tax, which is 0.1 percent for every month since IE7's launch.

Leo: That is so wonderful. I want Newegg to do that. That's so great.

Steve: So the history of this is sort of fun. Chief executive Rusian Kogan told the BBC that he wants to recoup the time and costs which were involved in rendering his website into an antique browser.

Leo: It's the steampunk cost, the steampunk tax.

Steve: He says IE7 was launched in '06, and since then Microsoft has released two major updates to the software. And so the BBC reports that according to Mr. Kogan, the idea was born when the company started working on a major site relaunch. Kogan said that even though only 3 percent of his customers use that old version of the browser, his IT team had become preoccupied with making specific adaptations to make the pages display properly under IE7. And quoting him, he said, "I was constantly on the line to my web team. The amount of work and effort involved in making our website look normal under IE7 equaled the combined time of designing for Chrome, Safari, and Firefox."

Leo: Wow. That's actually legitimate, if it really is that hard. It should be a pre-IE8 tax because what of IE5.5 or 6? What about them?

Steve: And we have covered the news that even Microsoft is trying to get people to let go of IE6, and people won't because they've got corporate ware...

Leo: Right, Intranets, yeah.

Steve: ...that was written to it, and it won't work otherwise. So Kogan said it was unlikely that anyone would actually pay the charges. His goal is to encourage users to download a more up-to-date version of IE or use a different browser. And they had a screenshot in the story of where it showed on the receipt, on the web page, an IE7 tax. Whatever they sell, they're expensive stuff because it was a significant chunk of change if you insisted on using IE7 to do your purchase. So I got a big kick out of that.

Leo: Kind of amazing.

Steve: Yeah.

Leo: Bold. A bold, a new strategy. We should all do it.

Steve: It got a lot of good coverage, too. So this week and next week I'm going to amplify something that many people are reporting, and I'm pleased by it. And that is that SpinRite is becoming very effective, well, I mean, has been, but is becoming known to be effective in recovering solid-state drive technology. Cody reports from Lansing, Michigan. He said, "Good evening, Steve. I just wanted to share a quick note about" - oh, the subject was "SpinRite Saved My Raspberry Pi." And that may sound odd to people. We're not talking about something that comes out of the oven.

Leo: Oh, I think our audience knows what Raspberry Pi is.

Steve: Yeah. He said, "Good evening, Steve. I just wanted to share a quick note about SpinRite that I found interesting. After months of anticipation" - because as people may know, they immediately sold out of their first batch and had to get more. "After months of anticipation I received my Raspberry Pi, a $35 ARM-based computer. I rushed home after work and started tinkering. I have had a 16GB Class 10 SD card for a while with this specific application in mind. No matter how I wrote the image to the SD card, I was experiencing a wide range of issues when trying to boot my Pi. I gave up on the card finally and dusted off a smaller 4GB one, and everything worked. So I decided to see if SpinRite could even see the SD card." And by that he means he hooked it to his computer and booted SpinRite to see if SpinRite would pick up the card's presence because it would if it was supported by the BIOS. And in this case it was.

He said, "And it did. So I decided to run a Level 2 scan." And the only thing I would suggest, well, no, a Level 2 scan or a Level 1 scan, you want the quick read-only scan on solid-state memory. He says, "It took a few hours. Then I booted back into Windows, attempted to run the installer utility to write the image to the disk once again, which had continually failed. To my surprise and great pleasure, I was then able to boot the Raspberry Pi into the Debian Linux distribution made for it. I thought I would pass along this unlikely use of your product to help out any Pi owners who might be facing grief with their devices. Thanks. Cody, Lansing, Michigan."

So I just wanted to reinforce. We've mentioned it a couple times. But you don't want to use the Level 4 that performs a read/write, read/write exercise because that's needlessly fatiguing the limited write cycles on solid-state memory. But by all means, run a Level 2 because what that does is it forces the card's processor, whether it's an SD drive or a thumb drive or a SSD card of any sort, it forces the processor in there to verify that it's able to read the sectors, and it does so in a way which will induce the processor to either rewrite the sector if it's soft, or to relocate it if needs to move it. So I'm seeing multiple reports now, after having mentioned it on the podcast, of people who probably wouldn't have thought that SpinRite would help solid-state devices, and we're seeing that it does.

Leo: That's very interesting, yeah.

Steve: And I'll share another one next week.

Leo: So don't run it on Level 4. Run it on Level 2, is the tip.

Steve: Yeah, 1 or 2.

Leo: All right, Steve. Ready? You feel good?

Steve: Let's go through 10 questions.

Leo: You've got your special thinking cap on? All right. Here's Question #1. I need a thinking cap to say this name right. He's from Normal, Illinois, but his name is anything but. I apologize. I think I might get this right. Thanesh Rajandran in Normal, Illinois says, how do you research a topic, Steve? How do you do what you do so well? Long-time Security Now! listener here. Six months ago I was forced to pause my listening because it was my last semester as a grad student, and things were getting very hectic. Now I'm playing catch-up. I miss listening to your voice, in a good way, and the richness you share in every podcast. I love it.

It's true. Security Now!, one of our most popular shows, and there's a good reason. It's all that thinking and smartness coming off of Steve.

The question I have is sort of off topic from security, but I think that, nonetheless, many may find it useful. I know I would. When you go about researching a topic, as you have done with the health topics, Vitamin D or ketosis, or with security topics like WPS, you always wind up being able to discuss the topics from the ground up, starting with the fundamentals. It's really amazing. So I'm curious. Do you have a process? When you begin the task of researching a topic, are there certain keywords you use when you Google search? Are there default websites you visit initially and branch out from there? I don't have a SpinRite story to share yet, but I have my hopes up since my seven-year-old laptop is starting to get a little cranky. Hey, that's nothing to hope for.

Steve: I was just going to say.

Leo: I hope I get to use it.

Steve: Okay, but I don't think that's a good thing.

Leo: Thank you, Thanesh.

Steve: Yes. Remember that it does - SpinRite is useful for prevention.

Leo: Yeah, in fact, now's the time to use it.

Steve: Exactly. Something's a little cranky, I'd run SpinRite, frankly, just to see, like to get some reading on what's going on. But to answer your question, what is probably not maybe obvious is just the sheer number of hours that I spend on these things.

Leo: Oh, here it comes. He's going to want a raise now. All right...

Steve: No, no. But when you're talking about the Security Now! podcast being strong, you know, Leo...

Leo: You work hard.

Steve: ...from just looking at the materials that I prepare...

Leo: You work very hard, yeah.

Steve: Yeah, I mean, I really do invest in this podcast. And similarly, I invest in the things - in my own passions. And so before I surfaced with a podcast about, for example, the "Over the Sugar Hill" stuff, I'd read six texts, I mean, six books. And I have a biochemistry text and a ton of research. So I would say, first off, it's that this is not something I do for a few hours. This is something I do, in the case of low-carb stuff, for six months before anyone else is really aware of it. So it's where my life goes for a long time. And I think the key for me to explaining these things is just understanding them.

Leo: Yes.

Steve: I'm never afraid to say "I don't know." And one of the problems I think that techie people maybe have more than others, I'm just sort of aware of it, is because their identity might be a little caught up in being smart and being, oh, he's the computer guy in the family, it's often hard for people to say "I don't know." Like they want to be the know-it-all, the wizard, that always has an answer.

Leo: It's a good thing. You can say that.

Steve: Yes, because the problem is, if you don't acknowledge you don't know, then you're less inclined to go find out. And so I recognize when it's like, I read something, and it's like, okay, wait a minute, rather than on any level needing to believe that I understand it, if I really don't, I go find out. And so I think that's the only thing I really have is a clarity of what I don't know, which then drives me to find out, and then I'm able to explain it in a way that I think is clear.

Leo: Albert Einstein said, "If you can't explain something simply, you don't understand it well enough."

Steve: Yup.

Leo: I'd also point out I just read some research that was fascinating, that smart people are oftentimes less logical and more prone to making errors and mistakes because they believe in their ability so much.

Steve: Yup. I saw that, too.

Leo: Wasn't that fascinating? They don't second-guess themselves. They don't think hard enough. So that's a challenge that I'm sure people like you have to really constantly give yourself is, wait a minute, I think I understand this. I probably should look into it.

Steve: Well, I had a neat techie employee who worked with me on the R&D side of SpinRite 20 years ago, back in the SpinRite 2 and 3 era. And he was young. And when there would be like a mystery, a bug, he would leap forward and guess. He couldn't resist guessing. And he was, he was GATES. G-A-T-E-S is an acronym for some sort of, like, a young version of MENSA.

Leo: Gifted and talented, yeah, those programs are all over the country, and they're really good programs, the GATE programs. But...

Steve: Yeah, the problem was he - his name was Jim. And I said, well, now, here's the problem, Jim, is you now have an ego stake in being right. We don't know if your guess is correct, but now you want it to be. And I said, I have no problems not guessing, saying I don't know what's wrong. And I believe that makes me more open to seeing what is actually going on, rather than sort of trying to bias that result in my direction. So I just - I don't want to have a stake. It's why, for example, I've invested substantially already in the design of this ketosis measuring device.

Leo: Ho ho ho, he's building it, ladies and gentlemen.

Steve: Yeah.

Leo: You put that on your head? What the hell?

Steve: You blow in there.

Leo: For those listening, it is breadboarded. Lots of - I see six, five pots on there. I see a bunch of circuitry, wires. And there is a little - there's a USB interface. Oh, that's cool, so it's USB. And a thing, I see a thing, I guess that you could buy that off the shelf, that you blow into.

Steve: Oh, no, that's a made-from-scratch...

Leo: You made that, too.

Steve: It's a chamber that has the sensor located inside...

Leo: Dude, you rock.

Steve: ...which is measuring - anyway. So the point is...

Leo: You're calling that the Ketoflute, right, as I remember.

Steve: The Ketoflute, yeah. And I've invested heavily. I've had friends blowing in it. I mean, it's working. But I still don't know...

Leo: Hey, come here. Can I get you to blow into this?

Steve: I still don't know if I'm going to have anything.

Leo: Right, this is research.

Steve: Yeah, it's pure R&D.

Leo: Steve, this is why we love you. I'm getting tears in my eyes. This is amazing.

Steve: I need to answer the question. Maybe it'll work; maybe not. But again, if I have to go, well, okay, now I know, at least I've satisfied my curiosity without needing it to be something it isn't. So...

Leo: You see?

Steve: It's important to be a pure researcher.

Leo: And everybody needs to have that. If people could develop that mentality, it'd be so great. If you don't know, investigate. And you probably don't know. That's the problem. Smart people think they know.

Steve: Well, and that was the lesson from the Portable Dog Killer. I just encourage people to go build something because, oh, the act of doing that teaches you so much.

Leo: Right. Question 2 from an anonymous listener: Steve, with all the recent security breaches of super-popular, multiuser sites like LinkedIn, what steps should a company take when such a breach occurs? Do they disable user logins and password changing till they verify that the original hole has been plugged? Should they force users to change their password on first login? Should the company randomize the passwords, send users an email with a new password? I realize the security hole in that email, but given that the leaked password could be unhashed within hours, changing the passwords would essentially make the leak useless. Thanks again for a great podcast. What is the right response?

Steve: Well, I loved the question because it's the right one to ask. And we can answer it by stepping back a little bit and asking what is it which has been lost, which such a company needs to regain? And what's been lost is their confidence in their ability to authenticate existing users. So they learn with as much surprise as the rest of their customers or users that they've been breached, that somehow out posted on the Internet are all the hashes of their users' logins. So certainly they definitely need to solve the problem of the breach. Otherwise, anything they do can just get releaked. So absolutely they need to, as quickly as they can, figure out what it is that happened, how it happened, how that data got loose.

So there's the forensic side they need. They also have to, exactly as our anonymous listener suggests, then be, A, 100 percent skeptical of any subsequent login and even, depending upon the nature of their service, logins which could have occurred from the time of the breach until discovery. That is, in the case of that, I want to say form.me. I can't remember what the site was. There was LinkedIn, eHarmony, and there was another one [Last.fm], where apparently their database had been floating around for years on the Internet.

Leo: Oh, yeah, yeah.

Steve: So we have this issue of when did the breach occur, which is different from when we found out about it. From the moment that it occurred, anything that had happened since then is suspect. And so certainly the company needs to factor in what services they're offering, what kinds of things they do. This is different if it's people posting random comments to random blogs versus PayPal or Visa and MasterCard, for example, where there could be dramatically more consequences to failure of their authentication mechanism. So there's when did it happen, when did we find out, what are the consequences of authentication compromise then, then solving the problem so that any changes won't continue to leak out.

But then, and here's really where we come down to what action they should take, is how do they reestablish authentication? And, sadly, the way the world is right now, it is the relatively insecure email loop, as we call it, is probably the only way to do this. Now, the best practice is not to simply blindly mail updated links to everyone because that's just - you would, first of all, that creates a huge burden. Suddenly your authentication ability goes to your entire user base.

What probably makes more sense is for the company to deny all subsequent logons with the existing compromise password, that is, set flags throughout everyone's account saying that you cannot log on right now. When new visitors come, they get an explanation, which they should have had emailed to them. So certainly the company wants to be responsible and immediately send out email explaining that there has been a breach that they're pursuing to figure out how bad this is, what the consequences are, so forth.

But then the idea is users who want to log on should request at that time that they be emailed updated credentials to the existing password. They can't obviously use - I'm sorry, to the existing email account. They obviously can't use a new email account. Otherwise the bad guy could attempt to logon as them and immediately commandeer. So it has to be the email account that was used, which they need to prove continuing ownership of. That email contains a link which they click which then authenticates that they're the recipient of email that was sent on demand. Then they're allowed to change their password under hopefully stronger hash and strong password management technology, and reauthenticate.

So it's a tricky situation. And what you would hope is that state-of-the-art companies already had that in place, that is, that they're not scurrying around waiting for several weeks to be able to respond affirmatively. It would be nice if, even though they never want their password databases to get compromised, if they had planned for it, and it was built in so that they were able to initiate this kind of process on demand. And they probably have it in place for the instances already where, on an individual basis, this needs to be done. So this is why in general it doesn't take forever to get this to happen is unfortunately they just have to do it on a mass basis rather than individual users saying, hey, I need to do password recovery.

Leo: It was anonymous, and there's some speculation that maybe this was the CEO of LinkedIn that was asking, but I don't think that's probably the case.

Steve: No. Hopefully he's got IT people. Although, given that they used an unsalted MD5...

Leo: Maybe he doesn't have any IT people anymore.

Steve: Yeah. I'm happy to answer your question.

Leo: Thank you, Reed. Jared in Australia wonders about testing connection bandwidth: When you use websites to test your connection speed, for instance Speedtest.net, and press "Begin Test," it finds the fastest server for testing geographically, but that's not always the closest. However, after the test finishes and the results are shown, why in general is the upload speed always the same, even when the test is run multiple times for accuracy, whereas the download speed is often varying around? For instance, on my DSL I get 9.4Mb download on average, but sometimes it's 9.35, sometimes it's higher, but it's usually between 9.35 and 9.40. The difference in the upload speed is much smaller, 0.86 to 0.87Mb. So why, Steve? Can you explain this?

Steve: Okay. So he wondered why there's the huge difference.

Leo: Why the variation in download but not upload.

Steve: Yeah. Now, actually there's less variation in download than upload when you look at it as a percentage rather than as an absolute, which is probably the proper way to look at it. He sees between 9.4 and 9.35Mb download, but that's only a difference in half of the second significant digit, from 9.35 to 9.4, whereas he reports that in uploading it's 0.86 to 0.87, which is a difference of one whole second digit. So even though it's counterintuitive, there's actually more percentage variation in what he's seeing in upload than in download.

But in general, I liked the question because everyone wants to know what their bandwidth is, typically, in downloading things. And as we've looked at the Internet, we understand that it is based on a best effort packet delivery, where routers that are overloaded have official sanction to simply discard packets, which will stall the traffic and require them to be resent. We also know that most downloads are over TCP, which requires that the packets come in, ultimately arrive in sequence. So if one packet is lost and other packets are sent, then the lost one needs to get retransmitted, and that generally means retransmitting everything from the lost one back.

The point is that there will, just naturally, this is not like in the modem days where we had wires connecting the two points. Instead, we're sort of spraying packets out into the ether, and they're wandering their way toward their destination. So the proper means of measuring bandwidth is to simply use the largest number you can ever get. That will be...

Leo: That's your max.

Steve: ...your actual bandwidth. Anything less is a consequence of some momentary loss along the way. So run the test 10 times and see what the best is you can get.

Leo: Don't average it, look at your peak.

Steve: Right, exactly.

Leo: Yeah, that makes sense.

Steve: That's the best stable number.

Leo: Question 4 from Ireland. Bart B. in Maynooth, Ireland suggests a method for determining a user's DNS server without JavaScript. Remember in 356 Steve talked about a few ways Google might be able to know what your DNS server is. Google was warning people when they had DNSChanger that they had malware. He says: I think it would be easy to do in the following way, entirely server side, without any reliance on JavaScript. What you need is for your web server - Google - to communicate with an authoritative DNS server under your control. You set up a DNS zone for the purpose of testing DNS servers, and you run that DNS server. You have that server log the IP requests to it for a given subdomain. Logging to a database with rsyslog would do it for you without even needing to hack your DNS server.

Your web server inserts a request for a JPEG - this is very complicated. I hope I'm following this, and I maybe reading this wrong. But your web server inserts a request for a JPEG image - when I say "you," I mean Google, or whoever's trying to do this. Google's web server inserts a request for a JPEG image on a one-time subdomain of their controlled test domain into the website it returns to the client. This will cause the browser to resolve that domain using the visitor's DNS server. Since the subdomain is a one-off, it can't be cached anywhere, so that DNS server will have to request the relevant A record from the authoritative server, in other words, that server we set up, the DNS server. The web server can then check the DNS server's logs to see where the request for the one-time domain came from, hence telling the web server what the visitor's DNS server is. This would work if you got one request a minute, maybe.

No JavaScript needed, and even a moderately skilled sysadmin could hack such a system together in a day. If I had to do it, I'd send up a BIND DNS server on CentOS 6 logging into MySQL, using the default rsyslog syslog implementation that comes with RHEL (Red Hat Enterprise Level) CentOS. I'd then create a simple CGI script running in Apache to query the MySQL server for the relevant logs. Show off.

Thanks for the show and for SpinRite. Keep up the great work. There is now more than one episode of Security Now! for every day of the year - that's true - quite an achievement for a show where each episode is at least an hour long these days. Hat's off to you guys. Bart.

Steve: So I wanted to acknowledge Bart's note and the similar notes from many listeners.

Leo: It's an interesting mindbender. How would you do this?

Steve: Well, yes. And I've done it because everyone will remember the spoofability test. I wrote a pseudo DNS server, of course in my case in assembly language, which does just that. When you use the spoofability test, it sends a bunch of uniquely named images to your browser, causing your browser to ask its DNS servers for the IP, which causes them to ask GRC's authoritative server for the IP, and as a consequence...

Leo: Oh, clever.

Steve: ...I get the queries from the users' DNS servers that allows me to determine how spoofable they are.

Leo: Don't you, I mean, isn't there an issue, though, with timing? I mean, you have to say, well, I mean, if you were Google, and you had a million requests a second, this is going to be a very difficult thing to do.

Steve: Yeah. Exactly. It can work. I love the fact that so many of our listeners are on the ball to this degree that they've picked up on it. I've implemented a system, as I said, just like it for the spoofability test, so it certainly does work. And so I wanted to acknowledge everybody who said, hey, Steve, you could do it this way, too.

Leo: Have to be a low-traffic site, though. I mean, if you got 20 requests in a few seconds, it's going to be hard to say, well, that one came from here, and that one came from here; right?

Steve: Ah, no. The way you do it, though, is, for example, you ask for zqrd7 blah blah blah dot...

Leo: So you have a unique JPEG for each time.

Steve: Unique domain name, and you tie that to the user sessions, exactly.

Leo: You're just generating unique domain names. That makes sense. So that way you can just say, well, what domain name asked - who asked for this domain?

Steve: To whom did we send this query.

Leo: Right, that makes sense.

Steve: Yeah.

Leo: So then you could handle it, I presume, millions of times a second. I wonder how they do it now? Do you think they do it that way?

Steve: I wouldn't be at all surprised. I mean, that's a nice, solid, robust way of doing it.

And GRC does lots of spoofability tests. I can handle huge numbers of people doing that, yeah.

Leo: Okay. Jessica Tallon, Lancaster, U.K., notes LastPass Password Iterations: [Bad accent] Hello, Steve. Sorry, Jessica, that's mean. Hello, hello, hello. Mary Poppins. I religiously listen to Security Now! and would first and foremost like to thank you on the fantastic job you and Leo do, not to mention the wonderful "Over the Sugar Hill" episodes. I remember a while back you mentioning Last Pass would soon start offering password iterations. You said they were in the implementation phase; however, I don't recall you following up on that.

I was poking around my LastPass settings today and noticed, when you log into your vault and check the Settings option along the left-hand side in the General first tab, you notice a new entry, Password Iterations. By default, at least for existing users, it seems to be set to 1, which of course means no added security. But it does recommend cranking it up to 500. I cranked mine up to 1,000. That's the highest it recommends. I thought it should be something worth noting as it isn't automatic, and the user must take action to ensure they get this added security. Thanks again, Jessica Tallon.

Steve: Yes, I wanted to pass that on. It's a very good point. Anyone signing up for new a LastPass account will have their default set to 500.

Leo: Ah.

Steve: But any of us who have been using LastPass since I first did the careful comprehensive podcast about it, or anyone who knew of it before then, will have, as I do, or did, rather, it's still set to 1. So it needs you, the LastPass user, to go there and crank it up. They allow the setting up to some huge number, 100,000 rounds. But the problem is, all devices that you use LastPass from would have to be fast enough to do that, and things like iPads or iPhones or something may not. So they recommend not going - you could go to a thousand, if you wanted to.

But they're already using SHA-256, which is a deliberately slower hashing algorithm than SHA-1. So the reason you do these multiple rounds is to slow down any brute-force attacks. So they recommend 500. Our listeners probably want to go a thousand. And the only consequence is a slight delay when you're authenticating, just that one time that you're providing your password. In your browser it's got to crank doing this crypto X number of iterations where you decide what X is. Then it sends the final hash to LastPass. So I just wanted to - I thought it was a great note to let everyone go look in your vault, click on Options or Settings, and right there you'll see, in red, they've got - it's highlighted for me. It was in red because mine was still set to 1.

Leo: Yeah, so was mine, yeah.

Steve: Of course, yeah, I cranked it up.

Leo: Yeah, so...

Steve: So thank you very much, Jessica.

Leo: Yeah. And you think 500's adequate?

Steve: Yeah, I really do. I'm sure those guys recognize - they used a slow algorithm and so forth.

Leo: So this is to make your password, your login to LastPass password more secure.

Steve: More resistant to brute-force attack.

Leo: It slows them down.

Steve: Exactly.

Leo: Okay, good. I'm doing that right now. And since it is both client and server side, as you said, you want to make sure that you don't set it too high because your iPhone could take a long time to log in.

Steve: Because, yeah, all of this, all of the hashing is done on the client. So some clients are slow.

Leo: Right, although I think an iPhone's pretty fast these days.

Steve: I think it probably is.

Leo: Mike Calmus in Washington, D.C. wonders about Microsoft's "you're doing it wrong" attitude toward security: Steve, I submitted the following problem report to Microsoft regarding Internet Explorer and got the subsequent response. I wonder if you could comment on this attitude industry-wide and possibly help get Microsoft to do something about this issue. Thanks for Security Now!. Love the show.

My message to Microsoft: "When client certificate authentication is used in a website in Internet Explorer, particularly with a Common Access Card (CAC), IE caches the certificate information such that the browser must be completely closed, all windows and tabs, for another user to use the system. If the browser is not restarted, the original user's certificate will be presented to the website, and that user's information may be then disclosed. This occurs even after an Access Card has been removed from the system. This occurs in all versions of Internet Explorer that we've tried."

Their Response: "Hello, Mike. Thanks for your message. Completely closing out all browsing sessions is considered a best practice for ensuring that any information present in the browsing session is completely removed, including authentication cookies and SessionIDs." Then they point to a discussion on MSDN. "Regards, Nate."

Steve: I'm no apologist, lord knows, for Microsoft. But I think - I thought about this for a while. And I don't disagree, frankly, with Microsoft's statement. The problem is really that we're asking for something from our browsers that they're just not very good at. I mean, this is similar to the advice to log out, manually log out of a website that you're persistently logged into. Otherwise, as we all know, somebody else could come along, and the browser doesn't know it's not still you. So the problem of the common access card not being pulled constantly is an efficiency and performance tradeoff.

I guess my feeling is, if you're needing to use your browser in a mode where you really need user authentication security, then perhaps enforcing the use of the private browsing option that all of our browsers now have where, where when you terminate private browsing, it expressly isn't saving anything persistently on the machine, I think that's probably something that the server serving the pages could and should detect and verify. And unfortunately, it is the user's responsibility to revoke authentication because otherwise our browser-based authentication is sticky for the sake of convenience. So it's one of those security versus convenience tradeoffs.

Leo: Yeah. All right. So I think that that makes sense. The choice they've made makes sense, is your point. You could argue about it. But I think it's not just for these CAC certificates. There's lots of reasons why you need to restart your session. The problem is, in Windows, well, no, I guess Windows actually does it better than Mac. On the Mac you can close all windows, and the app will still be running. On Windows, when you close the last window, the app closes. So presumably, if you've closed the last window, you now have restarted the browser; right?

Steve: Right.

Leo: So actually that's another reason why what they do is probably the right thing to do. There has to be a window still open for the browser to be running.

Steve: Yup.

Leo: Jody in Florida wanted some clarification about there's two kinds of salt? Well, yeah, you have your sea salt and - no, not that kind. Steve, in your discussion about salted hashes in Episode 357, you grabbed my attention when you seemed to distinguish between a generic and a per-user salt. I'm familiar with the situation where each user has their own unique, random salt value and assumed that's what's always done to implemented salted hashes. Are you saying sometimes there's only one? Are you saying, Steve, sometimes there's only one salt value system-wide, used to compute hashes for an entire user database? If so, then your opinion that the salt should not be stored with the hash values makes a lot more sense. In the case of per-user salt values, it always seemed to me storing the salt values on a separate server from the hashes isn't going to provide significantly more security because it's already one per user. I hope I'm not the only one that could use some clarification on this. Thanks, Jody.

Steve: So, Jody, yes. One thing to remember is that many of these database exploits are obscure SQL database table traversals which we talked about years ago where you inject some cross-site scripting and explore what the table names are and end up dumping it out, like through the browser, actually, and acquire a database that was never expected or supposed to be exposed publicly through that channel. But what that means is that the bad guys haven't actually penetrated your network. They're not roaming around from machine to machine. And it's not like you got infected by Flame or some serious spyware. Certainly that can happen, and could. But typically they've just gotten the sort of a shoehorn in that lets them get the database, which means they have no access, for example, to a system-wide hash which is not stored with the database. If you stored the hash, the per-user hash along with the hash, and they get the database, then they get both of those things.

So that's why last week I explained why really best practice is to do both, have a system-wide hash not in the database but just sort of like in the code somewhere, which could escape, it could become known, but it also may not. And if not, it just provides you with additional security. And then also you want to do a per-user hash because that just makes it zero effort to implement, and you really do get better security.

Leo: Bill Prast, Tampa Bay, Florida, wonders about a career - oh, this is the one you were talking about - wonders about a career in Internet security: I love Security Now!, wake up listening, throughout the day and at night, and I listen still. [Laughing] Just to Security Now!, over and over and over again. I'm an IT Security AS degree seeker, currently prepping for a CCNA security credential. You and Leo give me an inspiration. I'm a 27-year-old single dad and night-time student. Good for you, Bill. He's working hard. I work in the IT field now. It's great experience, not so much money. What kind of career outlook do you think I'll see in a year in the IT security field? What can I expect to face in the IT security field? Bill Prast, IT technician at General Telecom LLC, studying for his security credential. I think that's great. Good on you, Bill.

Steve: And for what it's worth, many of our listeners are students and have been inspired by the podcast to focus on security.

Leo: Oh, we're required listening in a lot of classrooms.

Steve: Yeah, we're assigned. And I really do think there is a bright future. I mean, I wish there weren't from the standpoint of, boy, isn't it too bad. But increasingly in the future, security won't be something outsourced. It won't be something that is an afterthought. It won't be something where a company hires a security consultancy to, like, set things up and then wander off. There will be an in-house security person. And I just think - one of my theories of employment in the future is specificity. The more specific your skills are, you can be found on the Internet, you are something that people understand. And being security, I think, makes a lot of sense. It is, unfortunately, a growth industry.

Leo: Yeah, exactly. Good business to be in today.

Steve: And Leo, you've got to go...

Leo: No, I don't have to go. No, no, no. Let's keep going. I'm having a good time.

Steve: Two more questions.

Leo: Two more questions. We can make it. We're trying to keep the show a little shorter per request of many people on Twitter. Not too short. Don't worry. It ain't gonna be a 10-minute show. No way.

Steve: No, we're an hour and 20 minutes right now.

Leo: Yeah, we're good. I think an hour and a half is the right length for the show. Ben Moore in Southaven, Mississippi, wonders what the heck is JavaFX? I'm a longtime listener to Security Now!, proud owner of SpinRite. I pretty much quit installing the Java Runtime Environment on new computers. Still have a lot of older ones where it's installed. Recently these older computers started upgrading to Java 7, but when they do they're also silently installing something called JavaFX. What the heck is JavaFX, and why do I need it? Maybe I should just uninstall Java everywhere.

Steve: [Sighing] Okay. So this is more bad news.

Leo: Oh, no.

Steve: Yeah. This is Oracle deciding that they want a piece of the Flash/Adobe AIR/Silverlight market. This is their special effects delivery platform. It used to be a script that would run under Java, and they've now compiled it into byte code as a separate library, and they're now promoting it as essentially the same thing as Adobe AIR or Microsoft Silverlight, one of these content delivery platforms, to provide a library of helper effects for people who want to do this in Java. And sadly, they're just sending it out, along with Java 7, which we know will have things that are exploitable. There will be mistakes there. People are getting it even if they don't use it, which means that websites, web browsers will display something, fall through some glitch in the code, and get their users in trouble.

Leo: Great.

Steve: So I say, more than ever, if you don't know you need Java, get rid of it. I love what Apple has done. The notion of it disabling itself and requiring you to manually say, yes, turn Java on...

Leo: That's how it should be.

Steve: Oh. It is exactly the way...

Leo: On a per-session basis, yeah.

Steve: Per instance, yes.

Leo: So it doesn't just remember that.

Steve: And if you leave it on, it turns itself off again. That's where we're headed...

Leo: It's the right way to do it.

Steve: ...is that kind of security.

Leo: And that would be good for everything. You don't need to have it running always.

Steve: That's called NoScript, Leo.

Leo: Yeah, oh, yeah, that. Oh, yeah, that. All right. You're going to win eventually. Charlie Guthrie in Richmond, Virginia - I'm sorry, Richmond, California, just down the road apiece - wonders about his voice as a password: Steve, Vanguard, the mutual fund company, now offers the choice of using voice recognition as a password. Not as a password for their website, only when you're interested in talking to them on the phone. Before when you called they asked you several prearranged security questions. Now you just have to repeat a set phrase, and if the voiceprint matches, they'll talk to you. I don't think you've ever discussed this technique on your show. I was wondering if you consider it to be secure, that is, secure assuming they do everything else right, like securely storing the voiceprint and so on. That's interesting. Voice calls are only 8-bit sound, so there's not a lot of information there. Maybe it's enough.

Steve: Right. Certainly the loss of fidelity over a phone is a big problem. What's interesting is that I did a little research because I was fascinated back in my high school years with speech synthesis. And in fact one of the things I did at Stanford's AI lab was program their DEC PDP-10 to sing the Eagles song "Desperado."

Leo: [Laughing] "Desperado." Was it good?

Steve: Actually it was. There was something called a Votrax synthesizer which was a phoneme, something called a formant synthesizer. The idea, the theory of speech is that you have what's called a buzz source, which is our vocal cords, and then the actual physical...

Leo: The mouth shapes it, yeah.

Steve: Yeah, well, actually our throat and tongue, all of the aspects of our larynx physically shapes that. And forensic studies that have been done by the FBI and other law enforcement have shown a remarkable specificity for recognizing specific voices. That is, for example, in a courtroom, they have shown better than 0.3, lower than 0.3 percent error. So extremely low error. Now, this isn't - it's not just you speak and we will identify you from our entire user base. No. The way this works is, we have figured out who you are before. In setting this up, you've identified yourself. You're saying this is who you are. We have your voice on record. And so it's an AB comparison system where it verifies - it's very much like a thumbprint, where at the DMV you gave them your thumbprint, and then later on they say is this the same thumb that we saw before.

Leo: The thumb's harder to record, though, than voice. What if I have a recording of that person saying that sentence?

Steve: It is absolutely spoofable, yes.

Leo: Very easily.

Steve: Yes. So it's not super secure. But over a phone line, I guess my point is that, if you know who you're expecting it to be, then you actually can rule them out or not. And it is incredibly difficult for someone who has not recorded the authentic voice to spoof them because no two people have the same physical throat mechanics. And so it's both physical throat sizes and also then - and that's sort of like static aspects, and then the dynamic aspects of the actual way they speak that specific phrase. So I think we're going to see more of it. Certainly it's been fodder for sci-fi for a long time.

Leo: Cutting off people's thumbs has also been fodder for more than sci-fi.

Steve: Right.

Leo: Great episode. Great questions. Thank you all for asking. You of course can go to GRC.com/feedback to ask your questions. That's Steve's site. You can also follow Steve on Twitter. He's @SGgrc. He also has @SGpad and @SGvlc. You didn't really follow the Microsoft Surface announcement, huh?

Steve: I didn't.

Leo: I'm curious, well, at some point you'll - it's one of those things where they didn't announce availability, price or anything. So it's going to be - they said when Windows 8 RTMs, but that's - who knows when that is? September? I don't know.

Steve: Well, and I'm just not a Win8 person, Leo. I'm staying with XP till...

Leo: I guess you're the wrong person to concern yourself about this yet.

Steve: Yeah, that whole pane look, or whatever the hell they call it, it just looks awful to me.

Leo: Yeah, I can't really imagine you - it's like having a command line person use a GUI. It's just a little too much.

Steve: Yeah, that was a painful transition, too.

Leo: I'm sure that was very difficult for you.

Steve: I only launched Windows in order to use Micrografx Designer back in the early days, when I actually needed graphics. Otherwise I was happily in my command prompt, writing SpinRite in assembly language.

Leo: And I presume, because I know you have some Macs, but you don't use OS X day to day, so I presume you did not order one of those Retina Display MacBooks.

Steve: No, but I did salivate. I just don't use it enough to justify a couple thousand dollars.

Leo: Neither do I, but I did it anyway.

Steve: I know you did.

Leo: And now I can't use anything else. I have to use my iPad, my iPhone, and my Retina because everything else looks blurry.

Steve: It really is spectacular.

Leo: It does make a difference. You start to get used to it. It's not a good thing, actually. Don't get used to it.

Steve: I noticed that when I had my iPad 3 underneath my antiglare plastic for the one I carry around outside. And the two that I have in the house are not antiglare. And, oh, they look so nice because the antiglare filter does knock back some of the sharpness of the screen.

Leo: They've done something, because I know you don't like these shiny screens, but they've done something in the new MacBook Retina that is much less glary. I don't know, maybe there's micro scoring or something, because there's some glare, but it's blurry. It's not the mirror image that you used to see. So it is, it's much nicer in that regard.

Steve: Good.

Leo: And I'm sure they'll put that on the iPad next time. Steve Gibson is at GRC.com. That's where you go to find SpinRite, the world's finest hard drive maintenance and recovery utility.

Steve: Don't be shy.

Leo: Don't be shy. Buy, buy, buy. Come on down. You should have, like, a sale someday. 10 percent off for the first 20 buyers. Nope, nope, nope. I said it, he didn't. Don't get your hopes up. You can also get a lot of free stuff. See, that's the point. Steve gives you everything except that one thing. And maybe the Ketosis Flute. Think we're going to have to charge for that.

Steve: Well, and what we do, I'll note that even somebody who bought SpinRite 20 years ago, we will still give you a discount on today. So we are never going to leave anybody, any of our users high and dry.

Leo: Steve is conscientious, that's for sure. He also spends a lot of his money on things like transcriptions of this show. He makes 16Kb versions. That's all available at GRC.com. For the larger file size or the video, you can get that at TWiT.tv. We do Security Now! Wednesdays, 11:00 a.m. Pacific, 2:00 p.m. Eastern time, soon to be even more on time. I'm working on it. I'm going to get out of bed earlier so I could be here on time. 11:00 a.m. Pacific, or 1800 UTC, for those of you worldwide. And I just wanted to mention that we've got a team of geeks. I thought they were in a biker gang, but they say no, we're an IPTV gang in from Canada, actually Adam Erstelle, Kirk Fierback, Ken Delaney, Lyle Bryant - the handwriting's not mine - and Grant Backus [ph]. Thanks for you guys to come in. And they all enjoy the show. They were just on tenterhooks the whole time, Steve, seriously, the best audience is during this show because they're like the super nerds.

Steve: Glad to have them.

Leo: It's like having five Sheldons. No, I'm just kidding. Thank you, Steve. We'll see you next week on Security Now!.

Steve: Thanks, Leo.