SECURITY NOW!

Transcript of Episode #303

Password Haystacks

Description: Steve shares something of a revelation about the true nature of passwords and why "password entropy" really doesn't matter. He explains, therefore, how it's possible for passwords to be both memorable AND impossible to crack at the same time.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-303.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-303-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode 303, recorded June 1, 2011: Password Haystacks.

It's time for Security Now!, the show that covers your security and privacy online, with our guru, our man of the hour, Mr. Steve Gibson of GRC.com, the guy who wrote SpinRite, the world's best hard drive maintenance utility, but also spends a lot of time on security these days and has written many great security tools. Why are you giggling?

Steve Gibson: Well, first of all, I'm probably more like man of the hour and a half.

Leo: Or two. Or two.

Steve: Or two in some cases, yes.

Leo: But today we actually have something that is hotly awaited. You teased us last week.

Steve: Yes. I did. I got a lot of tweet feedback from people saying, oh, my - and I got a sense also for the fact that people are listening to the podcast over the course of the week because these tweets were coming in over the course of the week as people would catch up and listen to the podcast and then get to the tease for this week at the end of last week's podcast. And they would send me a note saying, oh, what have you come up with?

Leo: What is it? What is it? We can't wait a week.

Steve: Now I have to wait for a week. And in some cases, when it came in yesterday, I said, no, you don't, you just have to wait till tomorrow.

Leo: That's true. That's true.

Steve: That's now today.

Leo: So do you want to reiterate the tease? This came to you almost like Archimedes in his bath.

Steve: Well, and we should tell everyone, you know what's going on because I thought it would be much more fun for you to grok it and be able to work with me as we explain this. So what I'll say is, I'll remind people that I had been working on an idea that I called the Passcode Designer, which is up on GRC. If you go to GRC.com/passcodedesigner.htm, that's actually the reason I learned JavaScript. I'd never, I mean, I knew it was evil, but I hadn't used it, and I didn't know it because all the stuff I've ever done has been server-side code.

Leo: It's kind of the ultimate high-level language, and you do low-level languages.

Steve: Yeah, and I have to say, Leo, I'm very comfortable with it now. And I had forgotten, frankly, because I'd been writing assembly language all my life, and I hadn't had any need to mess around with high-level languages - I mean, I do see when I'm over in UNIX land and things, but that's sort of like, okay. And I have Perl as my scripting language over on UNIX. But after I've written, like, a chunk of functionality, I'll look at it and think, my, that's only 10 lines. Look how much I got done in 10 lines.

Leo: Good morning, Steve. Time to wake up and smell the coffee.

Steve: Hello, yes. The last person on Earth to understand, gee, those high-level languages are kind of handy.

Leo: If you know C, JavaScript in structure is very C-like.

Steve: It is. But, oh, boy, I mean...

Leo: But it's object oriented, which makes a big difference.

Steve: Well, yeah. There's an object-oriented flavor. But for me mostly it was that its scoping rules are really bad, so that you have to go to real - jump through hoops in order to, like, make sure you don't have a name collision with a module that you might import, somebody else's library. So it doesn't handle scoping very well. It's very lax about type conversion. But now I'm using that to my benefit.

There was a little chunk of code I shared, that I just wrote the other day, with the guys in the newsgroup, where I was deliberately using an array that contained integers in one place - oh, in fact it's on the "haystacks" page we're going to be talking about - where I'm assembling the first line of that chart showing the size of the alphabet. In one place I'm using the integers and knowing that JavaScript will convert them to strings, so I'm concatenating them onto a string after adding a plus sign. So it's 26 plus 26 plus 10 plus 33 for a large alphabet. And then the line right below I'm summing that into an integer, knowing that JavaScript will leave it alone, so that that's where I'm able to say "equals 95." And so, I mean, once you understand the rules, and you're careful, I mean, one of the things I guess I'm bringing to this is, because I'm coming to it from a lifetime of assembly language programming, my focus is extreme because it's had to be to survive in assembler.

Leo: That's true. I mean, you talk about scoping, assembly doesn't really, I mean, scoping rules are a high-level language construct.

Steve: Right. And so anyway, as a consequence, I don't want to set up another tease, but I'm not quite done. But today we've got round one.

Leo: Well, if you go to Steve's Passcode Designer site, you'll see the big, boom, "OBSOLETE" stamp right across it. We saw that last week. So today we discuss what it was, the insight Steve had that obsoleted the work he'd done previously and how...

Steve: Well, and all of the common wisdom that we, into our sixth year, continue to share. I think it was before we began recording, Leo, that I mentioned that I immediately scrapped my WPA WiFi password because it's obsolete. And how many times have I mentioned that I can't type it in. I've never had my iPod Touch on my WiFi because I just can't enter it. The only way to get it in was a copy and paste because it was 63 characters of nightmare. Now I've got something I can enter with my eyes closed. Friends can come over, and I can easily put them on my guest WiFi. And none of us have any less security. So it's a realization about what it is about passwords that actually matter, and it's different than what we thought.

Leo: So I always do this. I log into my computer to do the show. Seems like they always do this on Wednesday. Apple's got an update. And this is the one we've been waiting for.

Steve: Yes, it is. And it's good news, but unfortunately not as good as we were hoping. This is Apple responding to the new MacDefender threat that we've talked about for the last couple weeks that has been causing some serious trouble for Apple users and for Apple's own AppleCare folks, who've been getting, like, one out of every two calls they get now is about this MacDefender problem.

Between the time we last spoke about it and now, MacDefender the malware was updated so that it no longer required the social engineering of getting the user to enter their administrator password in order to install itself permanently. So we're seeing the typical evolution of this cat-and-mouse game. Apple just yesterday, on May 31, released an update, their third for 2011. It's tiny, it's only 2.1MB, so it's not one of these multi-hundred megabyte, replace your whole OS updates. And it fixed three things.

The third, under the title of "Malware Removal," they said, "Impact: Remove the MacDefender malware if detected." Which means, whereas they were telling, you know, the official policy at Apple was for the AppleCare employees handling customers not to take responsibility for removing malware, presumably Apple already had in the works that they were going to automate this and just do it for everybody. So the good news is, if it detects the MacDefender malware, all the variations that Apple knew, it will remove it.

So in the description they said: "The installation process for this update will search for and remove known variants of the MacDefender malware. If a known variant was detected and removed, the user will be notified via an alert after the update is installed. Additional information is available in this Knowledge Base article...." And if anyone's curious, because there is some more information which is sort of interesting, shows you dialogues and things, it's support.apple.com/kb/HT4651. And in fact there it says: "Security Update 2011-003 provides additional protection by checking for the MacDefender malware and its known variants." It says, "If MacDefender malware is found, the system will quit this malware, delete any persistent files, and correct any modifications made to configuration or login files. After MacDefender is identified and removed, the message below will be displayed the next time an administrator account logs in." And it shows you a little pop-up note just indicating that MacDefender had been found and was removed. So this isn't something which is fighting - MacDefender is not currently fighting Apple tooth and nail to hold on. However, Ed Bott, who was writing for ZDNet, just wrote that the new Apple antivirus signatures, meaning what was just updated, were bypassed within hours by malware authors.

Leo: Oh, dear.

Steve: And this is an update from this morning at 6:00 a.m. He said: "The bad guys have wasted no time. Hours after Apple released this update and the initial set of definitions, a new variant of MacDefender is already in the wild."

Leo: See, that's the problem with definition-based antivirus fighting. They're looking for strings. So all you have to do is change the strings, and the search engine stops working.

Steve: Yes. We don't have any behavior-based protection yet. And he said: "This one has a new name, Mdinstall.pkg, and it has been specifically formulated to skate past Apple's malware-blocking code. The file has a date and time stamp from last night, 9:24 p.m. Pacific time. That's less than eight hours after Apple's security update was released. On a test system using Safari with default settings, it behaved exactly as before, beginning the installation process with no password required. As PC virus experts know, this cat-and-mouse game can go on indefinitely. Your move, Apple."

Leo: That's what's really scary is that it doesn't require an admin password. That's the thing that worries me. Now, there's one thing Apple users should absolutely do if they're using Safari. There's a checkbox right in the General tab of Safari Preferences that they need to uncheck, this "Open safe files after downloading," because it isn't a safe file, and yet Safari will automatically open it.

Steve: Because it assumes safe unless it knows otherwise.

Leo: Right. So just uncheck that. It's a little bit of an inconvenience because PDFs don't just pop open. But they really shouldn't be anyway.

Steve: As we well know about PDFs.

Leo: As we well know.

Steve: Over under Windows, yes. So my thousand-yard read of this is Apple is maturing. They're developing big market share. And the hackers now have Macintoshes. So Apple and the Mac OS move into, unfortunately, where everybody else is, where Android is and Windows is, becoming a high enough profile target that those things about it that have not yet been hardened sufficiently are going to cause some problems. And it won't take Apple users long to wise up to it.

Leo: Yeah. I guess, I mean, but if you don't get a warning, it's more than wising up to it. This is scary.

Steve: Well, and as you and I have said, we have the problem also that, as a class, probably the majority of Apple users who...

Leo: We don't care. We haven't had to.

Steve: Exactly. Just don't have their guard up, haven't been afraid to click on things. I mean, someone sent me, a friend, a good friend sent me a forwarded email with a Windows Media file, WMV, Windows Media Video, that was apparently really, really funny. And she was saying, this is just fantastic. It's like, I didn't open it. I don't know what it was. I just - I can't. She didn't create it. And I saw this whole list of forwards, so she's not a super-sophisticated email user, either, because everyone she'd ever sent it to or received it from was there in this huge list. It's like, nothing can make me look at that. I don't care how funny it is. I just - it's not worth it. So those are the habits, unfortunately, that all PC users are going to have to be developing. And Windows users are a little bit ahead of the Mac users at this point.

Leo: Yeah. It makes me sad, but it was inevitable, I think.

Steve: It really was inevitable, Leo, that Apple is...

Leo: What's interesting is, this solves the raging debate between people who thought the Apple platform was inherently more secure. And I was in that camp early on but realized a couple of years ago, after seeing Pwn2Own and all these other security flaws, that it wasn't merely that we were more secure, it was really that the hackers didn't care about the Apple platform. It's a numbers game.

Steve: Our systems - it's a complexity game, from my viewpoint. These things are so complicated, the targets. Our computers today have - they're feature rich. Users want features. And the fact that Windows keeps selling, despite the fact that it's, I mean, viruses and malware on Windows is old news. But people still use it. And for example, look at that period of time when the Mac really was the safer choice. Windows was still charging ahead full speed. What this really says is that, okay, people realize there are problems. This is not all safe. But, boy, look at all those features. Or I can run all the software that I want under Windows, whereas I have lesser choice over on the Mac. And so what do they choose? Even though there's more security on the Mac traditionally, it didn't slow Windows down at all. Not detectably, anyway.

Leo: No. And the numbers game is that, as Windows users get more sophisticated, and 80 percent of Windows users now run security software, that 20 percent remaining is roughly equal to the penetration of the Mac in the marketplace. So now the Mac is at parity in terms of insecured users, unsecured users, so of course it's going to attack. I mean, it's actually, you could just watch that number as it went up to 70 percent, which is now roughly equal to 20 percent unsecured Windows users. Okay, I guess it's worth the effort. And hackers are watching it, and they go, yeah, let's do it. And they've had such success with MacDefender. I'm sure this is not the last of it.

Steve: Well, and fun, too. I mean, it's a new target. It's a whole new crop.

Leo: Let's not ignore the fun factor.

Steve: Let's not encourage them.

Leo: No.

Steve: Well, speaking of Windows and malware, I was tipped off by one of my very good participants over in the GRC newsgroups and someone who tweets stuff to me, Simon Zerafa, whom I've mentioned before, that Microsoft has in beta a new rootkit removal tool. So this is new. It's connect.microsoft.com/systemsweeper. And this is brand new from Microsoft. I haven't seen it mentioned anywhere else. I immediately sent the news out to everyone following me on Twitter and got a lot of people responding, hey, you know, thanks, didn't know about it, add this to the toolkit. This is a downloadable ISO image which you can install onto a CD or a USB drive. And so it boots an OS, a stripped-down Windows, which contains an antimalware and rootkit remover. And so it's - and it's free.

Leo: You have to boot into a clean version of the OS because otherwise you won't see the rootkit.

Steve: Precisely. And that's the trick is you have to - it's only by booting something you know is safe that you're then able to look into the target system and see whether there's behavior which is indicating known malware or the presence of a rootkit. And so Microsoft recognized that they're going to have to take this move, too. I mean, to their credit, rootkits are among the hardest problems to deal with. They're like, as we know, the next-generation malware is able to very rigorously hide itself from antimalware that's looking for it. And the trick is that, if you get in there first and modify the tools that the antimalware, antispyware, antivirus tools use themselves to find the malware, then it's able to hide. So anyway, this is connect.microsoft.com/systemsweeper. And I don't have any mature evaluation of it, how it compares to anything else. Being from Microsoft, I imagine it will always be there now, and it will get better over time, as all of their stuff does. So...

Leo: You know, I'm looking at this, it says, "Thank you for contacting Microsoft support." This is clearly intended to be a site that they would, when you call Microsoft saying I've got a problem, they would direct you to.

Steve: Exactly. It's like, oh, go here and download this and run it.

Leo: It says, "You've been directed here to download and install the beta version." So, yeah, it's interesting. They're not publicizing it. But if you call Support with a problem, they're going to say, hmm, maybe you need this.

Steve: And there are 32- and 64-bit versions. You can boot either on a given platform. But you have to choose the one that matches the target computer that you're wanting to scan. So I would imagine our super-savvy listeners will probably grab one of each and burn some CDs or create some bootable USB drives and have a very cool little bootable OS scanner for systems which they think may have a problem. And in fact I have already heard from someone who followed my tweet, got the kit, and discovered a rootkit that was unknown to them at the time.

Leo: Wow. Oh, man.

Steve: Yeah.

Leo: That's interesting.

Steve: So come to think of it, it's probably a good idea just to grab one and do the scan, even though you don't, I mean, that's just it, you don't know you have a rootkit because it's all about hiding itself.

Leo: And if you're, as I'm sure many of our listeners are, the geek that people call on when they've got problems, and you have some CDs - I know we all do, we have that little bag of CDs we bring around - two CDs, a 32 and a 64 of this, obviously. Good place to burn that now.

Steve: Okay. So before that happened, the big news of the week that I had at the top of my list was just, like, oh, crap.

Leo: Oh, gee.

Steve: We talked extensively about the breach at RSA and how, despite the fact that they never confirmed it - I immediately took the position, and I think I blogged it, that - oh, I did, I'm sure I did - that, look, the only thing they had to lose that was really important were the keys, the master keys to the kingdom, which were the secret private keys for the RSA SecurID tokens.

Well, now we have two confirmed reports of attacks on government contractors. The first was Lockheed Martin, who confirms it came under attack using spoofed RSA SecurID tags. And then next was a company called L-3 Communications that are much less well known, but sort of an off-the-map, high-end government contractor that's providing, like, communication technologies for the wireless attack technologies that our Defense Department has. So we now - we have confirmation that, in the quotes from these companies, that they detected attacks using compromised SecurID credentials, which is what we know escaped from RSA. So that's what we expected, and it's now confirmed.

And clearly these companies were lulled into some unfortunately false sense of security, followed RSA's weak recommendations. I mean, there's nothing you can do except absolutely stop using any of the authentication that might have been compromised. And RSA wasn't telling people to do that because that would be too horrific. Instead they were saying, oh, well, make sure that people don't get enough information that would allow them to leverage the fact that they may have compromised credentials, and they only need a username or a password or something other than the SecurID technology, which apparently got loose. So that was bad advice to follow, and we've got at least two contractors. I have heard more are under attack, but we don't have confirmation of those. So that's about as bad as it could get for RSA.

I commented a couple weeks ago when I noted that Netflix was now the No. 1 traffic generator on the Internet. And you and I have talked about it, Leo. And what surprised me was that BitTorrent was No. 2. It was like, wow. I mean, I just - I wasn't aware that there was so much torrenting going on. And anyway, I picked up a story that I thought I would share with our listeners because there is a takeaway for our listeners. And that is that Voltage Pictures, which is the production studio for the film "Hurt Locker," has decided it's going to go after, in court, and sue 24,583 BitTorrent users who illegally, they are alleging, downloaded and presumably watched "Hurt Locker," the movie that they produced.

Leo: Yeah. Almost 25, what is it, 20...

Steve: Just shy, it's 24,500 BitTorrent users. They've already filed lawsuits against 5,000

BitTorrent users. And the demographic breakdown I thought was sort of interesting. Of the large amount, the 24,583, 10,532, so nearly half, but by far the majority, are Comcast customers. About half of that amount, 5,239 are Verizon customers. About half again that amount, 2,699 are Charter. And about half again that amount, 1,750 are Time Warner customers. Now, in the case of Verizon and Charter, they are only disclosing the names of either 100 or 150, respectively, customers by IP per month.

So it's going to take a few years for this Voltage Pictures company to get around to everybody. But they've put a stake in the ground and said, this is what we're going to do. They have said that they would prefer to reach cash settlements with these customers as opposed to taking each one to court individually. I don't have any idea, I don't think they want you just to pay the cost of the movie, like the DVD or the Netflix download fee. They probably want something punitive as well because they have to be wanting to send a message to people who are doing this. But anyway, I said there was a takeaway. And I'm not a BitTorrent user, so I can't speak to the practicality of this. But I wouldn't do this from home.

Leo: Do it from work? Don't tell my employees that, please, okay?

Steve: Starbucks has free WiFi. That's where I would go.

Leo: I wonder. I bet you they must filter, or they'll want to start filtering.

Steve: I don't know, but...

Leo: I think that the courts have ruled that you're libel for what people do with your network, aren't you?

Steve: Yes, we just saw that recently. There was a very - it was a very - it got a lot of press. I didn't mention it on the podcast because I thought, okay, well, our users all know about that. But it was some guy who got in trouble for something his neighbor or some unknown person was doing. It's like, "Well, it's your IP, sir." "But I didn't do this." And they said, "Well, it's your IP."

Leo: Well, and that, by the way, that's one of the problems with these John Doe warrants that companies like these "Hurt Locker" guys are doing, is that it's IP based. So, I mean, BitTorrent is not anonymous. BitTorrent, your IP is known. And it's IP based. So what they do is, as you said, they subpoena Comcast and say, well, who is this? But it could be your neighbor using your open WiFi.

Steve: Well, I mean, years ago, I mentioned this years ago, there were - I got five IPs that were attacking GRC, that were on the Cox Cable network and appeared to be in Orange County. So I thought, well, I wonder what is going on? I mean, why is GRC being attacked? And this was back in the old DDoS days. So I called one of my FBI contacts, and I said, hey, can we find out who these guys are? Because I'd like to pay them a house call. And they said, yeah, sure. So they opened a case, and they asked Cox for the names that corresponded with these IPs. One was literally down the street.

So the FBI guy called the family, who said, "Hey, we've got someone that we work with would like to come over. You probably don't know this, but your computer is infected with malware." And they said, "What?" And they had a couple teenagers who were - oh, it's funny, too, because when I went into the house, it was summertime, and so they were both there. They looked like they'd been deboned, these two teenagers.

Leo: Yeah, that's what teenagers look like.

Steve: They were, like, limp on the couch. They didn't really move at all.

Leo: I know that well.

Steve: But they did...

Leo: I know that look very well.

Steve: They did complain that their computer could no longer burn CDs of illegally obtained music.

Leo: It's kind of a bummer, man. Can't burn it, man.

Steve: This thing was so infested with junk. And I can't remember now the current popular music downloading...

Leo: It's probably LimeWire, or maybe then it was Gnutella.

Steve: Before that. Gnutella, it was Gnutella. They had Gnutella. And so I didn't want to get them in trouble. I didn't want to be, like, the nark that arrived at the front door. So I was gentle with the family. And I realized that the kids were going to do what the kids were going to do. But I explained that this machine would no longer burn CDs because it could barely boot. It was so busy, there was like a war going on of king of the hill of malware in this machine. So it was interesting to see a real-world case of that. But I visited them because I had their IP from the not-very-sophisticated attack that they were doing on GRC's web servers.

Leo: But they, again, they weren't doing it. Their machines were.

Steve: Exactly. It was a bot.

Leo: They probably weren't smart enough to attack you.

Steve: They had an IRC-controlled bot. And it was one of many things that Gnutella had happily brought into their system. And I guess the mixed blessing was they were able to boot their CD after I cleaned their machine.

Leo: Man, that Gibson guy, he is rad. He fixed our machine, dude.

Steve: Oh, in fact I do remember the one comment that I got out of one of the teenagers when I was explaining to the parents who were there and concerned that this thing - that their machine had been all infested. And I said, "Well, yeah, unfortunately, it's because of the file downloading probably," I said. "I see that there's Gnutella here." And this one teenager said, "Wait, that's our music. We need that for our music."

Leo: Exactly what my kids would say.

Steve: And I thought, oh, uh-huh.

Leo: I give them an allowance to buy music, and they still keep installing LimeWire. It's like, don't you understand, the music industry would love, love to prosecute me. Please stop.

Steve: And years ago I did speak to the American Bar Association, their technical conference at the ABA, I was the keynote. And they asked some questions after I was through sort of bringing them up to speed about what was going on in the world. And they said, well, with all that's going on, how can we, like, keep our machines safe? Oh, and they said, if we have teenagers? And I said, okay. We all, all of you who are parents, and I have had plenty of girlfriends with teenagers, so I'm a virtual parent, we understand that there's no way to control them. And there was like all these heads nodded out in the audience. And I said, "Give them their own machine." I mean, and these were attorneys I was talking to, so they could afford their own machine, even though this was a while ago. I mean, it's easier to do now than ever. I mean, they probably all do have their own machine now.

Leo: Oh, yeah.

Steve: But there is no safe way to share a computer with someone who's going to be downloading everything that happens. And stuff their friends bring over, it's like, oh, you've got to see this, you know. And it's just going to be a disaster. But as you said, Leo, it all filters out

through the one communal IP. And we now know from this example that IPs are not safe against BitTorrent enforcement. So as I said to our listeners, don't do it from home. Go somewhere to get your stuff.

Leo: There's your advice from Steve Gibson, security expert. There are legitimate uses, many legitimate uses for BitTorrent, including downloading Linux distros and so forth.

Steve: And those you can do from home.

Leo: You can do those from your home.

Steve: In fact, I did get some flak back when I talked about that way with BitTorrent, back through Twitter, people educating me that, just as you said, Leo, there are people who depend upon BitTorrent for, like, nightly downloads and things that are being legitimately shared, not just copies of "Hurt Locker."

Leo: Yeah. So, I mean, people, we like to emphasize that because I don't want BitTorrent to go away. It's not a piracy tool, unlike, let's say, LimeWire or Gnutella or Kazaa, which really were piracy tools.

Steve: Yes. Okay. So since we last spoke we have a new zero-day problem with all versions of IE under all versions of Windows. Microsoft is blowing it off because it requires some social engineering. Unfortunately, IE is still by far the majority browser. And there's just no doubt to me, in my mind, that we're going to see this escalate fast. This was demonstrated by a hacker at the recent Hack in a Box conference in Amsterdam. It requires a drag-and-drop operation, that is, a mechanical, user-end, client-side, pick this up and drop it over. That's something that's a new feature in HTML5. So non-HTML5 browsers will be safe.

Microsoft was advised of this problem back toward the end of January. And so, like, they've had six months. They believed they had fixed it and claimed it fixed before the release of IE9. But it's still broken. And this hacker demonstrated under a state-of-the-art, fully patched IE9 that he was able to do this. He set up - he has a Facebook account with 180 Facebook friends. He set up a puzzle. He created an app which was a puzzle and shared this with his friends. He collected 80 Facebook session cookies. And this happens even if you're HTTPS, even if you've got maximum security. Basically this puzzle was a drag-and-drop puzzle. And so Facebook people said, oh, how cool, solve the puzzle. I don't know what it was, rearrange the pieces to form a picture kind of thing. That's all it takes.

It turns out that Microsoft made a mistake with their security zones. We talked historically about the IE security zones, where you've got untrusted and trusted and local, and you're able to set the settings differently, depending. Well, it turns out that, if you leverage HTML5 with drag-and-drop and iFrames and security zones, put it all together, you're able to steal session cookies from users. Now, Microsoft's official statement is, oh, you know, we're not that worried about it because it requires manual action on the user. Well, wake up and smell the coffee, Microsoft.

Leo: Not that much manual action.

Steve: No. And, I mean, Facebook is a made-to-order target for this. So whereas, for example, Firesheep was stealing session cookies, but would be defeated even in open

WiFi if the whole session was kept secure, and of course in response to Firesheep we know that Facebook added persistent HTTPS connections. This works anyway because it's working at the browser end and sending these session cookies to a malicious site where, the moment someone gets them, they can log in, impersonate the user, and get up to mischief. And what they'll do is they'll then post other malware on those compromised users' pages and use it as a means of launching further exploits. So I expect Microsoft will fix this - where are we? We're June 1st.

Leo: Second Tuesday's coming up, couple weeks.

Steve: Yeah, but they've got the maximum time because we're one day past the first Tuesday. Or, I mean, if this were yesterday...

Leo: They got 15 days or something, yeah.

Steve: ...it would already be the first Tuesday. So they've got the maximum time to respond to this, if they could do so within that window. And, well, they already thought they fixed it with IE9. So they understand the nature of the problem, and they didn't fix it. So I'm sure they can tweak it again, and I'll be very surprised if in two podcasts from now we're not talking about the second Tuesday patch of the prior day compared to that podcast. And like I said, it would be #305 where we say, okay, they fixed the drag-and-drop problem. But until then, this is going to - people will have fun with it.

Leo: Tune in at Episode 305.

Steve: And we'll see how good our crystal ball is. Now, I got a lot of interest shown in something that got picked up and spread around about Google speeding up SSL. Don't know if you saw that, Leo.

Leo: Yeah, I did. That was a couple weeks ago. And I hadn't played with it, I just noted it. I'd love to know what you think.

Steve: Yeah. There's nothing really to play with.

Leo: Just wanted to say how easy it was.

Steve: Well, they wanted - what they said was that they were getting a 30 percent improvement, and that was big.

Leo: And it was kind of a hack, though.

Steve: Well, not kinda. It makes me feel uncomfortable.

Leo: Yeah, yeah.

Steve: Because, okay. We did a podcast on SSL, so anybody who wants to go back and, like, refresh on exactly how SSL works, it's all there for you, both in transcript and in audio form. I don't know which podcast it was, but it was in the not-too-distant past [SN-195]. We spent the session, the podcast going through exactly how SSL handshakes work, how the protocol works. One of the things that happens is that you need to guard against man-in-the-middle attacks, that is, for example, when the client says to the server, here's all the security ciphers that I know about, and here's all of the authentication key exchange protocols I'm aware of. Then the server says - it gets that list, and it selects the one that they will use from among those.

Well, one of the ways to attack SSL, and it has been successful in the past, is to weed out and remove from that list as it goes by, the man in the middle, as it goes by is you remove the strong ones, and you keep the weak ones. Like one of them is none at all. Like I'm sorry, I don't do security. And so you could actually have an SSL connection with no encryption if the server accepted it, and if the server thought that that's the best the client could do was none of the above.

So as a consequence of that, in order to guard against, during this protocol establishment while they're getting themselves synchronized and connected, the very last thing which is exchanged is called the "finished message," which incorporates in it the hash of all the prior messages sent, so that when the recipient of the finished message receives it, they can take a hash of everything they've received previously and make sure that it matches what the sender sent. And if it didn't, that would indicate either a glitch in the line or, more onerously, somebody fiddled with their traffic as they were exchanging it back and forth.

What Google does with what's called "SSL false start" is they allow the client to send its first traffic prior to receiving the acknowledgment from the server that everything is copacetic. And so it's pushing the security envelope. Now, Google understands that. They understand there are some dangers. So, for example, in their spec for this, which they have sent to the IETF for consideration - I'll read something from it because it explains what Google understands the problems are. They said:

"When the client has sent its 'ChangeCipherSpec' and 'Finished' messages, its default behavior following the standard RFC is not to send application data," that is, the stuff we want to have protected, "until it has received the server's 'ChangeCipherSpec' and 'Finished' messages, which completes the handshake. With the False Start protocol modification, the client MAY send application data earlier (under the new Cipher Spec) if each of the following conditions is satisfied." And they enumerate:

"The application layer has requested the TLS False Start option. The symmetric cipher defined by the cipher suite negotiated in this handshake has been whitelisted for use with False Start according to the Security Considerations in Section 6.1." So what they're saying there is they acknowledge that some ciphers are not safe to use with this. So this is what makes me think, okay, I hope they got this right. And going on:

"The key exchange method defined by the cipher suite negotiated in this handshake has [also] been whitelisted for use with False Start, according to the Security Considerations." And in the case of a handshake with client authentication, where the client is offering a certificate to authenticate who it is, which is not the normal case for anonymous SSL, for example, with users on the Internet talking to servers, "the client certificate type has been whitelisted for use with False Start according to the security considerations."

So it's like, okay, well, how much do we really need to save one handshake exchange during SSL? How badly do we need that to be 33 percent faster by sending our traffic forward to the server before we've received its confirmation that it agrees with the integrity of the handshake that's been established. I'm not so sure. It can be disabled. They've been playing with this since Chrome v9. And they've been sort of doing it quietly. They found that most, with very few exceptions, all the websites around are false start-compatible already.

So this doesn't require any change in the SSL protocol. This is just sort of fudging it in a way that it really wasn't designed to be fudged. And the list of non-compliant IP addresses or domains is so small that there is a - that now Chrome contains that list in it. And so if you're trying, if a client, a user is trying to connect to one of those very few non-false start-compliant sites, then Chrome will know not to do that. It can be turned off by command-line option, but whoever launches Chrome with a command line? I guess you could create a shortcut and put that into the command line, build it into the shortcut so it would always be off. Maybe it'll be an option or a feature through the UI at some point. But Google thinks this is really wonderful. I hope it turns out to be, and not have problems. I mean, it does sound like the kind of thing that hackers are going to just jump on and say, ooh, how can we subvert this?

Leo: Well, I'm really glad you talked about this because I read it, and I thought, I wonder what - this doesn't seem quite right.

Steve: Yeah. Basically they're jumping the gun. And it's like - and having to do it in such a way that they recognize the gun might misfire if cipher suites and authentications, key exchange protocols, are not hardened against some potential exploits of this. So it's like, okay. I love that Google is pushing themselves for more performance. We've seen a number of things, like there's a cool add-in that you can get where it really helps web designers see where their servers are slow, what things are slow, how pages are, like, does a forensic analysis of what takes time on a web page. Google, to their credit because they've got so much eggs in the basket of web-based, in-the-cloud stuff, are really working to improve the performance of our experience, which I'm glad for. I just - I hope that this doesn't end up biting us in the end.

From the Twitterverse I have a bunch of quick little bits of feedback that are interesting. Again, my tweeting friend Simon Zerafa told me about QuickJava, which is a plug-in, quickjavaplugin.blogspot.com. And it does what we were talking about last week for Firefox. It gives you granular control over Java, JavaScript, and a number of things. I didn't enumerate them here, but I just wanted to give Simon a thanks and also to tell our listeners that there is something, if you were interested in disabling Java and being able to reenable it easily, the quickjavaplugin.blogspot.com will point you to it for Firefox.

Ben Pike followed up, or @BenPike. He wrote, "I work for a major wireless retailer. We use Cellebrite machines to copy customer data from old phones to new phones." So just as you suspected, Leo, here's someone who heard us talking about that, the problem with the Cellebrite technology which police officers are apparently using when they pull people over to, without a warrant, obtain all of the data in their cell phones. And as you surmised, some wireless companies like Verizon and AT&T and so forth were using it in order to help their customers move their data over. And here's someone who's doing that.

@holtcg, whose name is Chris Holt, says, "Most of the Imation brand flash drives have a write-protect switch, perfect for carrying around your suite of antimalware tools." That's following up on last week's discussion in our Q&A of the problem of, if you were using forensics on a flash drive to help clean bad stuff off of someone's computer, how do you know you didn't infect your own drive? And so a write-protect switch on the flash drive does that. And he says that Imation brand flash drives often have them.

Bryan Dort said, "Holy cow, @freshbooks is sending me a cake..."

Leo: Yay.

Steve: "...for just signing up for their service."

Leo: Yay. See, I wasn't lying about that.

Steve: And @TechJeeper, whose name is Cody Dean, said, "Thanks for the recommendation of the book 'Zero Day' by @markrussinovich. I'm hooked on this book."

Leo: I'm seeing that everywhere, by the way. People really, really do love that book.

Steve: Yes. And then also @jlanners, whose name is Josh, said, "Thanks @sggrc and @leolaporte for 'Zero Day.' It was fantastic." So he finished the book. And I also saw someone blogging, saying "Do not read this if you're on a plane."

Leo: Wait'll you land safely.

Steve: And anyone who's read the first chapter knows why you do not want to be reading this book on a plane. It would, I mean, oh, it would really give you the creeps.

Leo: They need to do an Audible version. Dr. Mom's saying I should read the Audible version. I'd like to.

Steve: And, by the way, Mark has tweeted, and I did also get this from Simon Zerafa, Mark has tweeted that there will be a sequel.

Leo: Oh, good.

Steve: So he's going to do a follow-up.

Leo: Good, good, good.

Steve: And lastly, @zkam, he tweeted, "New privacy settings for Firefox." And this is hot off the press. This is not something we have yet generally. This is in their nightly builds that they published. But there is a new, very tasty-looking feature in Firefox which you get to from the URL bar by putting in "about:permissions." It is a new UI that shows every site that you have been to. At the top of the list is a global setting for all sites, where you're able to control on a site-by-site basis whether sites are allowed to store passwords, share location, set cookies, open pop-up windows, and maintain offline storage. So this is going to give us very granular privacy-related control once it's in the main Firefox build. At this point it's only available in the nightly build, and apparently it just happened. So I thank him for the tip and the heads-up. And we'll be sure and note when that actually goes out into mainstream.

And finally, I have, in my quest for strange but true new stories of SpinRite success, David J. Sauro just sent this, actually the mail is dated June 1st, so today. He said, "Steve, love the podcast with Leo. No idea of where I should send this, but I figured your sales department would read it and pass it on," as indeed Sue did. "Security Now! got me to investigate SpinRite, and it really is magical. Seriously. I'm sure there's some other coincidence going on here, but you have to hear this story.

"My machine started misbehaving all over the place. And I had heard so many success stories about SpinRite fixing everything from bad sectors to gremlins in systems. When I turned my computer on, it took forever to boot. But that really isn't out of the ordinary. After booting Windows, I realized now I had no network connection, and that my machine could not get an IP from my router. After some testing, I determined that somehow both of my NIC cards" - Network Interface Cards - "had died. Tried many cables, ports, et cetera. Nothing worked. To make sure it was not an OS-specific problem, I booted into my Ubuntu partition and could not get a network connection, either.

"I decided that I had enough and figured, what the hell, I'll run SpinRite. It was my last resort before buying new NICs. I let it run at Level 2 for three hours. SpinRite didn't find or report anything, and I rebooted the machine. Both of my NICs now work. How is this possible? It had to be a symptom of powering down the machine and switching out some internal cables or something. But in my mind, SpinRite can fix broken NICs. And I will be telling all of my friends. Thanks, Steve."

So of course what I think, given his symptoms, is that there were problems with his hard drive. SpinRite often, as we have said, fixes things without reporting anything, so it's just good to run. And then so it was probably a combination of things, one of which was problems on his hard drive, and then maybe an intermittent electrical connection with his NIC. But he got it, and it fixed his machine, so that's all good. So thanks, David, for the report.

Leo: Finally, the big reveal. We're going to finally find out about this new password technique.

Steve: Okay. So I started to create something based on a theory for how to make guaranteed strong passcodes, as I was calling them. We've talked about we have our - the Perfect Paper Passwords is the one-time password system which I created and is open sourced and published and discussed and actually being used a lot. I did a Google the other day for Perfect Paper Passwords, and it's all over the place. And people are actually using it to, you know, they've installed it in servers, and they're using it for their own free one-time password system.

For static passwords, we know that we need passwords which are not obvious, not in dictionaries. And in fact, Leo, I did some looking through password dictionaries, and I got a big kick out of the fact that your prior old password, you mentioned last week you used to use "monkey."

Leo: Yeah. Many moons ago.

Steve: That's #14 on the hit parade.

Leo: So in other words, it would take milliseconds to crack my system.

Steve: Yeah, the monkey, yeah, would be on your back. The #1 most common password is "123456."

Leo: Oh, boy. A lot of people use that.

Steve: And this is from a 35 million user password database that was hacked and escaped from a site, and so this was a statistical analysis of that. The word "password" is fourth in line. And there's some strange things, like "Michael" is on the list. I don't know, I mean, like, tens of thousands of people use "Michael" as their password.

Leo: Probably their name.

Steve: But, well, no, I don't think so. But many more people use "monkey." And if you use your own...

Leo: That's my name.

Steve: That's #14. Okay. So we know that passwords have to be - you cannot use passwords which are going to be on a most common list which the hackers have. Failing that, then they'll use a so-called dictionary attack, literally using a dictionary, often in the language of the user, but sometimes even in foreign languages, the idea being that they want to find your password. They want to figure out what it is. So trying passwords is the only thing they can do. And that's one of the key things, I mean, we all know that. But it sort of forms one of the pillars of something that I realized that I've never in all these years had occur to me or talked about. And from the reactions that I've had to this, no one has, either.

So all the attacker can do is check a password and see if it's right or not. Well, I also created the Perfect Passwords page on GRC which enforces an SSL connection. It'll only come up under SSL because it's using GRC's very good and fully documented on that page pseudo-gibberish generator, pseudorandom number generator, to present users with various forms of ultra-high entropy text. 63 characters long, for example, you're able to copy and paste and drop it into your - use it as a WPA password or drop it into your WiFi, and those things are super-high entropy and thus very useful for passwords.

Leo: And completely unmemorable. And untypeable.

Steve: Oh, my god, yeah. I mean, I've complained many times that my iPod Touch has never been on my own WiFi network because there's no way I can type in my own WPA password. And so what I realized was that, good as that kind of password is, the kind that the Perfect Passwords page gives you, it is not friendly. I mean, it's a non-user-friendly password.

The problem has always been that we've assumed that user-friendly passwords, things we could remember, were probably also weak. And what I hit on is that's not necessarily true. What matters when, as soon as the attacker has exhausted all of his lists, common password lists, maybe site-specific likely passwords based on the site they're trying to hack, or the specific user. You don't want to use your own name as a password because that might be your username also, so it might be that the bad guy knows something about you. Then they'll fall back to dictionaries. Then maybe dictionaries with a digit tacked on the end because we know now that some password policies require at least one digit.

So users who don't really, I don't know what it is, they don't think it's ever going to happen to them, or they're just trying to create a throwaway login because they want to post a comment on a blog and this dumb site requires them to create an account in order to do so, whatever, they'll just tack a zero on the end, or a one, or whatever their favorite digit is. So the bad guys who want to get in will try those tricks, too. So you can imagine there are things that bad guys could do, attackers, to try to figure out something that the lazy user has done.

When all else fails, when all of that fails, they fall back to the traditional, often spoken of, brute-force attack. Because we understand how it's possible to create every possible password, first you start with A, then B, then C, then D and so forth, up through Z. Then AA, AB, AC, AD and so forth up to AZ. Then BA, BB, BC, BD and so on. So it's possible, given time, to run through every possible password. That's why, in the past, we have chosen passwords that are nonmemorable, these horrors like what you get from the Perfect Passwords page at GRC, because they are just absolutely off the map. They're ultra-high entropy, and there is no way to guess what they are.

So what this means is that the only vulnerability after your password isn't going to be quickly found in a list is the bad guy trying them all. Trying them all, since they don't know how long your password is, and the only feedback they get back is yes, that was a match, or no, that wasn't. One of the most often seen lies told by Hollywood is when the cracker uses some algorithm, and one by one determines what the digits of the combination is.

Leo: Yeah, yeah, because...

Steve: Oh, we've got, exactly, oh, locked in another one. Exactly. We all know that isn't the way these things work. If the bad guy guesses wrong, the only information he gets is, that wasn't it, either, sucker. Try again. And the bad guy can try again.

So what occurred to me was that we get our protection from the size of the space that we force the bad guy to search. The larger the space, that is, the greater the number of combinations of wrong passwords the attacker has to try during a brute-force, try-everything attack, the greater security we have. So the first thing that says, the most obvious thing that says is longer is better. The longer it is, the better it is. Because length is one way we know - for example, say that we just had all lowercase. Don't do that, but say that we did because it makes the math easy for a second. Then every time we add one lowercase letter to lengthen our password, we've made it 26 times harder to attack us because, as we all know from the way binary works and even decimal, it's like, if it's easier to think of it in terms of decimal, we just imagine passwords as digits zero through nine. Every time you add a digit, it's 10 times more of them. We understand well how that works in decimal. The same thing works with alphabets and passwords. So longer clearly extends the password search space. And since the attacker doesn't, has no feedback as to the length of our particular password, and they're hoping we've done something dumb and used a small one, they start at the low end. Actually they could only start at the short end because nothing sets the upper end.

Now, if a site had a policy that limited passwords to 16 characters, sure, the attacker could start with a 16-character password and then work their way down. But they're sort of starting at the hard end, at the deep end of the pool rather than the shallow end. So attackers who are in a hurry, trying, because there are so many of these, they've got to do the best job they can, they start at the short end. So long is better.

Okay. The other thing we know is that making the alphabet as large as possible is also important. I use the example, for example, of just decimal. That's, like, the worst you could possible do would be a digits-only password because each character only makes it 10 times stronger. Whereas, for example, if it was lowercase, each character makes it 26 times stronger because you've got an alphabet of 26. But if you put in even just one uppercase character, suddenly now that's radically stronger because the alphabet is lowercase 26 plus uppercase 26, which is to say, by putting in an uppercase character, you have made the attacker use brute forcing that includes uppercase.

Now, think about that for a second. Again, the attacker is going to be as smart as - infinitely smart. We'll give them that. They are the smartest password cracker ever born. So one of the things, one of the other things they know is that 46, I can't remember the exact number, I think it's 46.67 percent of all passwords are all lowercase alphabetic. That's a true statistic, 46.67 percent, all lowercase alpha. So the smart password cracker says, well, if he's going to make me brute force because I haven't been able to discover it so far, and if I'm going to have to try all passwords, I'm first going to try all lowercase passwords because the search is much smaller because the alphabet is only 26 characters. So the bad guy can search some ways out in length using all lowercase. And 46.67 percent of the time, that's where the unwise password will live. Even if it's long, it's still going to be there. So the bad guy's going to try that first before trying any passwords that have uppercase in them, or digits in them, or symbols, special characters like exclamation point, pound sign, so forth.

So the other rule of this, make the search as bad as possible, as hard as possible, in addition to length, which clearly lengthens the search, is you absolutely want to use at least one uppercase, one lowercase, one digit, and one symbol because what you've then done is you've moved your password out of any of the abbreviated searches. Those will all happen first if the attacker is infinitely smart. Finally, with all of those having failed out to whatever length this guy was searching, will try - will have to start brute forcing passwords containing one of each type of symbol, which is the last thing the bad guy wants to do.

Okay. What hit me like a thunderbolt was the password, and on the page where I've documented this I show a, I think it may have been 23 characters of random gibberish that I did get from my own Perfect Passwords page, I just snipped a chunk out, compared to the password D0g and 21 dots.

Leo: 21 dots? What kind of password's that?

Steve: It's stronger than the first one.

Leo: Wow. That's really interesting.

Steve: And isn't that cool. What hit me was entropy doesn't matter. You have to have some. You've got to have enough so that you're not, like, just all dots. That would be bad.

Leo: What you're really saying is that length trumps entropy. In a way.

Steve: As they always said, size matters, Leo

Leo: Size matters.

Steve: Size does matter. And, yes, that's exactly right. Length matters radically more than a password's entropy. As long as your password is not in a dictionary or in a list, then the only way to attack it is brute force. And brute force, trying everything, only gets yes or no. It's either it didn't work or it did.

Leo: It doesn't give him a clue, unlike the movies, as to what's next.

Steve: Yes. He doesn't know you have 21 dots after the word "D0g."

Leo: In fact, he might even know you used dots, but he has to know 20 won't work. 22 won't work.

Steve: Correct. It's got to be an exact match. Now, I am not suggesting...

Leo: Don't use dots.

Steve: ...that people use dots. Nor am I suggesting that it be as simple as that. For example, you could put some dots, some somethings in front, some somethings in the middle, some somethings afterwards. Or, and here's what I would encourage people to do is invent your own scheme.

Leo: But have a pattern that's memorable, it's okay; right?

Steve: Well, and Leo, I immediately changed my WPA WiFi password. You can imagine, now our listeners can imagine what it kind of looks like, except it doesn't matter if you can imagine what it looks like because there's no way to tell. The only attack that we know of against WPA is an offline, brute-force attack, trying them all until you get a packet that has clearly been decrypted.

Leo: So I'm just looking at your brute-force cracker. And "monkey," in the worst-case scenario, would be cracked at .0000000321 seconds.

Steve: Yeah, we should tell our listeners that there is a new service...

Leo: This is such a good service, I love it.

Steve: ...a new page at GRC. You can find it from the main menu under Services. I called it "Password Haystacks" because this is all about hiding your needle in a large haystack. And so what the page does, which is just GRC.com/haystack.htm - but if you leave off the .htm it'll put it on for you - it's a cool little real-time client-side calculator I wrote in JavaScript which you can play with. You can put passwords of various constructions in. And what this does is, this is not a password strength meter. That is, if you put in "monkey," we know that's 14th most common, so that won't survive a second or two.

Leo: But even with brute force it's pretty quickly...

Steve: Even with brute force, it's all lowercase, and it's not very long.

Leo: But if I add 10 dots, it suddenly becomes centuries, thousands of centuries to crack.

Steve: Yes. Yes.

Leo: Now, 10 dots, somebody might guess.

Steve: Yes. If dots became popular, bad guys would start trying dots.

Leo: At first I thought, oh, I'll just put happy faces. But then I realized that's a little too ordered.

Steve: Yeah. I would be creative, use some, like, maybe how about like start with an open bracket, then a couple dashes, then something. But the point is, as you said, Leo, you can even take a simple word, a dictionary word will no longer be found if it's padded. And that's what I call this. I call this "password padding" because this notion is padding a password that is already good enough makes it unbreakable and leaves it memorable. You can invent your own padding methodology. Don't tell anybody. I mean, so that's one of the weaknesses is somebody could see it and go, wow, that's cool. But look, one glance at it, and they would be able to memorize it, unlike one of the nightmare high-entropy passwords.

This is actually a low-entropy password, but it exists in a massive search space so it cannot be found within the lifetime of the universe if it's sufficiently long. And in fact this little Haystack page shows you, makes assumptions, three different types of attack - an online attack, assuming a thousand attempts per second...

Leo: Which is pretty fast, but still.

Steve: Which is, yeah, I wanted to worst-case it. It's probably more like 10 or maybe 100 if you had multiple, if there wasn't a lockout policy and if you had multiple, like instances all trying to guess the password at the same time. And then an offline attack is the one we were talking about, for example, with the WPA password, where you get some data of some sort, and you sit there trying to crack it, trying all possible ones. Or like, for example, we talked about LastPass. The concern there was that users may have had weak master passwords to protect their LastPass database.

Now we know, because your password is hashed locally, you can use one of these large haystack passwords, or a "padded password," as I call it, to come up with something that is incredibly robust, I mean, cannot be cracked. You've got to go to GRC.com/haystack and play around with this because, as you saw, Leo, as long as you've got uppercase, lowercase alpha and a symbol, and you'll probably have plenty of symbols because you'll probably pad them with symbols, but you don't have to. You can also pad them with zeroes, or 10101. I mean, you want to invent your own thing. And then you are really safe. You've got the best of both worlds.

Leo: Now, I'm going to bring up two possible problems.

Steve: Okay.

Leo: One, and we mentioned this also, remember we talked about that page a couple of - the year-old page where the guy said "Dogs are fun" is better than a random password, and we kind of debunked that. But one of the things that was a problem in general with this stuff is, if somebody's looking over your shoulder, and they see you going dot dot dot dot dot dot dot or whatever it is, they might be able to guess that pattern.

Steve: That's true.

Leo: More significant, if you use the same pattern on multiple sites, that's a concern, isn't it, because your pattern might be guessed.

Steve: Yes. And it's why I still have one more trick up my sleeve that I mentioned at the top of the show.

Leo: Of course you do. Of course you do.

Steve: So I would, for example, you wouldn't want to write down what your padding methodology is. You know what it is. The point is you can memorize it. It's easy. It's something you invented, and you add it to passwords yourself. But the reason I gave, specifically, the reason I gave the example of WPA and LastPass is there is no way to compromise the password other than by brute force. That is, it's hashed in your browser, or it's in your router.

But it's exactly as you said, Leo. If a malicious site, well, for example, wasn't storing hashed passwords, if they were storing the password in plaintext, and if they lost control of their database, as we know happens too often, then a bad guy could see your username, which might be your email address, and a low entropy, although unbreakable by brute force, they'd be looking at the plaintext and go, oh, well, I see how this guy forms his passwords, and then go try to log on as you elsewhere. So you still need to have a password which is changing from one site to the other. So our longstanding advice about not reusing passwords on multiple sites stands. But the idea is you can take any good password and make it great, I mean, make it unbreakable from any kind of offline attack. And there is no downside to your use with this kind of password with a WPA WiFi or in LastPass, where it's hashing locally.

Leo: Good point. I still use SuperGenPass, which is a JavaScript-based hashing algorithm that takes a master password, hashes it with your domain name that you're on, and creates a, I think it's a 10-character password. But what I've been doing unconsciously is exactly what you suggest here, which is padding it with four more easily memorable digits, in effect a PIN. So I have a unique password for every site. I have a recognizable, memorable pattern that I add to. Those extra four digits give me a 14-digit password that is mixed characters, upper and lower, and numbers. And it's easy for me to recreate the password. But even if you knew my SuperGen master password, you wouldn't know that I always add these - well, now you do. You wouldn't know that I always add these digits to it.

Steve: Exactly.

Leo: So that's the kind of thing you're talking about.

Steve: Yes. Well, yeah. And with WPA, as far as we know, and it's been looked at very, very closely, there is no attack against it other than brute force. That is the only attack against WPA. And so what struck me is that, so long as you're not worried about your scheme getting loose - and there's no way for it to get loose from WPA or LastPass because that's all being done with you locally. There's some concern if you do this for logins on the 'Net because unfortunately our browsers are not hashing our passwords for us before they leave. That would be really cool, if sites began hashing locally, and so the remote server never had that. But that's not the way things work today. So the downside is the lack of entropy on a long password, that is, a padded password, does, if someone sees it, arranges to see it somehow, it betrays the scheme. But given that an attacker is only doing brute force, a site is hashing passwords, for example, it buys you a huge amount of strength at very little cost.

Leo: Do rainbow tables help, or anything like that?

Steve: No. The rainbow tables are useful, for example, up to a certain size. But again, if you look at GRC.com/haystack - I sound like Dvorak.

Leo: You do.

Steve: If you look at that page, play with it a bit, and you'll see with a deep, what I call a "deep alphabet," meaning at least one of each type of character, so that the brute forcer is forced out of any of the quicker searches, that's an alphabet of 95 characters. And every, I mean, so every character you add makes it 95 times more difficult. Which may not sound like a lot, but that's why I actually - I show the exact number, and then I show it as a power of 10. And then I do the division for the user, dividing by a thousand attacks per second and, what, a hundred billion attacks per second, and a hundred trillion attacks per second. Now, a hundred billion may seem farfetched, except that I just increased it because there was a page I found, actually it was Simon again who pointed me to it, a security researcher has used AMD GPUs and built a homebrew setup that is cracking passwords at 33 billion per second.

Leo: Wow. But, now, they have to have access to the password data in a lump at that point.

Steve: Yes, that's got to be an offline password, yes.

Leo: But that happens. Looks what happened at the PlayStation Network. The Sony password database was hacked and downloaded. And now the attacker has, at their leisure, the time to do this massive brute-force attack.

Steve: So a hundred billion passwords per second is only three times what currently exists right now. And the reason I did the hundred trillion is imagine the NSA, who probably the reason, if there's a graphics card shortage, you know who bought them all. So I just, I wanted to really worst-case the performance of offline hacking and brute forcing and create a calculator that people could use to give them a real-world sense for how strong this is. But again, Leo, if it's long, and you force them into a brute force, which any long password will, because by definition that won't be in a table, the only way they can get it is by looking, by trying them all.

Leo: Ironically, your padding, here's a - I put a random password, a 10-character random password in here which is just numbers, upper and lowercase letters. 2.37 hours in that massive cracking array scenario. But if I add 123456, that very commonly known password, 1.54 hundred thousand centuries.

Steve: Yes.

Leo: And so even - and you can remember 123456. The point being that, yes, by itself, that would be a crappy password. But by adding that to a generated password you're adding a huge amount of security, simply by adding length.

Steve: Exactly, even though it's low entropy.

Leo: It's entropy free.

Steve: Length matters.

Leo: Length matters. What an interesting thing. So everybody then - the takeaway from this would be - somebody asked in the chatroom, well, Steve, you should write a program to do this. No, you don't need a program to do this. This is the whole point.

Steve: Yes, exactly.

Leo: Everybody should, in their mind, come up with whatever pattern they could remember and they could use, and then just tag that on to everything that you create.

Steve: Yeah.

Leo: And it's simple. So we're not saying use that pattern as your sole password. That is important.

Steve: Correct. And so, for example, my WPA password - and I'm so happy, my iPod Touch is now on my WiFi network.

Leo: Woohoo.

Steve: And guests can come over, and I can say, oh, yeah, I mean, I now have a password that I can type in. It's a miracle.

Leo: Yay.

Steve: And it's just as strong as that nightmare that I used to have.

Leo: So how did you generate that password? Did you generate a good, short good password and then tag this pattern that you've invented onto it, or did you not even, I mean, in other words, you still need a base that's unique. Not for WPA, you don't.

Steve: Well, you really don't.

Leo: Not for WPA, you don't.

Steve: Let's say, for example, open bracket, 10 dashes.

Leo: Right.

Steve: And then something and 10 more dashes, and something else and 10 dashes.

Leo: That's long.

Steve: And then close bracket.

Leo: That's as good as your 64-character password; right?

Steve: Oh, absolutely, because even though, I mean, even though WPA accepts 63 characters, it hashes it down to many fewer bits, which means that those are not, all those passwords are not technically unique. There'll be many collisions within the SHA hash which WPA uses.

Leo: So you could make your password be bracket dash dash dash password bracket dash dash dash, and that would be...

Steve: Or dash dash dash bracket closed bracket, yeah.

Leo: Yeah. That would be a nice, long password and easy to remember. Very little entropy there, but very, very hard to brute force.

Steve: Exactly.

Leo: Yeah, yeah. Now, will people start searching for patterns once this becomes widely used?

Steve: The only thing that could be done - the problem is there is, I'm not going to say infinite number of patterns, but...

Leo: There's quite a few.

Steve: ...people are very clever. We've also got forward slash, back slash. We could make little teepees.

Leo: Right. But that's why we don't use smileys or any kind of pattern pattern.

Steve: Yeah, again, remember, it's got to be exact. So we have, during this podcast, we have massively expanded what an attacker would have to do before trying everything. The problem is, it's truly a massive expansion because all we've said is "make it longer." You could put it in the beginning, in the middle, in the end. And you can put anything, anything you want in the beginning, the middle, the end. Or, as I did for my own WPA, put a couple special things spaced out among some other numbers of something else, and just no one's going to find it. There's just too much uncertainty there. And the only way to find it is try them all. And GRC.com/haystack will just show you exactly how many you've created. And what's surprising is that they do add up so fast. As you said, Leo, just adding 123456, the most common password in the world, hugely strengthens a password that could otherwise be broken with a massive array in an hour or two. Now it's trillions of centuries.

Leo: Yeah, every effective. Really cool. And a great insight. And that's why I love this show. So if you can't remember the URL, if you go to GRC.com, surely you can remember that, it's in the Services menu, Password Haystacks. Just go to there. Actually, while you're there you should browse around. There's so much great stuff. There's of course the Security Now! podcast. And there's 200, I'm sorry, 303 of them now, including 16KB versions for the bandwidth impaired. Full transcripts, too, for those who like to read.

A lot of people in the chatroom have been asking about passwords in general, and LastPass specifically. This is a great time to go back in time at that website. You can read the transcripts of Steve's discussion of LastPass, Steve's discussion of passwords. And it's just a great resource for learning. Of course he also offers a lot of other great free tools, including the Haystack. It's all at GRC.com. And while you're there, please buy ShieldsUP!. I mean SpinRite. Don't buy ShieldsUP!, it's free. Please buy SpinRite, the world's best hard drive maintenance and recovery utility. You want this thing. If you've got a hard drive, you need SpinRite. And it keeps Steve in quinti venti lattes.

Steve: Well, it allows me the opportunity to come up with these kinds of things.

Leo: Yeah, exactly.

Steve: It gives me the chance to think about this. And I've got one more idea. I don't know how it's going to work out. I've got to do some cryptanalysis, believe it or not. But so, I mean, it doesn't obsolete this. This is good in the meantime. But this journey has opened up some other ideas. So I'm not going to tease everybody. I don't know when it's going to be. But we'll have a Q&A next week, and then I'm sure some great content the week after. But I've got one more thing, I think, up my sleeve.

Leo: Oh, there's more than one more thing up Steve Gibson's sleeve, I can guarantee you that. Steve's on Twitter @SGgrc. You might want to follow him there. And of course we do the show every Wednesday, 11:00 a.m. Pacific, 2:00 p.m. Eastern, 1800 UTC at live.twit.tv. So you can watch us do it live if you want to get the latest, freshest version. But if you miss it, hey, and I understand, people actually have jobs and things like that, you can always catch it later at TWiT.tv/sn or on Steve's site, GRC.com. Steve, thanks so much. This is great.

Steve: Thanks, Leo.

Leo: I'm going to go change all my passwords.

Steve: Talk to you next week.

Leo: See you next time on Security Now!.