SECURITY NOW!

Transcript of Episode #297

Pass-Sentences??

Description: After catching up with a number of extra-interesting security news of the week, Steve and Leo explore the recently raised suggestion that using a three-word "pass-sentence" such as "I like tomatoes" would be MORE secure (and far more memorable) than "J4f6<2". Short sentences are certainly easier to remember ... but more secure?

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-297.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-297-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode 297, recorded April 20, 2011: Pass-Sentences.

It's time for Security Now!, the show that covers your security online, with the king of all of this, Mr. Steve Gibson of GRC.com. Steve is the man who first discovered spyware, coined the term, wrote the first antispyware program. But he's been doing security for a lot longer than that. He's also the author of a great hard drive maintenance and recovery utility that's still number one: SpinRite. Hey, Steve.

Steve Gibson: I'm glad that we've dropped the "the man, the myth, the legend"...

Leo: Oh, you're that, too.

Steve: ...intro. I don't really miss it.

Leo: He blushes every time. I don't want to make you blush. And I really should say, first and foremost, one of my dearest old friends. And always a pleasure. We had you on TWiT a week and a half ago with Pournelle and Dvorak, and it's just...

Steve: It's really fun.

Leo: It's just so great that us oldsters, because we're the same age, can still participate in technology.

Steve: Yeah, we're still getting to our computers.

Leo: Well, more than that. We have a context.

Steve: We may be sitting on balls, but we are in front of our keyboards.

Leo: You're not sitting on a ball, are you?

Steve: No, I'm not. Although...

Leo: I am. No, I think what we bring to the table, though, is this memory and this history. Aaron Newcomb was doing FLOSS Weekly, he was in here, as you know, a minute ago. And we were talking about old computers and your first computer and stuff. And I think that there's a whole generation that grew up with super powerful computers and iPads and stuff, and they don't really - they don't know the history of all of this.

Steve: Well, and I see things on the 'Net from people who are clearly newbies, which I know is great. Everyone is at one point.

Leo: Everybody's got to start somewhere, yup.

Steve: We were once. But there certainly is something that you get from having seen a lot of things before where you say, okay, well, let's put this into context. And in fact that sort of is a perfect segue into today's topic because somehow through Twitter, and maybe you were the genesis of this, I started getting a lot of people asking me about a blog posting which actually is not even new. It was originally posted in I think it was September of '07 by a guy in Denmark who's not really a security guy. He's sort of a new media guy. Actually, I think he's, like, in charge of new media for some fashion designer company.

Leo: Oh, you're kidding.

Steve: No.

Leo: I should have dug deeper. I mean, I think I saw it on Hacker News.

Steve: Anyway, the concept is he makes the assertion that three-word sentences make - are, like, vastly stronger than anything else you can do for a password. And so...

Leo: And memorable.

Steve: And the advantage being, yes, like "I like pudding" or something. And so he makes the assertion that that's vastly stronger than, like, random gibberish passwords. And so I started getting everyone in Twitter saying, is this true? Does this work? Does this make sense? And what's compelling is that the graphical design of this page is very nice. And unfortunately he's made a number of logical mistakes.

Leo: Oh, shoot. I thought it was too good to be true.

Steve: Yeah, it is too good.

Leo: Oh, shoot.

Steve: But and then in his FAQ that he has, he also contradicts himself in a number of places. So, I mean, I wanted to give this our type of Security Now! rigor.

Leo: Good.

Steve: And also sort of move the bar a little further in our discussion of passwords, which we haven't covered for, like, five and a half years because it turns out that was the original series of topics when we began Security Now! with Episode 2 or something, I think it was. So I think everyone's going to find it very interesting. And we're going to talk a lot about bits and password technology and binary stuff. Some interesting news. Some broke just an hour ago. I may have scooped you on one of these things, Leo.

Leo: Oh, good.

Steve: So we'll see.

Leo: I love this. Well, we always have - I love this show. There's always something to learn. There's always something new. And I meant to send you an email about this, and I'm so glad that our listeners did - they're very proactive - because I really want to hear what's wrong with this.

Steve: Well, and before I forget, I want to mention that there are, throughout this, a number of URLs. And I was thinking, how am I going to share these with people? Well, first of all, I should mention that I recently grabbed the domain GRC.sc, as in shortcut, so that I could do my own URL shortener and give it the kind of security features that we would like it to have. I haven't yet implemented that. But then I realized I had recently tweeted all of these things. And so even for old-school people, I mean, I got excited when I saw that Jerry Pournelle had a Twitter account, and I forgot to give him some feedback about that when I was on TWiT with you guys a week and a half ago. Then I realized it's not really an account that he's tweeting on. Rather it's some bot that takes topics from his blog or something. So it's some sort of an automated gizmo.

Leo: We've been working on getting him on Twitter.

Steve: Yeah. And anyway, so even for the people who are not on Twitter, if you just go to Twitter.com/SGgrc, what that gives you is my feed, and all of my recent tweets will be there. So, for example, I informed people several days ago that this was going to be our topic this week because I finally thought, okay, this needs to be what I talk about to satisfy all the people who are asking me. So where we have strange URLs, I've already tweeted these things, so you can find them, you can just pick them up at Twitter.com/SGgrc.

Leo: That's a great idea. That's great. Thank you, Steve. So before we get to that, and of course we have security updates and a lot more to talk about, as well...

Steve: Some fun news, yes.

Leo: Steve's been busy. And people are saying in the chatroom, is Steve going to cover the iOS tracking issue? Well, don't you know Steve by now? Of course he is.

Steve: Top of the list. You see it right in front of you there, Leo?

Leo: I do.

Steve: iOS is and has been tracking us.

Leo: So, yeah, this broke, this iOS thing just broke yesterday, I think.

Steve: Well, actually it was blogged by one of the discoverers earlier today, today being the 20th, which is Wednesday, of April. There's an O'Reilly conference going on in Santa Clara from yesterday, today, and tomorrow, April 19th through the 21st, the so-called Where 2.0 Conference. And two guys, Pete Warden and Alasdair Allan are planning to announce today their discovery that our iPhone and 3G iPads, that is, those devices with cellular connectivity, so not, for example, the iPod Touch nor the WiFi-only iPad, that those cellular connected devices from apparently starting with v4 of iOS, so not prior to that, but starting with v4, those devices are regularly recording the position of their location and synching it with iTunes to build a large and growing composite file called - it's titled "consolidated.db."

And what they discovered was this file contains a wealth of information about the location of that device or devices over time, including the longitude and latitude and a time/date stamp for that location. And Alasdair also wrote, because he was the person who did this blog posting on O'Reilly's site that got picked up by the various news spools, he said, "For example, in my own case, I discovered a list of hundreds of thousands of wireless access points that my iPhone has been in range of during the last year."

Leo: Wow.

Steve: So as far as they know - and this is significant, I want to make sure we don't overblow this - as far as they know, this is not being shared with Apple. That is, they don't believe it's being transmitted to Apple. But it really does beg the question, what is this for? Why is this file being built? What purpose does it have? How is it accessible to the outside world? They raise the point that, from a privacy standpoint, once it's known that it exists, and now everyone knows it exists, if anyone had access to it, then your location and whereabouts through time, when you're traveling around with your iPhone, as most people do, or their iPad, as I do, is all there.

For the Mac, they produce a free little bit of software which you can get to from the link from their blog posting. And I imagine this will be much in the news in the next day or two. And again, you can get the link, I have it as an O'Reilly shortened link. But again, I linked to it from my Twitter feed, so Twitter.com/SGgrc, and you'll find my first note of this earlier this morning on April 20th. Their blog posting, Alasdair's blog posting has a link to the software.

So I grabbed it and downloaded - it was tiny, I think 132K or something - ran it, and immediately I'm looking, sure enough, at a map of the U.S. with a little concentration showing in Southern California. And I zoomed way in on that. It also has - it has a timeline slider along the horizontal and a zoom scale on the vertical. And I zoomed in, and sure enough, I mean, it knows where I've been spending all my time. It was right there on the map.

So they have asked Apple what's going on, and there's as yet been no reply, though I don't know what length of time Apple had to reply in. I'm sure we will, within a day or two, know much more about this. And apparently there's even more stuff in this file that they will be talking about. For example, I only mentioned the longitude, latitude, and timestamp. But we also know that all WiFi contact, we know from what Alasdair wrote, all WiFi contact is being logged in this, too. So I wonder why? And we don't know.

Leo: Could be diagnostic.

Steve: Eh, yeah, I mean, again, it's like, okay, don't know why that is. I'm not hugely concerned. He does make a point that, if you use encrypted backups through iTunes, then this file is encrypted. So it's only if your...

Leo: Well, it's on the phone unencrypted, but the backup is encrypted.

Steve: Correct, the backup on your Mac is kept encrypted. So anyway, so if someone had access to it, and you were concerned with someone knowing where you had been and when, then that's all there. So it does sound like something that we need the option of disabling or manually enabling or Apple telling us what their plan is for this thing. So in any event, users should know that their phones are doing this. And anyone listening to this podcast and my Twitter feed now knows.

Leo: Amazing.

Steve: Microsoft has also released a third antispyware tool. Now, as our listeners know, my theory of Microsoft's slow march forward is that this is just Microsoft's compromise with people who are selling things for Windows that Microsoft recognizes they actually really do have to incorporate into Windows, but they can't do it all at once because everyone would scream unfair competition and antitrust and so forth.

We first saw this with the firewall. I remember Gregor Freund, who was one of the cofounders of ZoneAlarm, Zone Labs, one of the co-founders of Zone Labs, I remember when he was invited up to Redmond, along with all the other personal firewall vendors, for a meeting where Microsoft was disclosing that they were going to put a firewall for the first time ever in XP. And I don't know who they were trying to pacify. But they said, oh, don't worry, this is not going to compete with you commercial firewall guys because we're doing inbound only, not outbound, and it's off normally. So most people will never turn it on.

And so I remember Gregor and I used to chat on the phone frequently back then. And he sort of scratched his head, and he says, well, you know, we're not real happy about it. But they're right that it's really you know, it's not really going to impact what ZoneAlarm's functionality or Symantec's Norton firewall and so forth, all of the various things. We had, what BlackICE at the time and other firewalls of that era. But as we know, Microsoft has slowly moved that forward. In SP2 of XP, now it's on by default. And then now, in Windows 7, and I think in Vista, it also does outbound blocking and application-level blocking.

So today, although it took many, many years and many versions and service packs, they're pretty much done with personal firewalls. And one wonders why you would add a third-party firewall. So I think we're seeing and have been seeing the same thing happening with antivirus. They're very slowly creeping forward, adding features, adding capabilities. And in some cases, I mean, the MSRT is the Microsoft - no, sorry, Security Essentials. Microsoft Security Essentials was clearly a big step forward in stepping into this territory. The MSRT was the scanner that we've talked about on a number of occasions which is updated with every second Tuesday of the month, which they introduced to defend themselves against the problem people were having of updating Windows and then having that crash their systems because they didn't know that they had a rootkit installed which was assuming fixed entry points for functions which would change when Microsoft revised the core Windows things.

So what we have now, like in the last couple days, is a third thing from Microsoft. Microsoft calls it the "Safety Scanner," Microsoft Safety Scanner. You can find it simply by putting "Microsoft safety" into Google, and it'll immediately jump to suggest that the third word should be scanner. I do not suggest you use this as a three-word password, however.

Leo: We'll talk about that.

Steve: We'll be discussing this later. So what this is, is basically it's built on the MSRT, the Microsoft Software Removal Tool. However, this brings a full and mature set of antiviral, antimalware signatures with it. It's big. It's 72MB. And interestingly, it only survives for 10 days from the time you download it. But also interestingly, you don't need to install it. It doesn't install. It's a standalone EXE called MSERT. Maybe that's for expanded or extended or something, I don't know. It's available in 32- and 64-bit versions and runs from, for example, a USB drive, a little USB stick, thumb drive, with no problem.

So if you want to use it, the idea being it's standalone and is a full mature scanner. It's been tested versus the other current commercial scanners, and it has not found to be wanting. That is, it's state of the art in terms of what it finds and sees. And I've already had some feedback through Twitter from some people who have used it under safe mode, for example, to successfully scan and remove malware from their machine. And Microsoft still...

Leo: Now, it doesn't replace an antivirus. It's kind of a different beast; right?

Steve: Correct. It is a scan-on-demand scanner. And they say on that page, quote, "The Microsoft Safety Scanner is not a replacement for using an antivirus software program that provides ongoing protection." So this is the third asset in this arsenal. So we have Security Essentials running all the time, MRT being updated automatically and running monthly, and you can of course do, as we know, if you type "MRT" in the Run dialogue of Windows, you can get it to come up, and then you can ask it to do a full scan. And this looks, I mean, this is MRT but with a full complement of virus patterns. I mean, it's the same MRT dialogue. You can ask for a quick or a full scan in the same fashion. And it looks the same from a UI standpoint.

So anyway, I wanted to let everyone know that that's there. And, for example, if a friend said they think they have a problem with their computer, you could boot, from a CD, boot a safe version of Windows that you know doesn't have a rootkit installed, and then easily run this from a thumb drive to scan someone's system on an on-demand basis. Which I think makes it pretty cool.

Leo: Yeah, that's great. Just got a note in our chatroom. There's a company called, I think it's a German company called Ashampoo which makes an antimalware solution along with other things, office suites and so forth. And they just sent a note to all their customers saying that their customer database had just been hacked. And they discovered it, but apparently a lot of information was stolen, including addresses, billing information, credit card - well, wait a minute. I'm sorry. Credit card information definitely not affected. Sorry. So there you go. So it's just email addresses. But we have to add that to your - don't you have a new section on this show called...

Steve: Yeah, and it was empty. And so now...

Leo: Well, I just have something for you.

Steve: It's the A&B, Attacks and Breaches, section.

Leo: There's one. There's one every week, I'm sure, if you really look.

Steve: Oh, and actually a lot of them are, like, sort of off to the side or not big, you know, it doesn't quite make it onto the radar. But, yes. We're seeing problems. And I'll tell you, it really does seem to be a growing rate and scale of these kinds of breaches now.

Leo: Yup, yup.

Steve: Which is why I created a section for it.

Leo: Now, I wanted to ask you about this next topic because I use, as you know, I use Dropbox. And Miguel de Icaza, who is a great developer and a really important guy in the open source community, said, "What the hell?" Because apparently Dropbox has been assuring everybody that they use strong encryption that they can't decrypt.

Steve: Well, yeah. And there's two things. There's two issues. One is that Dropbox recently updated their terms of service to say explicitly what was always apparently implicit. Quoting from their new terms of service, they say: "As set forth in our privacy policy, and in compliance with United States law, Dropbox cooperates with United States law enforcement when it receives valid legal process, which may require Dropbox to provide the contents of your private Dropbox. In these cases, Dropbox will remove Dropbox's encryption from the files before providing them to law enforcement."

Now, this sums up more perfectly than I ever could why I chose Jungle Disk for my own remote cloud-based backup. And that is, I did a full security analysis of Jungle Disk and verified that all that is ever being put up to Amazon's S3 cloud stuff is pre-encrypted data. That is, my Jungle Disk client has the key and everything it sends. So all Amazon gets is opaque pseudorandom noise that they have no ability to decrypt. I mean, it's full TNO, Trust No One, as my acronym for this, which is the only way I would ever store something in the cloud. So here Dropbox has formally acknowledged that they have the ability to decrypt the contents of all of their users' data, and that they will do so when ordered to by a court order from the United States.

Leo: So as Miguel points out, well, if they can do it by court order, then they've had that capability all along. So they essentially misrepresented the encryption capabilities.

Steve: Well, and see..

Leo: And he says this is a larger issue, not so much government, but that means employees could do it. And even with a company that has very strong data policies like Google we see these things happen.

Steve: Very, very good point. It means that keys could get compromised; keys could get lost. Or, as you say, you could have a bad apple employee who realizes, hey, we're hosting a celebrity. I wonder what he's storing in his Dropbox?

Leo: So I make sure I don't put anything of a private nature in my Dropbox. But I'm going to make sure I don't. And you're right. I think if you're going to do it, if you want to store something like financial records, use Jungle Disk.

Steve: Well, and here's another - well, or, and this works, too...

Leo: Pre-encrypt.

Steve: Exactly. Only store stuff that you have encrypted up there, where you're pre-encrypting that data. And this is why, when I see someone saying "industry standard AES 256-bit encryption," it's like, that means nothing. I mean, unfortunately it catches out people who don't listen to this podcast, who assume that, if you're using state-of-the-art encryption, then you must be safe. No. I mean, I would imagine that means that the link is encrypted. And it does sound like they're storing it in an encrypted fashion. But they're storing it with a key that they have. So that doesn't really help.

Leo: Yeah. That's the question, who has the key?

Steve: Right. And the best solution is for no one but you to have the key. And the only way to do that is to pre-encrypt and only store encrypted stuff in the cloud. Now, the other issue that came up was a question of their authentication. Someone named Derek Newton, who is a security researcher, was poking around in Dropbox-like applications, and he just decided he would take a look and see what they left behind, what was left behind after they installed. What he found was that, specifically in the case of Dropbox, there is a single file called config.db, which is an SQLite database file, which contains the email address, the dropbox_path, that is, where the Dropbox folder is on your system, which is being synchronized to the Dropbox in the cloud, and the host_id. Any SQLite DB-compatible client is able to open this file and look at it.

And what he determined by experimentation is that the only thing that identifies you to Dropbox is the host_id. There is no other lockage of that file to a given system. And so what he posted - and again, I learned about this from people saying in Twitter, hey, Steve, what do you think about this? And this has been a constant flow for the last couple weeks. And I mentioned last week that I hadn't had a chance to dig into this, but I would, to look into it and verify it. So I did want to follow up for everyone who's been wondering.

So what this means is that, if you weren't protecting this file, or if anything got onto your system which was able to grab this file through social engineering attack or spyware or malware, whatever, if you lost control of that file such that it was in any way exfiltrated from your control, then that file can be installed on any other system. And that provides the sole authentication of you, the instance of you, to Dropbox such that, with no other information, no username, password, no logon, anything, that authenticates that new system. And there is - it doesn't appear as a new machine in the set of machines that you have authorized to use. It's merely a clone of that first one, which then has full access, unencrypted access, to your Dropbox contents. Which to me says these guys aren't really looking at security.

I mean, on one hand we know now that they can decrypt the contents of our Dropboxes.

And this could clearly have been done in a way that was more secure. Even if you change, if the user changes his username and password, that doesn't invalidate the host_id. It still functions. And so if somebody had it, their connectivity survives across a user changing his username and password. So it's just they really could have easily done a much better job of hashing username and password into this, tying it in some fashion, for example, to the serial numbers of the hard drives on the system. I mean, just anything to make it more difficult than simply one file which you can put on any machine anywhere, and suddenly it's authenticated just as solidly as the system it came from.

Leo: Yeah, that's not good.

Steve: So not good news over on the Dropbox side.

Leo: You know, there are alternatives. LaCie has a similar service to Dropbox that's Java based. I don't know if it's more secure. But I think maybe it's time to look and see what the other alter- I love Dropbox. And I hope they respond to this by making it more secure. That would make everybody happy.

Steve: I think they can. I mean, one would imagine they will because it's so trivial. I mean, all they have to do is listen to this podcast for a while.

Leo: Right, and add some encryption features. The other one to look at, I'll take a look at, is from LaCie, it's called Wuala. Randal Schwartz told me about it. It's Wuala.com. Very similar to Dropbox. I'll look and see if they say, when they say all files get encrypted - see, that's the thing, is "get encrypted." Well, what does that mean? Where, is the question.

Steve: Yeah, exactly. And that's just it. Unless there is a full security analysis available of how it works and what it does, you just can't trust it.

Leo: Here's what Wuala says. It says all files are directly encrypted on your desktop. Your password never leaves your computer. Not even we as a provider can access your files or your password.

Steve: Well, that's all good sounding.

Leo: That's what you want - validated, of course.

Steve: Yeah.

Leo: I'm going to take a look at them. Randal Schwartz recommended them. He likes them a lot, so I'm going to take a look at them as an alternative to Dropbox.

Steve: So under my miscellaneous category, I did note, and I just got a kick out of this, that Bitcoin's BTC, which is the three-letter acronym for the currency, the Bitcoin currency, Bitcoin's currency to U.S. dollar exchange rate has reached a new all-time high of $1.12 per bitcoin on one of the major bitcoin exchanges, which was mtgox, the @mtgox market. So I just thought I would let our listeners who are following Bitcoin and interested in knowing that bitcoins keep getting more valuable, ephemeral as they are.

Leo: More valuable.

Steve: More valuable. Well, I mean, you can trade them in for dollars. You can get actual dollars...

Leo: But that worries me. That worries me. I feel like that's money laundering by somebody, somewhere.

Steve: But that's privacy. Isn't that a good thing?

Leo: No, but I - yeah. But I've got to wonder, who's buying bitcoin for real greenbacks, and why?

Steve: Yeah, I don't know. I just think it's cool technology. And it works. And we have a full security analysis of it, and I can vouch for the fact that...

Leo: No, Bitcoin's good. I'm not saying that. The only thing that worries me at all is this notion of there being a market for greenbacks.

Steve: Yeah. So I've been getting questions, mostly over in the GRC newsgroup, about when GRC would offer ShieldsUP!-style analysis scanning, port scanning, on IPv6. And so I contacted Level 3, and I said, hey, what does it take to get some IPv6 block? And they said, all you have to do is ask. Are you ready? And I said, uh, not quite yet. So then I called my local T1 provider. I have, as always, two T1s coming here into me. And for the first time really in a long time I'm glad I've got T1s because I don't think Cox or Community Cablevision or whoever does my cable modem, I don't think they would know how to respond to this question at this point. Ultimately they're probably going to have to. But I don't even know if they'll make IPv6 service visible to their customers. Who knows what they're going to do in order to actually roll this out.

But I have a direct connection with Cogent, who is my provider of connectivity on the other side of my pair of T1s. And the engineer from Cogent has been with me since before Cogent bought the T1 business from Verio. You'll remember all that, Leo.

Leo: Oh, yeah.

Steve: And so I said to them, hey, I'd like to have IPv6 service, the idea being that I'd get it here at home, you know, slash office, where I hang out and work, and update my router to handle it because I've got an old Cisco router, but there's firmware available for Cisco's IOS. And there was a name collision, of course, with Apple's iOS, but Cisco was there first. Cisco's IOS has been able to handle IPv6 for many years, since I think '03. So I had to update that and my various other stuff and then begin to perform some experiments and bring myself up to speed and write some code and be able to support IPv6 access of ShieldsUP!.

So they had never done that before, when I asked them. Which gives you some sense, and this is just in the last couple weeks, gives you some sense for where IPv6 still is, is that Level 3 wasn't pushing it, but they did say that new deployments of large customers, when they were setting up new people, they were often using IPv6, though still with some IPv4. But IPv6 was in the conversation. Over the course of a week or two, because I know the engineer at Cogent, they were able to move things around and provide me with IPv6 service. They gave me a choice of how many IPs I wanted. Is everyone sitting down? The normal allocation for a customer, the typical sort of, like, minimum that you get, if you've got a network, is a /64. Now...

Leo: How many addresses is that?

Steve: Okay. IPv6 uses 128 bits of addressing. So a /64, and this is the normal networking that is used, is 64 bits for the network number, and 64 bits for the machine in the network. So we're used to this notion of there not being a fixed division between the network and the machine. So, for example, we have, like, a /24 will be 24 bits for network. And IPv4 we have a 32-bit address. That leaves you with 8 bits for machine. So, for example, the famous 192.168.something.something, well, technically that's a /16 because you have .something.something. But it's normally used in a /64 mode where you'll have 192.168.0.something, meaning that only the last byte in that network is set to which machine you have so that you could have 253 machines on that network, which is more than anyone has at home, typically.

So but this changes when we move to IPv6. Just because the reason we began to constrain the size of the number of machines in the network was that it gave us many more networks. And we started to have to need, we needed that variable boundary between networks and machine numbers because we were running out of available networks. So they sort of pushed that boundary downwards, making each network smaller so that we would have more networks. Which is probably the best way to say it. But with 128 bits of addressing under IPv6, we don't ever need more networks. So typically they're going to just divide the 128 bits of IP space in half and not argue about it. You get a 64-bit network often if you're like a regular customer with an ISP. So that's 2^64 different IPs. And I did the math earlier, I don't have it in front of me, but it's billions of billions of machines. Wait, 2^64, I can do it.

Leo: You got your programmer's calculator there?

Steve: You bet I do. 2^64 is...

Leo: A lot.

Steve: ...1.8 times 10^19.

Leo: Whoa. That's pretty big. That's how many IP addresses you get?

Steve: Yes. No, well, actually no. That's what most people get. John said, well, you know, Steve, if you ever needed to, like, renumber yourself, then we would make you move to a different /64. So we're going to give you a /48.

Leo: Oh, well.

Steve: Okay. So that...

Leo: So lower is more.

Steve: Yes. The 48 is the number of bits they have over on their network side. So 128 minus 48 is the number of bits I have.

Leo: I get it.

Steve: So I have 80 bits. Which is 1.2 times 10^24. Now, okay. So 10^24, that's 24 zeroes. Now, a million is six zeroes. So this would be four sets of million. So 1.2 million million million million IPs I'm going to have. Now, I don't know what I need all those for. But...

Leo: It just shows you how much there is.

Steve: Yes. That's why I wanted to illustrate this. It's just it gives us a sense for, I mean, and you can imagine, in an environment where we're all running out, and everyone's oh, no, no, the sky is falling, IPv4, now the registries have run out, now, blah blah, we're hearing all this. So you'd think maybe they'd learn the lesson and be a little more parsimonious with these IPs. It's like, well, how many do you really have over there? How many do you need? But the point is, even in this climate of, well we ran out of IPv4 space, so we don't want that to happen again. But with 128 bits, they're just saying, aw...

Leo: Take all you want.

Steve: You can have, Gibson, you've got, what, five, six computers over there? We're going to give you 1.2 million million million million IPs, where you can just get lost. I mean, I can't even write that down. It's more of a burden than it is a benefit. But that's what - and so it's funny, too, because then I wrote back, I said, okay, John, how many do you have? Because he just seemed to be peeling them off with such alacrity. How many do you have? And he said, well, we received a /32 from ARIN, that's the North America registrar, and another /32 from RIPE, which is the European one.

So /32 means 32 bits of network, and then so like they have a - so the 32 bits are fixed. And their number is 2001:550, which is a way in IPv6 parlance, and we'll be discussing all of that in the future, that's their network. They then have - they can do anything they want to with the balance of the bits out of 128. And since 32 from 128 is 96, they've got 96 bits that they can allocate any way they want to. So, for example, when they gave me a /48, they have 65,538 networks of that size. So they have 65,000 /48s, of which they're giving me one. Most people they'll give /64s. So they've got vastly more of those networks. Actually they have another - that gives them, if they use 64s, that gives them another 16 bits of allocation. So that's another 65,000. So they're just not worried. Oh, and if they do actually run out, they just ask for another one.

Leo: Because there's plenty.

Steve: Yes, well, there are 4.3 billion of those /32 networks because we know that /32 - think about that. /32, that's the total number of IPs that we have now in IPv4. That's the total number of networks of that size that the registrars can just peel off at will, 4.3 billion networks, each containing 65,000, 65,000 /48s.

Leo: Wow.

Steve: So, yeah. We're fine.

Leo: In the words of Tamahome, you sound like Carl Sagan. Billions and billions of IP addresses.

Steve: I did love this. He said, in his note he wrote he said, so customers, really, they can have a 48, a 56, or a 64. He says, to get a /47, which would be like, you know, one more bit above 48 because you're then squeezing the network down to get more machines. He says, "To get a /47, a customer would have to prove a need (which no one has done yet)."

Leo: Or will ever do.

Steve: Okay, because, yes...

Leo: Try and prove a need for a 64. That's more than the whole Internet now.

Steve: Oh, exactly. That would be needing 1.4 times 10^14 individual IPs. Which, as you said, is like many, many, many millions of Internets now.

Leo: I have an unusual application. I'm going to be...

Steve: And you know, I've got to say, as I was thinking about this in the context of

ShieldsUP!, how many times have we run across situations where a problem was that it was possible to scan, that is, Nimda and Code Red and Blaster, they used to be scanning the Internet. And even though we think of four billion IPs being a lot, if you've got a botnet, for example, you can scan a lot of IPs in a relatively short time. And that's one thing that dramatically changes because it's no longer the case that a large percentage in a small region of allocation are going to be valid IPs. You have 128 bits, they're going to be relatively sparsely allocated. I mean, I'm going to use a handful of IPs out of - actually I'm behind some powerful NAT-y kind of stuff, so...

Leo: Yeah, you don't really even need more than one, yeah.

Steve: Yet here's this massive block which is all given to me. And God help something that tries to scan that. So scanning really...

Leo: Oh, that's a good point.

Steve: Yeah, scanning, the nature of scanning, really does change in an IPv6 world. Now, you might argue that, well, you just scan the, like, zero, one, and two of each range. Well, but I'm not going to put my IP, the one I use, down there. It's going to be floating hidden somewhere in the mist.

Leo: Up there in the billions.

Steve: Oh, yeah.

Leo: Wow, that's so cool. Well, I just hope this all happens, that's all. I mean...

Steve: Yeah. That'll be fun to talk about because...

Leo: APNIC ran out of IP addresses, you know, IPv4 addresses.

Steve: Yup. And there sure is some reluctance for it to happen. Otherwise Microsoft wouldn't be offering millions of dollars for Nortel's block of IPs. Clearly there's some reluctance to make this move.

So a customer of ours, of GRC's, David Ward, I guess he has a computer company called Wellmax Computer. He just wrote a short note that I wanted to share. He said, "I had a friend bring a laptop to me and tell me that the fan was failing. The machine would freeze after about five minutes into the Windows XP boot process." Wow, it was taking more than five minutes to boot?

Anyway, he says, "He had taken it to three other techs before me, who all said the same thing, that the fan was failing, so the machine was overheating and then stopping the boot. I experienced the same problem, however, after booting into safe mode. And having the machine stay on for an hour, I knew this was not a fan issue, but a hard drive issue. Sure enough, after running SpinRite for over 180 hours," he said, parens, "(It's okay, at one point it said it had another 1,089 hours to go.)" Because, remember, it estimates how long...

Leo: It's guessing, yeah.

Steve: Yes, it estimates how long it's going to take based on how long it's taken so far. But often bad spots are in the beginning, and so things pick up the pace dramatically after SpinRite deals with the early problems. So he says, "It finally finished after finding and fixing many bad sectors. Now the machine booted successfully and ran very well. All the customer could do was shake his head and utter the phrase, 'Amazing,' several times. Thanks so much. David Ward, Wellmax Computer."

Leo: Amazing.

Steve: Amazing. So thank you, David, for your testimonial.

Leo: So I tweeted this. I think I saw it in Hacker News. It's funny because I didn't realize it was a four-year-old article. But a fellow named Thomas Baekdal, I think that's how you say it, in Denmark, published an article on his blog four years ago which has gotten a lot of attention lately, saying who needs those really hard-to-remember random punctuation passwords, when you could just use three common or uncommon English words and do better? I believed it. Looked good to me. But we've got to give it the Steve test. Steve Gibson is going to explain why three English-language passwords may not be the best solution. All right. Let's talk. Three words. "This is fun." Is that a better password?

Steve: Oh, god, no.

Leo: Than "asiw>"...

Steve: No.

Leo: No?

Steve: No. In fact, that was one of the conclusions in bold on Thomas's page. He says "this is fun" is 10 times more secure to use than "J4fS<2." Now, just look at what it took me to communicate the second one versus "this is fun," and that gives you a clue. But let's start a little bit back further.

Leo: Let's get the science. Let's do the math, yeah.

Steve: Yeah. So once upon a time we had the notion of a password. And we understand that that became unusable, unsafe, because of dictionary attacks where, well, first of all, people, I mean, famously would use the word "password" as their password, or all nines, or 1234567, or something guessable by somebody who knew them. There were all kinds of reasons. So then we sort of, in my own thinking, evolved to passphrases where you had the notion of longer is better. And that's sort of where my own thinking moved through with this notion of something that was memorable. But then, in thinking about it more and also literally doing the math, I moved to passcodes. And our listeners have heard me referring to passcodes now for some time. And so the question is, what about short pass sentences? Does that make sense?

So let's back up a little bit and look at the attack model, that is, the threat model for the use of passwords. I think this whole thing, the whole notion of something the user knows as one factor of authentication remains important because, not only is it the original technology for online authentication, but, sadly, we're seeing such slow adoption of other things like the YubiKey or the smartphone-based one-time password technology, I mean, we have all the technology that we need to solve the problem. But it's just adoption rate. And what's interesting is much could be done to strengthen even the use of passwords, much more than has been done. And what I'm hoping we're going to see is that happening over on the server side. And we'll talk about what that is because that's a function of sort of this password technology.

So we start with something the user knows, which is the idea being that it's a secret. And obviously you'd like it to be unguessable by somebody who knows you. You'd like it to be not easily brute-forceable, like not "aaaaaaa" as a password. So as we've talked, there are two technologies, essentially, for a user's interaction with logging in online. At the server side, they either store the password, or they hash it, and then they store the hash. The idea being that, unless you're in France, where there seems to be some legislation requiring that the user's password itself be made available, unless that's the case, there's really no need for an authentication system to store your password.

What you can store is something which you can uniquely derive from the password, and that provides some privacy protection for the user because we have the cool cryptographic technology now known as a one-way hash, a cryptographic hash that can take a password and turn it into essentially a pseudorandom token for which there is no way to reverse that process. You can't go backwards because it's inherently an information lossy process. So that allows an authentication provider to just store that hash. And when asked to reauthenticate, the user provides the same thing, which gets hashed in the same fashion and produces an identical result, which can then be compared for identicalness. That prevents someone from saying, well, you almost got it right, you know, you didn't get the punctuation correct, or you didn't get the capitalization correct. There's no way they can do that. All they can say is, nope, that's not what you gave us before, try again.

Now, cleverly, rainbow tables were developed as a means of getting around this one-way hash concept. A rainbow table is essentially a dictionary of all the results of hashing everything. So you sit there in what's called an "offline mode" - and we'll make that distinction very clear in a minute - an offline mode with your field programmable gate arrays, your graphics processing units, your GPUs cranking away, whatever it is. Or you network this among a whole bunch of people, producing rainbow tables, the idea being that you just brute-force everything through the hashing function, and you store in a database, a huge database, like a terabyte database, the result of all of those hashing operations. In that case, if you ever do get access to the result of the hash, you can look that up in this rainbow table, and it provides you the reverse direction, only because it did all of the forward directions, and it saved them all. That allows you to essentially look up the result of the hash and figure out what you could input that would give you that result.

It turns out, the good news is, that's easily defeated, too. All you have to do is add something which in cryptographic circles we call "salting." You add some salt to the hash, meaning that you take anything, and it could just be some hopefully not easily guessable blob, and you always append it, or prepend it, whichever, doesn't really matter, or even mix it in with what you're going to hash. Which essentially produces a custom hash function. It's the way to think of it: Salt gives you a custom hash function so that, with your particular salt, no one else's rainbow tables will work because you've mixed something in beforehand.

Now, if they did get access to your salt, somebody could generate rainbow tables for that. But that's sort of a - that's a different problem. It prevents a single, like an SHA-1 or an MD5 hash, which are well-known public uniform hashes, it prevents rainbow tables for being generated for those hashes. And by the way, those exist. They're big, but they're available on the Internet, rainbow tables for those hashes. So you would never want to use those without adding some salt. When you do, it makes those existing pre-created rainbow tables obsolete.

Now, I talked about online versus offline. The reason this is important is it's a function of how many opportunities an attacker has to test a password. If something is offline, that is to say it's - for example, TrueCrypt. We've talked about cracking TrueCrypt. The idea would be you have a hard drive that you want to encrypt. And it is unfortunately subject to an offline attack, meaning that an attacker could set himself up so that, at an extremely high speed, they are presenting to the TrueCrypt algorithm, which remember is open source, so there's no secrets there, which wouldn't make their job impossible if it weren't open source, but it's just easier that it is. They would present to that algorithm every possible password they can come up with as fast as possible, run it through TrueCrypt's front end to create a key, and then see whether that key decrypts the beginning of the hard drive, like a hard drive sample that they have taken. And the point is that nothing theoretically prevents them from doing that at an incredibly high speed.

Leo: Well, but doesn't a good program - I would imagine TrueCrypt would slow them down. I mean, I know SSH slows you down. Doesn't TrueCrypt say, oh, you've made 10 guesses, let's wait a minute?

Steve: Well, actually TrueCrypt itself could. But the algorithm - but the problem is the bad guys can simply, since it's open source, they have access to...

Leo: Oh, they can read the file, yeah.

Steve: ...anything that does that, and they could just short-circuit it.

Leo: Got it.

Steve: So the problem is, in an offline attack like that, where you have access to everything - and, you know, cracking DVDs is the same thing. Anything offline, then the idea is that it scales linearly. That is, the more processing power, the more GPUs that you bring to it, the more you can try. But notice that online attacks are completely different.

Leo: Oh, yeah. And that's what he's talking about, we should say.

Steve: Yes. Well, he discusses the issue. But yes, he says...

Leo: Baekdal is assuming in all of his calculations you could do a hundred tries per second.

Steve: Exactly. Now, I think that that's probably very generous.

Leo: Yeah, that's way fast.

Steve: Because, yes, because we know that when we're trying to log onto something, we put our username and password in, click Submit, and we kind of sit around and wait for typically a couple seconds, whether we're right or wrong, before the system comes back and says, oh, that's correct, or oop, nope, sorry, something's highlighted in red that your username and password don't match, try again. So my point is, and this is really true...

Leo: By the way, somebody claiming to be Baekdal is in our chatroom. So I'll watch what he says, and I'll give you his feedback.

Steve: And I think it probably is because he did see that I was going to be talking about this, and he did send me a tweet yesterday because he knew that I was going to be talking about this.

Leo: Yeah. And as we mentioned, he's not a security expert, he's a new media guy.

Steve: Well, and he makes a number of very good points. But I would only take issue with the notion that three simple short English words are more secure than something completely random. And I can do the math...

Leo: And we're going to talk, we're going to explain it, yes.

Steve: Exactly.

Leo: But so far so good. He's made a very conservative assumption, which is that you'd have a hundred shots a second at this.

Steve: Well, okay. Now, maybe if you - and this would be a function of the way the website is designed. You might be able to have, like, a hundred clients or a hundred agents all connecting to a web service at once, trying to get into the same account, in which case that would scale up your rate. But no matter what, you've got Internet roundtrip delays. You've got lost packets. You've got TCP retransmissions. There's, like, substantial overhead which scales back dramatically the number of attempts that any kind of brute-force guessing is able to make. Which dramatically changes the math. And this is, more than anything, that's the point that I would like to get across to our listeners is it is very different to choose a secure password for an online service where the attacker would be reduced to an online attack versus anything like TrueCrypt or an attacker that could do an offline attack.

Now, this isn't an absolute because, as we mentioned earlier, and we do talk about weekly, it is often the case that databases are being stolen. And so an online service whose database was stolen could then be subjected to, could have its accounts subjected to an offline, that is to say, very high-speed attack. So it's not necessarily - we can't guarantee that an attacker would be using the same online transaction delay that we are. But I think it's a reasonable assumption.

Leo: Okay. So whatever he says is, we're talking about in the realm of attacking a web password.

Steve: Yeah. Yeah.

Leo: Without physical access to the server.

Steve: So what we really need, when I talked earlier about some changes that could be made, that we're in desperate need of having made, and that is that, in an online scenario, and you mentioned something a second ago...

Leo: What?

Steve: ...that does, that introduces a delay.

Leo: Oh, well, a lot of programs do. I mean, on the Mac, SSH. A lot of programs do.

Steve: Right, secure shell.

Leo: Yeah. Or if you're logging into your Macintosh, if you enter the password twice or three times wrong, it just pauses. Doesn't pause very long. It gets longer each time. But it pauses.

Steve: And of course UNIX and Linuxes have done this for time immemorial, and specifically to prevent this kind of brute-force attack. All you have to do, in fact, you could make the time delay exponential, where each time you guess wrong, it doubles the length of time you have to wait. Well, we know how quickly powers of two pile up. And you don't even have to set a fixed lockout. When I was designing my ecommerce system, I was very conservative. And if somebody messes around with the ecommerce system in a way that my system detects doesn't make sense, they're blacklisted. I mean, it's like, I don't care, I'm sorry about that. And we've had some complaints from people who seem unable to type their name correctly.

Leo: No SpinRite for you.

Steve: Yeah. And again, I have no problem then with handling that through a human interaction loop. But we certainly don't want to allow an automated script of any kind to be able to pound away. So that is lacking to an alarming degree in today's websites. And that is something so simple to fix, if it was just universally applied that, like, a progressively longer delay was present.

Now, the flipside is they know that's going to cause some customer support problems. And so that tends to work against them doing that. But it's just, it's so simple to do, to introduce that delay of response on the server side, and radically scales up the security in the face of anyone trying to do an online attack. So the question then becomes, what's more secure or less secure? And again, I'm not meaning to be attacking Thomas at all. That wasn't my intent. But many people asked if short English words were more secure than a short bit of gibberish. Now...

Leo: And that's his assertion, so that's what we want to know.

Steve: Now, one of the things that I have referred to when we were talking about passwords in the past is something that is a point that Bruce Schneier made that I thought was really astute. And Microsoft's own security people have said the same thing. They've said the danger of using something that you can remember is that it's going to be too simple, because it's difficult to remember J4fS<2. That's arguably difficult to remember, more so than the phrase "this is fun." And so Bruce's point was that, use something difficult to remember which you write down and stick on a piece of paper in your wallet. And he made the point that we already know how to secure and manage little bits of paper. We do it. We have a wallet. We know how to handle that, more so than something which is easy and memorable, but as a consequence of that sacrifices security.

So in the tweets that I received from people asking me about this, they were using different little three-word sentences. Thomas on his site uses the sentence "This is fun." The problem is that the issue in terms of bits of vocabulary size and the strong tendency people would have to design a sentence which makes sense, that is, they're not - they would not tend to choose three truly random words from a really large vocabulary because "aardvark" has two A's, and who can remember how to spell that. Or tapioca might be a problem and so forth.

So in the samples that people were sending me, they were using three words from maybe a thousand-word conversational vocabulary, for example. Well, we know that, if you took three words chosen purely at random from a set of a thousand, well, that would be a thousand times a thousand times a thousand possible sentences. So there we're at 10^9, or one billion possibilities, so a thousand million billion.

The problem is that, if you instead were to choose characters from the full ASCII set, that is to say, lowercase alpha, that gives you 26 characters, uppercase alpha gives you another 26, the digits 1 through 10 gives you 10. And just the characters available on your keyboard give you another 35. So that's 26 plus 26 plus 10 plus 35 is 97. So that means that, if you randomly choose a random character from an alphabet of 97 characters, and you used, for example, a six-character token, which is what Thomas uses in his example, that would be 97^6 possible combinations. Which compared to taking three words from a thousand-word vocabulary, which gave us one billion combinations, using just six characters is 833 billion combinations, so 833 times stronger, all other things being equal.

Now, you could argue that, okay, wait a minute, the weakness in the argument is this thousand-word vocabulary. But it turns out, even if you used a 500,000-word vocabulary, which is the number of words in the English language, for example, it turns out that that doesn't scale much faster, either. 500,000 words in the English language is 18.93 binary bits, and then - if you use three of those. But to do that you're using really obscure words.

So my feeling is that, while it's tempting to use memorable sentences, short, three-word sentences, the problem is that we want to have multiples of those. We know that it's much safer to use - not to reuse the same password on different sites. Which is why of course LastPass famously solves this problem by giving us a local database which is locally encrypted and is able to figure out what domain we're trying to log onto and fill in these forms for us. So it solves the problem of a pass anything, whether it's a passcode or a passphrase or a pass-sentence, on a per-site basis.

If we're going to be using different sentences on different sites, then we still need to write them down to remember which sentence we used on which sites because many of us are logging into all kinds of different services throughout our day, and that number of services increasing. So we still need to have something written down to disambiguate. And that assumes we didn't have LastPass, which sort of solves the problem for us anyway.

So I guess my feeling is that, if we're going to use a single short sentence across the entire Internet, we know that's not safe. And if we're choosing it from a small vocabulary, it doesn't give us as many possibilities as - and again, you can of course add words. It doesn't give us as many possibilities as using a short token from a much larger character set or from a full-size character set - 97 characters, for example, are available on an English keyboard. And there isn't anything wrong with this notion of writing it down because, if we're going to use different ones for different sites, then we need to write them down anyway in order to remember which one we used where.

So I wanted to respond to everybody who was saying, hey, Steve, what do you think about this? Wouldn't it be great if this was really secure? My feeling is, well, if you want to use short sentences, make the words really obscure.

Now, the other thing that I didn't touch on is notice that, in the case of "this is fun," well, that makes grammatical, semantic sense. And that's another huge weakness of the whole sentence problem. And that is, we didn't use "barnacle itch robbery" as our three words. We used "this is fun." And in all the examples that I've seen people sending me, their takeaway from this was...

Leo: Use "this is fun."

Steve: Well, or...

Leo: Something grammatical.

Steve: "I like tapioca." Now, sure, it could be I hate tapioca, I despise tapioca, I slurp tapioca, I mean, you could have different things. But notice, if you impose the filter of this makes sense, this is a grammatically correct sentence, now you have hugely changed the odds in favor of the attacker because, for example, in the case of a thousand-word vocabulary, a conversational vocabulary, a thousand times a thousand times a thousand, that gives you a billion, well, only a small fraction of those billion possible arrangements of three randomly chosen words make sense, would actually be a sentence. And although we don't know that it exists today, we can imagine if this became in vogue that the extension of the classic dictionary attack would be the sentence attack. It would be an English sentence-generating algorithm that chose words, starting with small, short small ones, and then ramping up to larger ones, very much doing sort of a brute-force sentence attack, which is completely possible.

Leo: But what if, for instance, instead of using spaces in "this is fun," I used an arbitrary punctuation of my choosing, and something easy for me to remember, let's say "%," and I wrote "this%is%fun," it strikes me that we're not saying - Baekdal says it's computationally more difficult. We might dispute that, but it's certainly difficult enough, isn't it?

Steve: I really think so. Yes. Anything you do to obscure. Now, the other thing you can do, and we didn't talk about this, is case sensitivity. You would want to make sure that the system receiving the password was case sensitive. There are some, for example, that are deliberately not. For example, I remember when I was talking about LastPass...

Leo: Banks.

Steve: Well, but LastPass, the password is definitely case sensitive, but they deliberately lowercase the email address, which is the other portion of what you use to identify yourself with because email addresses are not case sensitive themselves. So they felt that they would have a problem with users who might formally use a capital for their first letter of their name, if that appears in their email address, but then either use it or not use it when they were logging on. And that ambiguity, since email doesn't care, they didn't want their system to care. But yes, Leo, as you made a point, we know that there exist systems which are deliberately case insensitive because they want to ease the logon burden of their users, and in the process mess up their security. So case is less strong, for example, than the use of...

Leo: Punctuation or numbers or...

Steve: ...punctuations and numbers instead. And in fact, you could also go hackeresque and use numbers instead of letters that look sort of the same. But again, I mean, I guess my feeling is the bottom line is, if it is too memorable, it is too easy because it is its memorability - memorability?

Leo: Yes, memorability.

Steve: Memorability. That didn't come out right somehow.

Leo: It's a word.

Steve: It is its memorability which we like, but to me that says somebody else could guess it. Somebody else would have some reason to include it in some big corpus of things that they were testing against. Now, the one thing that completely changes everything I've just said is that we're assuming knowledge on the part of the attacker that they don't have. That is, no attacker knows anything about the scheme the user could have used. That is...

Leo: And it would probably be foolish for them to make assumptions about that.

Steve: Yes.

Leo: Because it would...

Steve: Because there are an infinite number of schemes available. So, and that's one of the things that Thomas did on his page that was a little misleading, where he talked about the comparative strengths of different passwords. He showed, like, all lowercase as opposed to random symbols and said that the all lowercase one was weaker. Well, it's only weaker if the attacker knew that your password was from a limited character set of all lowercase. If the attacker didn't, and they still were having to try all possible characters in all possible character positions, then all lowercase is no weaker than random symbols because the attacker doesn't know what you've chosen. So anyway, so my feeling is, yes, Leo, you could use words if you do something to obfuscate them, and the attacker will not know what you did.

Leo: But there are the two issues, one of which you raised, which is that, if you use words, you're going to be tempted to use that password all the time, on more than one site, so that's a bad weakness.

Steve: You're going to be tempted to use nonrandom words in a sentence. And as soon as you do that, wham, the strength of what you have done just collapses.

Leo: Right, right. So that's the same problem, essentially, which is memorability is anti-security.

Steve: It implies a weakness.

Leo: Yeah, it's a weakness. And then there's a second issue which the chatroom raised which I thought was quite astute. If you use something that is a phrase, and somebody sees you type part of it, they are a lot closer to guessing your password than if you use random letters and numbers.

Steve: Very good. Now, and Thomas did, on his side, he made a point which I did like. And he said, if you use words, you can probably type it in easily and quickly. And it's like, well, okay, that's a good point. You would be less prone to typos than if you're trying to type J4fS<2. On the other hand, I've pretty much got that memorized by now, and I think all of our listeners have. Don't use that one.

Leo: And of course systems like LastPass, KeePass, 1Password, RoboForm, all use a master password which does not change. That might be a good candidate for a passphrase that is memorable to you. I tend to use longer passphrases than three words. And I also like to put punctuation in, commas, exclamation marks, and I capitalize. So all of that makes it more difficult to guess, especially if an attacker doesn't know I've used a passphrase.

Steve: And the one other thing that we've talked about before, but I'll remind our users, is never write the actual password down.

Leo: On a post-it note, and put it on your monitor.

Steve: Well, no. What I meant was, do something to it, like in what's written down. For example, always leave off the first or the second characters, which you know to put on, but no one who discovered that written down would know.

Leo: I actually do that.

Steve: They would, if they got a hold of your wallet that had your little post-it note in it, they'd go, ah, I've got all his passwords. But when they try one, it wouldn't work, and they would have no idea why.

Leo: And there's a scheme I'm sure that you could come up with that you would somehow modify your consistent passphrase with something like the name of the site that would make it recoverable. I use SuperGenPass for that, where I have a master pass which it hashes, in a one-way hash, the master pass and the name of the site to create a unique password for the site. All I have to remember is the master pass.

Steve: And then you're always able to regenerate that from that.

Leo: I can regenerate that. But ultimately LastPass, or KeePass if you like open source, is such a good solution because it creates strong passwords and memorizes them.

Steve: And when Thomas did this post back in August of '07, LastPass didn't exist. And frankly, LastPass has solved all of my online logon problems. There are some situations which are not online, where you can't have access to LastPass, and that's where this conversation, I think, is worth applying.

Leo: Right. His math, though, is not wrong at any point here, is it? Or are there errors?

Steve: No, I think his math is right. He just makes some assumptions. And the comparative strength issues I take issue with because he says, as I indicated, like all lowercase is less strong than uppercase, lowercase, numerals, and symbols, which really is not the case.

Leo: Got it. Very interesting stuff. I'm sure Thomas Baekdal would not disagree. He says the password he uses on his server is a 40+ character completely random password. He does not, and we should underscore this, use three words for offline sites. Always, if there is opportunity for an attacker to, at his leisure, attack, this isn't going to work.

Steve: And the one thing I forgot to mention that I keep meaning to mention is that, relative to brute-force attacks, we talked last week when this came up about this notion of memory hard functions. I also wanted to mention that, when we discussed WPA, the WiFi technology, these guys, because they did it with really good crypto in mind, they take the phrase, your login passphrase, and they salt it with the access point's SSID, which hopefully changes from one access point to another, that is, you did change your access point's SSID and did not leave it left to Linksys or D-Link or Cisco or whatever.

But then they hash it 4,096 times, not because doing it once isn't secure enough, but deliberately because they want it to take a long time so that it thwarts rainbow tables, both because it mixes the SSID as salt into the hashing function, but they do it 4,096 times so that there's a computational overhead for the actual production of the cryptographic key out the other end. And so that's one way of dealing with the offline attack problem, by just making it much, you know, 4,096 times longer to get a key, which you would then test against the crypto algorithm in a WiFi scenario. So really this is something that's been given a lot of time and attention because it's the way we authenticate, even today.

Leo: Slow the bad guys down.

Steve: Yup.

Leo: Steve, always fascinating. We have links to all of this information in the show notes. You have 16KB versions of the show, as well as the 64K full quality, and transcriptions available at your site, GRC.com. That's where you'll find SpinRite, the world's best hard drive maintenance and recovery utility, and lots of free utilities that Steve offers everybody for your security and convenience.

Steve: And something coming shortly called The Passcode Designer.

Leo: Oh. I like that.

Steve: That ties into all of this.

Leo: Yeah. So you'll find it, GRC.com. Steve is @SGgrc on Twitter. And next week we'll answer questions. So if you've got questions or comments or suggestions - and Thomas, if you want to respond - just go to GRC.com/feedback, and the feedback form there will go straight to Steve's inbox, where he will comb through it.

Steve, so nice to see you. Thank you so much. We do this show every Wednesday morning, 11:00 a.m. Pacific, 2:00 p.m. Eastern, at live.twit.tv. I hope you'll join us for the live show. If not, you can always download it from Steve's site, our site, or anywhere finer podcasts are carried.

Steve: Thanks, Leo.

Leo: Thanks, Steve. We'll see you next week on Security Now!.