Password Policies

Description: After catching up with a bunch of interesting news, Steve and Leo examine a terrific piece of research performed by Dashlane, makers of a password manager. They have researched and presented the current state of the top 100 web retailers' password policies. Fascinating!

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-441.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-441-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We're going to talk about the top 100 retailers and their password policies. You'd be amazed at some of the easygoing password policies, low-security password policies some of these online stores have. We also have all the security news for you. Steve Gibson's next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 441, recorded February 4th, 2014: Password Policies.

It's time for Security Now!, the show that protects you, your loved ones, your privacy, all the stuff you need to stay safe online with this guy, the Explainer in Cheap - Chief, not Cheap.

Steve Gibson: Yeah, I'm the Explainer in Cheap.

Leo: Cheap. You're the Cheap Explainer. Well, it's a free show. I guess he is the Explainer in Cheap. We could charge for this stuff, you know. Mr. Steven "Tiberius" Gibson. He is, of course, the man in charge of GRC.com, the Gibson Research Corporation, has written many really useful tools. You've probably heard of a great many of them, including ShieldsUP!; and his bread-and-butter is SpinRite, the world's finest hard drive maintenance utility. It's time for a show, no questions, just answers show.

Steve: Correct.

Leo: Right?

Steve: Correct. What came to my attention last week, and I mentioned it at the end of, I think, the podcast last week was a study that was done by a company, Dashlane, who make a password management utility. Everyone of course is familiar with those. Everyone knows my favorite is LastPass. I want to make sure we give Dashlane credit for this research and let people know that they're there because we're taking advantage of their research, and I don't want to - I want to give them their due.

So what they did was they performed a set of tests to essentially reverse-engineer, from interacting just with the web interface of the industry's current top 100 web retailers, which they obtained from some other list - I mean, it's the who's who of the Internet. Amazon's there, Apple's there, Microsoft, I mean, we'll be talking about these names toward the end of the podcast because they reverse-engineered the login policies of this top 100 set of retailers and found some really interesting things. Like I was saying to you before we began, before you hit the Record button, there are some really curious things. They looked at the minimum password acceptable. And there are some sites whose minimum password is one.

Leo: What?

Steve: Well, first of all...

Leo: That's no password; right?

Steve: That's understandable because less than that would be none. And so basically they're saying did you leave the password field blank or not?

Leo: Right, right.

Steve: But there are also some that have a minimum length of three. And it's like, wait, wait. So someone wrote code to see whether you had a password of one or two characters, and that was not okay, but three was? Anyway, some really interesting things come out of this. This is a problem because there's a spreadsheet that's online, which I now understand is a Google spreadsheet. You just explained to me that you can't an infinite number of people looking at it. But I created a bit.ly shortcut, bit.ly/sn441, which is the number of the podcast, sheet, s-h-e-e-t [bit.ly/sn441sheet].

So our listeners, who will be listening at various times of day, will probably be able to bring this up because I doubt that there'll be more than a hundred at any one time. But this is the raw data from Dashlane's study, which we'll be looking at later on in the podcast, which is - and I've got all kinds of summaries and things that I've already established from that, which is really just fascinating because, for example, is there a limit to the number of times you can attempt to log in, and what is that limit, and so forth.

So, neat analysis that we're going to go into on today's Password Policies. I initially put in

2014, but that would make the title too long. So I thought, okay, well, we'll just leave it as Password Policies. And some interesting news. So this week on Security Now! we've got a new version of Firefox we'll be looking at. An emergency update to Flash from Adobe...

Leo: Yeah, I saw that one.

Steve: ...just happened today, and it's important for Windows PCs, Macs, and even Linux because this was a zero-day flaw which allows code to be executed in your machine if you let Flash run. So, bad news.

Leo: God.

Steve: Also, something picked up on the return of HoneyWords. We talked about honey passwords, shortened to honey words, I think, last summer when this happened. But for some reason they sort of came back. So I thought I would just mention them briefly. We're seeing a troubling rise in malicious ads. I found a very interesting Windows Firewall add-on that I want to talk about. And my own research brought me to why tossing a coin is not fair, which we'll talk about.

Leo: Really.

Steve: And did the NSA ask me to build a backdoor into CryptoLink? Someone did a blog posting positing that maybe that had happened. And much more, this week. So lots of fun stuff.

Leo: Wow. Boy, I would expect you would mention it if it had, but...

Steve: So, well, actually I said that I would simply stop talking about it. And I did stop talking about it, about CryptoLink. So...

Leo: Whoa. All right, well, shhhh. Whoa. Never Say Anything, NSA.

Steve: Today. Oh, what?

Leo: Go ahead.

Steve: I was going to say today we got a new version of Firefox, v27. And one of the things that they commented on was that TLS v1.2 has been implemented. Now, other people, in response to last week's comments about the HowsMySSL.com site, h-o-w-s-m-y-s-s-l dotcom, they commented that, well, in 26 you could turn it on, but it wasn't on by default. So when I updated to 27, which just happened hours ago, and I went to HowsMySSL, it was a little happier. Last week I got "bad" because it had two problems

with my v26. Well, then I got "improvable," I think it was, and it was still complaining that I was using TLS v1.1. So I thought, huh? What?

So anyway, what happened was - and to our listeners, who may also be affected by this, I went into - the way you go into Firefox's config is about:config you stick up in the URL address bar. That takes you to this overwhelming number of things you can tweak. And then in the search area you put in TLS. And that brings it down to a manageable eight or so things that involve TLS. Well, sure enough, it turns out that sometime in the past I had overridden Firefox's then-conservative max level, pushing it forward; but now it was pushed back because Firefox had been marching forward, and my override was now behind the times. It used to be setting the TLS max version higher than it was the default. But it was sticky, so now it was setting it lower than the default.

So if you right-click - and you can see that any overrides will be in bold in the Firefox about:config screen after you've whittled it down. So I just right-clicked and hit Reset, and it removed the highlighting, allowing v27, which is today's, to now run at TLS v1.2, where it wants to. And sure enough, HowsMySSL now says "probably okay," because remember it's pretty stingy about - it doesn't get very excited about even when everything is perfect. It's like, eh, it's probably okay. So that's the best you can get. However, they did, with v27 from last week, remove that one pesky 3DES SSL cipher. I remember I made you laugh, Leo, because the comment was "This cipher was meant to die with SSL 3.0 and is of unknown safety." It's like, eh, okay, fine. So anyway, that's gone now.

So Firefox have brought themselves up to the "probably okay" status with HowsMySSL, which is the best anybody can get. And nothing much else got fixed. They are continuing to move forward towards improving standard support and interoperability. So I dove into the, you know, what do the developers think, what do the webmasters think. And there were, like, all kinds of fixes down in the details of this got added and this and that now being supported and this we removed and blah blah blah. But no huge security changes or anything, just moving us forward.

And also taking TLS 1.2, making that now the default, rather than, as was the case before today, you had to, like, push it. It was there, but you had to turn it on on purpose. Now they said, okay, fine, it seems to be working. We'll let everybody use it So this is good news, that we're now at 1.2 across the board with our major browsers. It's been a while coming. The standard's been available for quite a while. But now that we're seeing servers supporting it and browsers are supporting it, some of those pesky edge cases where there were theoretical attacks that worried people, those are pretty much falling by the wayside. And that's good.

So unless you're using the latest version of IE, that is, IE 11, or Chrome, both of which will be immediately made aware automatically that Adobe has a newer version of Flash, you really do need to update. If you're running any kind of Flash blocker, as of course everyone is who's using NoScript, then you're probably more okay. The reason this is a problem is that an integer underflow bug has been found. This was detected by the guys at Kaspersky, who found it being used in the wild. So they, in doing their research, their malware research, they found instances where people were running - going to a page that ran Flash, and they were getting malware installed just by visiting that page that had Flash-enabled content. I don't know whether this was ads or something from the page.

Actually one of the things I want to talk about here in a second is a troubling increase in the level of malware we're seeing in ads. And of course ads are often delivering Flash content, to the annoyance of web surfers everywhere. But Adobe's got an update that

they pushed out under emergency terms to deal with what is this zero-day flaw. So it's important for people whose browsers are not updating themselves automatically to go get that. You really should either have, I mean, it's hard to work without Flash. It's still something that people are using and depending upon. There's a site that I like, NutritionData.Self.com, which is really a nice way to research the content of stuff that you eat. But all of their little charts require Flash. So I'm often wanting to do it on an iOS device, which famously doesn't support Flash. So I can see some of the stats, but not the little charts. That's sort of annoying.

Leo: We were talking - because we were talking about the exploit. We were talking about it on MacBreak Weekly. And who was it, Alex or Aaron Hillegass, one of the guests said, you know, they've been very successful in recoding everything in HTML 5. No problems. They've solved all the latency issues. Just - Flash should just die.

Steve: It really should.

Leo: There's no reason to use it anymore.

Steve: Yup.

Leo: And it locks you out of every iOS device. So besides the security issue, you're losing a big portion of the mobile audience. Just don't use it.

Steve: And of course the mobile audience is a growing portion of the audience.

Leo: Yeah. I presume the site you're talking about is from Self magazine, and you'd think they would care about that kind of stuff.

Steve: Yeah.

Leo: But they've already coded it, and they don't want to recode it, and...

Steve: Yes. Yeah. And in fact it's interesting, I moved all of my site's videos, like the SpinRite demo and other stuff, over to HTML5 some time ago. But the PDP-8 videos that I did years back I still haven't done. And I did the work to create a really nice set of files because actually there's three different formats you want. There's MP4, WebM, and OGV. And I came up with some simple batch files that I use that run FFmpeg to, like, automatically take an existing video in any format you have and re-transcribe it into - transcode it, that was the word I was looking for, transcode it into each of those three at an acceptably low bitrate and converts the audio and does all that stuff. And I just thought I really ought to share this with the world because it took a while to come up with all of the - if you've ever seen a command line of FFmpeg, it's like, oh, my goodness.

Leo: We do because every show we do is encoded with FFmpeg, and Russell Tammany has written a bunch of scripts using very, very long FFmpeg command lines. Holy moly, yeah.

Steve: So Honey Passwords, I guess it was Net-Security.org made a posting about this which Gizmodo picked up on. And when Gizmodo picks up on it, everybody starts tweeting me the link because it's like, oh, my god, you know. And when I saw it coming in, I'm thinking, wait a minute, this isn't news. This was, like, last summer. And so I checked to make sure that it wasn't anything new. And sure enough, this was interesting research by two guys whom we talked about, Ari Juels from RSA Labs, and of course Ron Rivest is the "R" of RSA, who's now at MIT. And this was HoneyWords, or sort of the honeypot passwords concept. The idea was that you would salt your web server's database with a bunch of fake user accounts with easily crackable hashes of passwords, like "password" and "123456." That is, you would pretend to be really - okay, I almost said "stupid," but I didn't. I didn't.

Leo: Silly.

Steve: Users who would have those really poor passwords. But you would tie them into a sentinel on your server. And the idea was that it would sort of be like a canary in the coal mine, where if anyone tried to log into these fake accounts using really easily crackable passwords, that would let you know that your database somehow had been exfiltrated from your control.

Leo: Clever.

Steve: Yeah, it is. And so this paper was published on May 2nd of 2013. Bruce Schneier blogged about it four days later, saying: "Here is a simple but clever idea. Seed password files with dummy entries that will trigger an alarm when used. That way a site can know when a hacker is trying to decrypt the password file." So since today is all about passwords, I thought, okay.

Leo: That's a good idea.

Steve: I'll acknowledge the fact that I did see all those tweets from people who picked up on Gizmodo's delayed announcement of something that we talked about some months ago.

Leo: It really does happen that, for some reason, lately a lot, blogs come back to stories they published months ago. And people don't look - or the worst thing is I see tweets from stories of years ago, and people actually - oh, I just read this. You've got to read the date.

Steve: Read the date.

Leo: Yeah, always a good idea. And go to the original source is the other thing I do. Very important. Go to the original. Follow the back tracks to the original story. And then you'll see, oh, yeah, this is ancient.

Steve: And I want the original source because I don't want the digested version.

Leo: Right, the interpretation, yes.

Steve: Yes. I want to go and get, you know, there's typically much more interesting information there. So the Wall Street Journal did a piece where they talked about the growing problem of malware in ads. And of course Yahoo! famously got infected, about maybe a couple weeks ago. I don't think I covered it, just because we just - we've been running out of time on the podcast, and I had to just say, okay, wait, there's just not enough time to talk about this. And there really wasn't much to say except that Yahoo! had been serving an ad that had been infecting people for some time. And I found some stats that I thought were interesting and a little troubling. RiskIQ, Inc. had been tracking malicious ads. In 2011 they found on the order of 70,000 of them. A year later...

Leo: What?

Steve: Oh, yeah. A year later, in 2012, 205,000. And last year, 384,000...

Leo: Oh, my god.

Steve: ...of them.

Leo: Now, what does that mean? Individual - I don't - what's the count of?

Steve: Yeah, that's a good question.

Leo: Online, I see it somewhere online thing?

Steve: Right. But they're unique instances of ads.

Leo: That's crazy.

Steve: Yeah. And Google in 2012 disabled ads from more than 123,000 sites. But a year later, across 2013, disabled ads from more than 400,000 sites.

Leo: Holy moly.

Steve: And so, finally, the last stat here is that the problem here is that - and we talked about this, for example, with CryptoLocker, where the older versions were now being caught by pretty much all of the AV. But those samples that I posted, when I posted them two weeks ago, they were only being seen by four or five out of 47. Well, these recent malicious ads are being missed by 44 out of 47 AV programs over on VirusTotal.

Leo: Now, I don't want to encourage this. But if you used AdBlock, you wouldn't be vulnerable to them; right?

Steve: Correct. These things, I mean, first of all, the malicious ad probably needs execution privileges. That is, it needs scripting or, as we were just talking about Flash, it needs Flash. I say "probably" because we have seen just static image vulnerabilities, where the...

Leo: Malformed JPGs, for instance.

Steve: Correctly, exactly. We have seen malformed JPGs - a JPG, a PNG, a whatever - that are being rendered in your machine. The bad guys cleverly find a rendering flaw that is able to - where they're able to, like, actually put code in the image. The image won't display, typically, but it'll crash the renderer. But in the process it runs code that they've also provided. Those are rare. What's more typical is that the ad actually is containing scripting and will run JavaScript or, god help you, Java itself, and take over your machine, execute the attacker's code. So really just - you could still show ads if you just didn't let them run script. And of course NoScript will do that for you. It solves that problem.

Leo: I have no ethical issue with doing that, by the way.

Steve: Right. The other thing that's cool, we talked about how Google is starting to render the images for you, which is a huge win. So rather than your page showing ads that your browser has fetched directly from the source of the image, if Google has provided the page, or if you're in Gmail, and that's typically where it's going to happen, you're looking at an HTML page with ads in it. Google will pull the image and render it essentially on its own server, then take a picture of it and sort of re-render it. And we've talked about the idea of transcoding the image. For example, Opera famously does that in order to reduce the size of overly large images for their mobile platform. When you're using Opera mobile, you actually get a good speed improvement because your browser's not having to go out and grab overly large images. So that's also going to inherently clean up your advertising images and prevent it from being a problem.

But I guess - this just sort of came across my radar, and I wanted to take a minute to say that, I mean, it makes sense that this would be a new vector because, if bad guys are able to purchase ads and get them into major ad feed streams, and if those ads bypass the AV - so even a web advertiser that is doing a good job of protecting, trying to protect their feed from malware, if ads are staying ahead of AV detection, they'll go

through. So it's uncommon that a static image is going to have a problem. So this is further reason not to let ads execute code. And you see them where they're, like, amazing animations in these ads, where fishes are swimming around, and bubbles are blowing, and the seaweed is undulating. And it's funny because I only see that when I look at other people's browsers. My browsers all have NoScript installed. And so I see ads, but I see their fallback image, which is, for one thing, way less distracting than these crazy Flash ads.

Leo: It's really, the other point to be made is that it is the ad networks that are really at fault here because they allow people to put an ad up unchecked. Here's a hundred dollars. Take my ad. And they don't check the code. They don't check the ad. They don't make sure everything's okay. And they're the ones who are propagating these viruses. I really don't understand why the ad networks aren't getting more heat from this because it's them - the Googles, the Facebooks, and all the other ones that are supplying these ads - that are making this possible.

Steve: Yeah. There was one piece I read following some links that talked about how, in some cases, when the companies behind the malicious, like, discovered malicious ads, they tried to backtrack. Not surprisingly, they were dead ends. They were like nonexistent companies.

Leo: Yeah, of course. So it might be easy to check that.

Steve: Precisely. Do it before you host their ad on hundreds of thousands of people's web browsers.

Leo: The problem is that these ad networks really have completely automated ways of buying an ad. Go on Facebook, and you can buy an ad directly, have it start showing up in people's pages right now, in seconds. And that shouldn't - they make it too easy, in other words.

Steve: Right. And so what they've done is they've essentially created a malware anonymizing service that allows...

Leo: Yes. Yes. Why don't they get more heat for that?

Steve: And a malware distribution...

Leo: We've got a malware distribution network here.

Steve: Yeah.

Leo: So you're going to legitimate sites. The site's safe, but the ad on the site isn't.

Steve: Precisely.

Leo: I think they really should get more heat for that. And frankly, it's going to kill their business because what do people do? They run AdBlock.

Steve: Yeah.

Leo: And I can't really blame people for running AdBlock if they're worried that ads are going to put malware on their system.

Steve: You know, AdBlock must allow some things through.

Leo: Oh, they do.

Steve: Because I was going to say, I...

Leo: You can make a deal with AdBlock to be a premium ad client that AdBlock will not block. So there's different - there's two different - there's AdBlock and Adblock Plus. And one is less good than the other, and I'm not sure which.

Steve: I have ABP on my Firefox. And I see ads. I just don't see really - I don't see things skipping across my screen and really bothering me.

Leo: Yeah. Adblock Plus was kind of - ABP was, for a while, I thought, the nonprofit kind of. But apparently - remember we saw that story that they were offering people...

Steve: And there is a checkbox at the bottom of the config screen. Let's see, configure...

Leo: It says: "How do we make money? We are being paid by some larger properties that serve nonintrusive advertisements." They have the Acceptable Ads Initiative.

Steve: Yeah, and, see, again, I'm kind of okay with that because...

Leo: I think it's extortion.

Steve: They're doing some curating so that...

Leo: Yeah. You know how they curate? Give me some money. I think it's extortion. They're saying, well, you know, you want people to see your ads, you'd better give us some money. "Extortion" is not the right word.

Steve: But you also can't have, hey, I've got color TV, you know, shouting out of your ad.

Leo: Well, they say that.

Steve: My point is, my experience is...

Leo: Yeah, they do do that, yeah.

Steve: Yeah. I'm never - I see other people's browsers, and I think, oh, my goodness, how do you sit in front of that?

Leo: Oh, yeah, it's garbage.

Steve: And mine's just very sedate.

Leo: And that's the other, I mean, so here's - I've mentioned my position before. But you're using a free site that is ad supported. By not looking at the ads, you're really cheating. You're saying, well, I'm not going to see the ads, but I still want the free service. So the ethical way would be not go to those sites that have really annoying ads, or see the ads. But to block them and go to the site is kind of, to me, it's clearly unethical. However, they're bringing this on themselves by having intrusive ads, malware-based ads. And I can't really blame somebody for wanting to block that.

Steve: Well, and the perfect compromise is just make sure you've got scripting shut down because I don't want to say it can't happen, because we know it can. You can have a malicious static image. But you're 99.999% safe if you simply block scripting because that's the way the malicious ads are being served. And if you block scripting, you'll still see their backup, static content. You just won't see - it won't look like a circus that's happening.

Leo: Yeah. It really is. Some of this stuff is terrible.

Steve: Oh, god. It's just awful. Okay. So I have the link here in the show notes, Leo, under Intelligence Law Blog. And this was, I guess, it says 2013/12, so it must have been December of last year that this - it's a very well-written blog posting. And I won't read the whole thing. I'm just going to say that this blogger, who's a listener, he's a fan of the show, says - he asks the question, "Did the NSA or FBI Threaten Steve Gibson to Force

Him to Abandon CryptoLink?" And he runs through the timeline where he explains how excited I was, and I was talking about it all the time and getting trademarks and buying domain names. And in fact I did buy CryptoLink.com. It's the only domain name I think I've ever purchased from someone. It was a chunk of change, actually, because I was like, this was - I was going to make this a commercial product, and I wanted CryptoLink.com, and so I got that.

And then he says, at one point, he says that I was speaking of CryptoLink often and enthusiastically and once said: "If you ever hear of me or learn of me discontinuing CryptoLink for no reason, you'll know that for whatever reason I felt no longer able to offer something whose security I could put my reputation behind because I'll kill it before I would compromise it." And then he goes on to say, like on some, like, last podcast...

Leo: March 4th, 2010 you spoke those fateful words.

Steve: Yup, and then never mentioned it again. And of course it's interesting because I did just mention it a couple weeks ago, but of course that's not fair. This blog post was more than a few weeks ago. It was eight weeks ago or six or something. Because I explained that - and I guess at some point I remember saying that. I did deliberately say, okay, I can't do this because - and we talked about it at the time. I was reading the handwriting on the wall, even pre-Snowden, that our government seemed to be really uncomfortable with crypto technology on the Internet that it didn't have a backdoor to. And my concern was that I would invest a year of my time developing CryptoLink to be commercial and then have someone approach me and say, "Steve, we need access to your customers' use of this tool." And there's no way I could allow that. I mean, in the same way that Ladar and others have said no, this is not okay.

And so I stopped working on it because I was concerned that was going to happen. And then of course we've had all of the NSA revelations of the past year, since Snowden, and so forth. So my current plan that I...

Leo: So just to be clear, you did that preemptively. It wasn't that you were approached.

Steve: Correct.

Leo: You did it because you figured eventually you would be approached, and you just didn't want to be in the position of doing a crypto solution.

Steve: Well, I didn't want to be in a position of having invested a year of my time in something that I then had to kill.

Leo: But just to be clear, the feds have not contacted you, did not contact you.

Steve: They have not. Absolutely have not, and never have.

Leo: Okay. And there's nothing to contact you over at this point.

Steve: And probably I couldn't say it. I wouldn't answer the question.

Leo: It depends on how they contact you. I mean, if they walked up to you, as they did to that woman at the security conference, at the RSA Conference, and said, would you mind if we put a backdoor in there, on mic? That's not - you can tell people that that happened. If they send you a National Security Letter, then it's a different...

Steve: And so my current plan is to do CryptoLink as freeware in the future.

Leo: Okay.

Steve: And then I feel different about it. If I always intend to make it free, then it's, first of all, and for some reason I can't, then okay, well...

Leo: No harm, no foul, yeah.

Steve: Yeah, exactly. So that's the plan. And by the way, I keep meaning to get to this. Is there something called Freelan.org? Can you put that in real quick, Leo?

Leo: I would check. Yeah, that sounds familiar.

Steve: Yeah, it's good. It's techie. It's not easy to use. And it looks like they did everything right. Yes, there it is. Someone's tweeted it to me, and I think I've got it in notes somewhere so that I can give them credit.

Leo: So this would be like a Hamachi replacement?

Steve: Yes. This is a very much done right, TNO, Hamachi replacement. It's not for the faint of heart. But it will allow you - it is cross-platform, Windows, Mac, and Linux. And it will allow you to build Hamachi-style private networks, like in your own IP space. You wouldn't want to use five-dot anymore. And I think they use nine-dot, which is owned by IBM. There was another one that I thought would be a better choice. But anyway, I wanted to just - this is completely sort of - I'm not prepared to do a full presentation about it. But I have spent some time digging into it. And it's very nice looking. So Freelan.org. And you can either do client-server or a peer-to-peer network or a hybrid, very much like Hamachi. And all of the crypto looks very correctly done.

Leo: Neat. Freelan, F-r-e-e-l-a-n dot o-r-g.

Steve: Yeah. So Windows users who are also a little more on the techie side, I ran across something called Windows Firewall Control, which is free. It's Bini, B-i-n-i, Soft.org [BiniSoft.org]. And they only - they have two things that they offer. One of them is unregistratable. Registratable? Registrationable? Anyway. Registerable, I guess, yeah. And that's something that allows...

Leo: I don't know.

Steve: Can't register it? Anyway, registrate...

Leo: Registerable. Registrable.

Steve: Register.

Leo: Ask Dr. Mom. She knows all that stuff.

Steve: You do not need to register, even to get...

Leo: Yes, that's right. Circumlocute, that's always the solution.

Steve: I hate adding stuff to my machine. But if it's there already, and I can tweak it, I'm much happier. Well, we've had Windows Firewall, famously, since XP. Initially it wasn't on by default, then they turned it on in SP2, and they've continued to add features to it to make it more powerful ever since. Windows Firewall Control is a very nifty add-on - I think you need Windows 7, doesn't support XP, but it runs from Windows 7 or Vista or 8 or 8.1, and also Server 2008 - that gives you control. Just from their description: "Windows Firewall Control is a nifty little application which extends the functionality of the Windows Firewall" - which, and this is my point, is already there in your machine - "and provides quick access to the most frequent options of Windows Firewall. It runs in the system tray and allows users to control the native firewall easily, without having to waste time by navigating to the specific part of the firewall." Believe me, it is a - it's hard to get in there. Microsoft doesn't make it easy to find.

So, for example, they say: "Windows Firewall Control offers four filtering modes which can be switched with just a single mouse click." High filtering, suddenly all outbound and inbound connections are blocked. This setting blocks all attempts to connect to and from your computer. Which is kind of cool. Sometimes you might want to do that.

Someone was asking the other day if there was - they wanted to run, I don't remember what it was, I think it was the Off The Grid, my Off The Grid password generator based on Latin Squares. They loved it, but they were trying to run it on a disconnected machine, and I ended up suggesting, well, you could do it in a VM. They just liked the idea of generating - of running the randomness without any access to GRC because I think GRC helps, it provides some of the entropy seeding of the ultra-high entropy random number generator that I developed for the Off The Grid thing.

But here you could just switch on high filtering to essentially disconnect yourself from the

Internet for some period of time. Medium filtering says outbound connections that do not match a rule are blocked. Only the programs that you allow can initiate outbound connections. And that's cool. So this is bringing back this notion of famously ZoneAlarm-style blocking, where you are able to control specifically which applications have access to the Internet. Low filtering, outbound connections that do not match a rule are allowed. The user can block the programs he doesn't want to initiate outbound connections. Or no filtering, Windows Firewall is turned off. It says avoid using this setting unless you have another firewall running on your computer. And then they've just got a ton of really interesting features. I won't go through them all at length. But I'm impressed.

So if this sounds interesting to you, there's just, like, all kinds of additional things that they've added to it that really impress me. And for a $10 donation, which you can make through PayPal, you get with registering this a notification system, where you can get notification, for example, of things that are trying to access the Internet that you haven't been made aware of. And of course, famously, I was beta testing ZoneAlarm, and that's how I discovered that I had the ad alert, or the adware stuff, the Radiate and Aureate, they were calling themselves at the time, spyware in my machine.

So anyway, I'm impressed with this little app. I'm not running it yet because I'm on XP. But anyone who's 7 or later can. And again, it just runs with the existing firewall, which I like a lot better than installing a third-party firewall into an existing system, since there's a perfectly good one already present.

And they do, by the way, for free, also make a USB flash drive controller, which enables you just to do a complete enable and disable of all USB flash drives on your machine. You can deny execution from the flash drive to protect yourself from anything running off of it. Many people just use their flash drives for data, and they want to be protected from, example, malware that might get into their system that way. And this could do that. Famously, probably, Stuxnet would have never been able to operate had this been present. You can also cause them to be mounted read-only, so your system can never write to flash drives, which could also come in handy. And that's a little freebie from the same people, which I think is neat.

Leo: Cool, yeah.

Steve: Okay. What I think we're starting to see is another - it's been described as a "land grab" by malware furiously going after point-of-sale terminals. Famously, of course, we've been talking for the last few weeks about the Target infiltration of malware into their Windows XP Embedded POS terminals, which ended up causing 110 million people's credit card data to be exfiltrated from Target over some period of time. Then Neiman Marcus has now confessed a similar breach. Just this morning they were in front of a Senate hearing, and Reuters was tweeting live from the hearing. Apparently their CIO said that a maximum of 1.1 million accounts were potentially exposed to malware in a data breach that they suffered, but probably somewhat less than that. And this occurred, they're saying, during transactions at 77 of 85 Neiman Marcus stores between July and October of last year, of 2013.

Leo: Boy, these point-of-sale systems really are a target, aren't they.

Steve: Yes. And that's what...

Leo: They're so vulnerable.

Steve: Yes. I think that's what is going to happen. In fact, I jumped on TNT yesterday morning to talk about this briefly with Mike and company because there was some additional news that had just surfaced. The RSA has uncovered a botnet which they're calling "Chewbacca." I don't know why.

Leo: [Wookiee sounds]

Steve: But it's a keylogging and memory-scraping botnet which they have acquired the binary and reverse-engineered the binary. They discovered that it uses Tor hidden servers. So there is a Tor client in the binary which goes through the Tor network to hide the server that it is contacting. And it's exfiltrating this user credit card and PIN and other transaction date from point-of-sale terminals wherever this is installed. It puts itself in the startup folder. It again takes advantage of the fact that most of these systems are running Windows XP, the embedded version. And unfortunately, in 62 days, that'll no longer be getting any security updates.

Leo: No, Embedded goes longer. I think Embedded...

Steve: Oh, it does. Oh, you're right, it does. Yes, yes, yes. But I wonder whether Embedded XP running out in some little terminal, is it getting updates?

Leo: Yeah, how is it being updated? Yeah, yeah.

Steve: I'd be surprised, actually, if it were pulling security updates because they probably figure, well, no one's able to browse anywhere, so why keep it current?

Leo: Oh. Oh. Lordy, lordy.

Steve: So I think what we're seeing is a new, unfortunately, a new and very vulnerable attack vector. Think of all the little chains and stores that are...

Leo: And they all use the same software; right? I mean, that's part of the problem.

Steve: Yup. There's a low number of original sources of this, and they're all leasing it or purchasing it, and it's probably got the same, and it's probably just old crap that's in there because it's like, eh, you know, hey, it works. Why update it? Well, why, indeed?

Okay. So this is completely random, or that's actually a bad pun because I'm now working, as I'll explain for a different reason in a minute, on the user interface side of the SQRL project. It's been coming along beautifully. And I'm now working on what the user sees as they use SQRL. And I'm doing this just prior to starting to write code because I'm

at the point now where we're ready to write code. And I was spending some time on the problem of generating entropy because one of the things that SQRL needs, as everything does that's doing crypto, is a really, really, really, really, really good source of randomness. And it's easy to say that. But we've talked about, for example, the bizarreness of the discovery that SSL certificates are using the same private key because they're being generated by UNIX systems or Linux systems which are turned on and immediately asked to generate a key.

Well, if you turn the same systems, the same version of Linux or UNIX on and immediately ask it to generate a key, it hasn't had a chance to get much randomness because they're all starting from the absolutely same initial condition when they boot. And so it turns out that enough companies did that, that they generated the same random keys for their SSL certificates. No one knew until the EFF started looking at all of the random keys and found, surprisingly, collisions between them. So that's an example of where we think we have something random, but we actually don't. And in a system like SQRL, we don't want to make that kind of mistake.

So I was brainstorming sources of randomness. And I've already talked about some that I think are kind of clever, like having the user turn their camera on and just shake it and point it around. And while SQRL is streaming the video in and just absorbing all of that absolutely high level of random noise that would go into a random pool in order to generate the user's single unique identity, which they can potentially then maintain for the rest of their lives and identify themselves uniquely on the Internet. So I was thinking about this. I just thought, well, you know, what about a coin toss? What about giving - again, I'm working on the user interface side. So one option would be, okay, get a coin. And we're going to set the counter to zero.

Leo: All right.

Steve: And you toss the coin. When it lands, you press heads or tails. Okay? That's one. Now you do it again. And if you did it 256 times, you would generate a 256-bit absolutely incredibly high entropy random number.

Leo: Right.

Steve: Or would you?

Leo: A heads is one and a tails is zero.

Steve: Exactly.

Leo: Yeah.

Steve: And 50/50 chance. And so wouldn't that be random?

Leo: Yeah.

Steve: So I thought, you know, I'd better make sure that that would be random before I suggest that to anybody. Turns out it's not.

Leo: What?

Steve: I found some interesting analysis. The guys, there's a group called "statweb" at Stanford that published a 31-page paper of physics that is the "Dynamical Bias in the Coin Toss." And it's a fascinating read. The show notes, which of course now we're publishing, the links are in the show notes. But their abstract on this 31-page paper reads: "We analyze the natural process of flipping a coin which is caught in the hand. We prove that vigorously flipped coins..."

Leo: Vigorously flipped.

Steve: Oh, yeah. And it matters. We have some takeaways here that I'll cover in a second. But "...vigorously flipped coins are biased to come up the same way they started."

Leo: Ah. Biased by how much, though?

Steve: Well, but, you know, I need...

Leo: Any bias is nonrandom, so...

Steve: Oh, exactly. "The amount of bias depends on a single parameter, the angle between the normal to the coin and the angular momentum vector."

Leo: Well.

Steve: And so I did say I was going to simplify this. "Measurements of this parameter based on high-speed photography are reported. For natural flips" - and look at this. You're scrolling through this right now. It's like it's head-spinning physics. "For natural flips, the chance of coming up as started is about 0.51." So not 0.50.

Leo: You're likely, if you flip a "1," to flip another "1," by a small amount.

Steve: Correct. Exactly. Or if you don't randomize the starting orientation, then you're going to be biased on the way it falls. So here's a set of really interesting takeaways from, I mean, like, okay, who hasn't flipped a coin or had an occasion to do that? So here

on the Security Now! podcast, here's the wisdom from the research of guys who really studied this: If the coin is tossed and caught, it has about a 51% chance of landing on the same face it was launched. So in other words, if it starts out as heads, there's a 51% chance it will end as heads. But if the coin is spun rather than tossed, it can have...

Leo: There's a flick that people do; right?

Steve: Yes. It can have a much-larger-than-50% chance of ending with the heavier side down. Not really surprising, when you think about it. Turns out that spun coins can exhibit a huge bias. It says some coins will fall tails-up 80% of the time. So if you can trick somebody into choosing tails or, no, choosing heads, and spin the coin, that can be a huge win for you, depending upon the coin.

Leo: So if you can get heads, flick it so that it spins. And the more vigorously you flip it, the better.

Steve: Well, now, spinning is meant like this.

Leo: Oh, like that. Not flicking it.

Steve: Twirling it.

Leo: Nobody's going to let you spin a coin that way.

Steve: Twirling it on a hard surface, yeah.

Leo: That's not how you flip a coin.

Steve: That's the problem. If the coin...

Leo: Okay. I want to pick heads, and I'm going to spin it.

Steve: Yes. And it turns out 80% of the time...

Leo: So this doesn't count, like flicking it like that? Because that's - I don't know.

Steve: They were talking about - they were definitely talking about spinning it on a hard surface. If the coin is tossed and allowed to clatter to the floor, they say, this probably adds randomness.

Leo: Yes, of course.

Steve: Okay. If the coin is tossed and - oh, sorry. Oh, I got that one twice. Oh, no. If the coin is tossed and allowed to clatter to the floor where it spins, as will sometimes happen, the above spinning bias probably comes into play. So that's not surprising. Now, get this one. A coin will land on its edge around one in 6000 throws, creating what they called a "flipistic singularity."

Leo: Yes.

Steve: So there you go, baby, you've hit the flipistic singularity.

Leo: You know, if you'd asked me, I would have thought it would be much less likely than one in 6,000.

Steve: I did, too. I've never had it happen.

Leo: I've never seen that happen.

Steve: Unh-unh. Never seen it happen. But they're saying, eh, one in 6,000. So, and you could probably prove that at home because...

Leo: Get going. Get to work. You could tell us next week.

Steve: If you do, I want a picture. Turn, you know, be sure you've got your webcam or your phone recording it. Otherwise we're not believing it.

Leo: Yeah. I imagine there's some bias introduced by the fact that they're using automated flippers and things.

Steve: No, no, no. They removed that. The same initial coin-flipping conditions produced the same coin flip result. That is - oh, yes, they did actually create - their automated coin flipper, they were able to set it up so that every time it flipped it, it got the same result. Period.

Leo: Yeah. Period. Wow. Because that's super consistent, yeah.

Steve: The same initial coin-flipping conditions produce the same coin flip result. So it says, that is, there's a certain amount of determinism to the coin flip.

Leo: Because it's a mechanical device, yeah.

Steve: Right. Oh, and the more robust coin toss, which is to say the more revolutions, lower the bias. So if you, for whatever reason, you have to get a coin toss, and say you're the person who's calling it, if someone does sort of a lame toss where it's not really flipping, it's like in the air, but it's not really spinning fast, eh, that's going to tend to have more bias than if it spins really fast. And I guess we didn't really learn much.

Leo: Just don't gamble with professionals is what I learned.

Steve: Finally, many people just tweeted something which I haven't had a chance to dig into because it's just hot off the press. But it's a breakthrough in zero-knowledge proofs. And I'll just quote from this because it's a nice teaser. I will dig into it and figure out what it means and talk about it probably next week. And this is a guy, Amit Sahai.

"As a graduate student at the Massachusetts Institute of Technology (MIT) in '96, Amit Sahai was fascinated by the strange notion of a 'zero-knowledge' proof" - which we've actually covered on the podcast, and I'll remind our listeners and point people here in a second at where we talked about Ali Baba's Cave, famously - "a type of mathematical protocol for convincing someone that something is true without revealing any details of why it is true. As Sahai mulled over this counterintuitive concept, it led him to consider an even more daring notion: What if it were possible to mask the inner workings, not just of a proof, but of a computer program, so that people could use the computer program without being able to figure out how it worked." That is, not being able to see into it and reverse-engineer it. And there's actually apparently a breakthrough that makes this possible, which is very cool.

So as an intro to this concept of zero-knowledge interactive proofs, that's the title of podcast No. 363, where we tell the tale of Ali Baba's Cave. So long-time listeners will remember the podcast about Ali Baba's Cave. If that sounds new to you, that was on August 1st of 2012. Go check out Security Now! No. 363. You're going to need it because we will assume that that's knowledge in evidence when we go into how this can be leveraged, apparently, into the design of software, which can be used, but cannot be understood.

Leo: I didn't understand it the first time, but okay, go ahead. And I was there.

Steve: I did want to mention, Leo, I heard you mentioning on MacBreak Weekly, as you said you would, my discovery of the overtrain...

Leo: Yeah.

Steve: ...of Touch ID. It has been a huge hit in the past week. I've just - there have been so many tweets from people, even people creating blogs and web pages and things to retell and reexplain how this operates. And so I did create - I don't know if I had a bit.ly shortcut for that.

Leo: You did.

Steve: Okay, bit.ly/sgtid. And that must in fact expand out to somebody's blog posting because...

Leo: No, no, it goes to YouTube and goes right to the part of the show where you walk us through it.

Steve: Oh, exactly, yes. Exactly. So for people who may have forgotten about it or haven't had a chance to try it, for what it's worth, now we've a much larger base of experience of people saying, wow, it works.

Leo: Yeah, yeah. Yeah, it was a really good tip. I was glad I could mention it on MacBreak Weekly. And I don't know if you saw, but it was written up on iMore.com, and I suspect more and more people will play with it. It's really good. By the way, that is not a garbage truck in your backyard, everybody. You're listening to Steve's. It's our regular NSA pickup of his documents.

Steve: Our weekly, yes, it turns out that you can scan them. And what they're actually doing at that large facility in Utah is they are taking pictures of the shreds.

Leo: The shreds, in case they need to reassemble them.

Steve: And Watson is reassembling them, just to see if there's anything good.

Leo: Awesome. Awesome.

Steve: Yeah.

Leo: It's just metadata. Don't worry about it.

Steve: I did want to mention I'm now working on, as I did say, on SQRL's UI. And what's interesting, and not surprising, is that my work on the user interface is feeding back into the technology because this thing will not succeed if it's confusing to anyone. And so now I'm working on the de-confusing part, where we want to have this thing still be feature-rich, without overwhelming people. So what's happened is, almost inevitably, as I've been struggling with how to make its operation clear, I've realized, okay, now, wait a minute. We have, like, too many degrees of freedom built into the back end of the technology. So I've been making some changes which have ended up in some really nice additional functionality, essentially.

So there is a new page finished as a result of this. It's not the UI yet. But the SQRL region of GRC now has 20 pages, and not all of them are finished, but this one is. No. 5

is what I call the Key Flow. And so it's got a URL. But, yeah, if, Leo, you click on - oh, you found it. There it is, yeah. And it'll make your eyes cross a little bit when you first see the flowchart. But if you spend some time, other people in our newsgroups, the GRC newsgroup where we're working on this stuff have spent some time looking at it and reading through it and said, wow, I actually understand that. So the way SQRL manages passwords and keys, which we've ended up adding some features actually, and at the same time reducing the complexity of the user interface. So we're getting there.

And I'm now working on the UI, the actual what-the-user-sees in order to make it all go. So if you're interested in this, I commend you to take a look at page no. 5 of the SQRL site, which is the Key Flow [GRC.com/sqrl/key-flow.htm]. I think you'll enjoy looking at the chart, thinking about it, and then reading the text below where I explain sort of step-by-step what each facet of that is about.

And as you mentioned, Leo, the source of my income that allows me to spend time on things that I don't charge for everyone of course knows is SpinRite. We had a really nice testimonial from a Philip Cooke. He said: "I started my day with a Blue Screen of Death" - whoops, that's not a good way to start your day.

Leo: No.

Steve: "...advising that I had an unmountable boot volume." Which of course Windows users have seen from time to time, unfortunately. He said: "Efforts by a Dell tech only led him to the conclusion that we should reformat the" - yeah, we. The tech and I are going to "reformat the drive and lose all my data. Nothing would recognize the drive, and all the chkdsk commands in the book could not even see it or result in anything but the same blue screen on every reboot. A Maxtor utility I ran advised me to return the drive for a replacement. But I couldn't."

He said: "After getting estimates ranging from $400 to $2700 to recover my data" - and, by the way, they don't come with a guarantee - "and trying numerous other 'tricks,'" he says in quotes, "recommended by online chats, et cetera, I was fortunate to come across SpinRite. At first the glowing testimonials" - which obviously he means on my site, and of course I'm reading one such, which he then contributed. But he said: "At first the glowing testimonials seemed just too good to be true, and I will admit that I thought they may even have been fake," as he now, again, contributes one that really does sound that way.

He said: "So I invested the $89, downloaded the file, and fired it up. At first I thought that it was going nowhere because after four hours it still said 2% complete." Which of course meant it hit a bad spot, and SpinRite was just going to work until it fixed it. He says: "I figured I would leave it running. And imagine my surprise when I came back, saw the message that it had completed. It booted up, ran chkdsk, and then started Windows. All I can say is wow. Thanks for taking the time to create this program. It's bad enough losing data, but I also saved the hours it would have taken to recreate my desktop, links, et cetera. Needless to say, I'm impressed. Philip Cooke." So thank you, Philip, for a very nice affirmation of SpinRite's capabilities.

Leo: Steve Gibson is here, and we are ready to talk about the subject of the day.

Steve: So I love this data because it, first of all, was methodically and very nicely pulled

together. And I liked it also because it's sort of a snapshot of where we are at the beginning of 2014, the current Internet password policy. So we know what the problems are. We've talked about it a lot. In non-multifactor, user-generated passwords, they're going to typically have low entropy. They're going to generally be reused. And they're going to be like their own favorite secret, the birthdates or dogs' names or sometimes a string of digits, 123456 and so forth. So the incoming password is already in trouble.

Then we have the problem of websites storing them in plaintext or in unsalted simple hashes. And of course we always talk about the large number of instances where databases of stored passwords get loose. The hackers run them against the hash that was being done and crack all of the easy ones because, again, the poor storage on the website couples with the poor generation on the user side, and people's passwords get loose. And of course then famously, since people are generally reusing these weak secrets that they have, their account data gets loose, and it's possible for them to get impersonated on other sites, let alone the one that just suffered the breach. So this is just a big mess.

And yesterday, when I was on with Mike, I sort of explained, on TNT briefly, I explained that one of the problems is that, like, why do we have these problems? Why are there so many websites with really inadequate security? And I think one of the reasons is that security is invisible. There's no visibility from the user's view into the security of the site. Yes, sometimes if there's no password policy shown, if it accepts a two-character password, that would be a concern. I mean, so there are some things that the user can test. But most people don't. And the problem is that we know that security tends to conflict with usability. That is, the more restrictive the policies are, the more people that are not going to be able to use their favorite password, and so they're going to grumble about that. So the tradeoff is made, well, we don't want to inconvenience our users. So we're not going to impose strict regulations on what they're able to do.

But more than that, it's really not possible to know what's happening behind the scenes, how a website's security actually looks. And if there were some way of making that really visible in a usable fashion, well, then, to a much greater degree than I think we see today, websites would take the time to improve the security that they're offering because then it would become a feature. Then it would become something that they could be held to account for. And so that's another one of the reasons that I really like the work that Dashlane did, is that to the degree that this gets some press and some publication beyond the scope of our own listeners, who probably already have much greater and better password habits than the typical Internet user, this is a good thing.

So Dashlane, first of all, just I want to make sure we give them credit for this. They are another one of the password manager sort of companies. And of course we know LastPass is the one I use, only because - it's just of inertia, I guess. It does everything that I want. And I was able to get a full readout on the technology and verify the way it functions. People are always tweeting me, what about 1Password or this password or that password, or even Dashlane? And I just say, well, it's probably okay. And sometimes I'll take a look at the security and confirm that it looks okay. But I just haven't done the deep dive.

So Dashlane is another of those. They're a little pricey for their registered version. They ask for $30 a year, whereas LastPass asks for $12. And I don't really see anything to differentiate them. They've got mobile platforms and cloud synchronization, cross-device synchronization and so forth. But they are a password manager. And actually I don't know if I see multifactor support in their list of things.

Leo: Yeah, I mean, that's one of the things I love about LastPass. I can use a YubiKey or Google Authenticator.

Steve: Yes, yes. Yeah. But anyway, I did want to give them a tip of the hat because they produced the data which was really interesting. Now, at the beginning of the day today, I grabbed their raw data and created a massively large PNG file. And that link, and I'm serving that from my site, is bit.ly/sn441, all lowercase. And that of course is the episode of Security Now!, sn441, all lowercase. And the problem of course just being a huge PNG, and this is of their raw data, is that you lose the labels of the companies, the websites, on the left and the headings of the columns on the top.

And so someone said in the dialogue in Twitter this morning, hey, wouldn't it be great to create a big - this really needs to be turned into a spreadsheet. And so this was initiated by Jon M., who is @Liquidretro is his Twitter handle. He created it. And then a few other people jumped in, and I added some highlighting to the right-hand side of the score columns in order to make it a little more clear how they were paired. Somebody else added coloration. So now we have this very cool Google spreadsheet, which I recommend people take a look at. There may be a problem, depending upon how many people are trying to get it at once because, Leo, you explained to me that Google puts a limit on how many people can.

Anyway, it is bit.ly/, and again, all lowercase, sn441sheet, s-h-e-e-t [bit.ly/sn441sheet]. And it is fascinating what Dashlane found when they essentially had to - I don't know if they did this manually, or they probably wrote a script that they were able to maybe aim at these different sites because they have reverse-engineered the password policies, the online login password policies of the Internet's current top 100 retailers. So, and there are some - there's a summary that I'll discuss in a second.

But as I was, like, browsing around in the spreadsheet, I came across, like, the most curious things. And I talked about some of them at the top of the show. But, for example, I could understand, because one of the columns, for example, is minimum password length. Do companies enforce a minimum length? And, if so, what is it?

So it turns out that many companies do support a minimum length of eight. Eight seems to be common. We would argue, those of us who are Security Now! podcast-involved people, that eight's not enough, that you need more than eight in order to get security. But the problem is of course that's going to really start hampering people. They're going to be upset that their eight-character password is not enough if someone asks for more than eight. But eight was regarded as enough, if it wasn't a common eight-character password like the word "password."

So what I was curious about, looking down the minimum password length column, was that there was, like, there were some, like Northern Tool and 1-800-Flowers, which is a major Internet website, has a minimum password length of one. Now, okay. So that means your password can be "q," and they'll say, okay, fine. So a minimum password length of one is basically checking between blank and anything. So all it's really doing is seeing do you have - did you leave the password field blank? So I can understand that test. But there's a company called Build that has a minimum password length of two, which is really curious to me. It's like, so somebody checked to make sure the password field wasn't blank. But if it's one, then they say, no, no, you can't have one character. You need at least two.

Leo: [Laughing]

Steve: Okay. But get this, Leo. CafePress has a minimum length of three, as does Scholastic, Inc., and Urban Outfitter, and Nutrisystem. It's like, major websites. So you can use a password of four characters at CafePress, but not three. I just - okay. So somebody had to write code to say, unh-unh, no, three-character password, not good. Four, oh, yeah, we like that. Four is fine. Wow. And a minimum length of - and then, like, one notch up, minimum password length of four - I'm sorry. CafePress is minimum length of three. So I misspoke just then. So three characters is enough for CafePress, but not two. It's like, oh, just crazy. Karmaloop, Vitacost, Fresh Direct, Shutterfly - Shutterfly has a minimum length of four. Victoria's Secret, minimum length of four. Maybe that's so you can make your password be "love." And ShopNBC, minimum length of four. So anyway, there were some surprising numbers in here. And again, I ask myself, who is going to write code to set a minimum length of two, three, or four? That's just - I don't get it.

And then, it was interesting, of the top 100 sites, even like the really, really, really bad ones, and there are some really bad ones at the bottom of this pile. By the way, that spreadsheet is sorted in order of the overall total, the score that each site achieved in the test. So what Dashlane did was they took all these different parameters, and they set weights on the different policies that they tested. And then they were able to sum them together. So a hundred is a perfect score. A minus a hundred is, like, the worst you can get. And so this list, this spreadsheet is sorted by final score.

So like the column on the very far right, you can see it goes from a hundred down to minus whatever it is, for like the worst one. But every single company masks the password during entry. And I have to say, that's one of the things that I've never understood. And so, for example, in my SQRL UI, it'll be masked, but right there will be a button saying show me the password that I'm entering. Because I have never understood, like, the great security value. It's almost as if not showing it to the user somehow conveys some magic, where it's like...

Leo: It's imaginary.

Steve: It's like, ooh, nothing in the world can see it. Everyone should understand that it's just the screen, just the display has been instructed not to show you the shape of the character whose button you just pressed. It exists everywhere else - in your mind, and in the computer, and in the browser, and in the field, and in the form, and on the wire, everywhere except just those little pixels are not being lit up in its shape. Which I just - the only possible reason is that you're, like, you're entering your password with a crowd gathered behind you.

Leo: Watching you.

Steve: All watching you.

Leo: Well, in fact, that's exactly how I enter my passwords. But I'm the only one.

Steve: But, Leo, you'll be glad to know, every single one...

Leo: They all do that. I don't understand it.

Steve: ...of the hundred, even the crazy sites that do everything else wrong, everything else wrong.

Leo: One-character passwords, but no one can see it.

Steve: Yes. They cannot tell when you type "q." Because, Leo, who would come up with "q"? It's only the most obvious one...

Leo: Well, you have a chance, one in 26.

Steve: ...up there. No, no, look at that. Look at your keyboard. "Q" is right...

Leo: "Q" is like the first one, yeah.

Steve: That's where your eyes immediately go, to "q," yeah. And I'll bet that's actually the one everyone's typing.

Leo: Probably.

Steve: So, okay. I just have to say that's the stupidest thing.

Leo: Thank you. I thought it was just me. I've always - and it's particularly annoying on smartphones.

Steve: Yes.

Leo: Or you know where they do it?

Steve: Whose keyboard you're typing things wrong all the time anyway.

Leo: All the time. You need to see it.

Steve: Yes.

Leo: It's just crazy.

Steve: Yeah. I mean, it ought to have, right there, turn off the dots. I don't want dots. I want to see what I'm typing. Because, yes, I mean, Apple did the clever idea of showing you the most recent one you've hit. But then you have to look at it every single time to make sure.

Leo: You've got to watch, yeah.

Steve: And the point is it's not hiding it from anybody except you.

Leo: You know, Facebook, to their credit, they know this. Obviously Mark's a geek. Mark Zuckerberg probably thought, this is stupid. I've got to do it. But after you enter it wrong once, it then puts it in plaintext. It says, okay, let me give you a hand, in the mobile devices anyway.

Steve: Okay, yeah.

Leo: Let me give you a hand. It's pretty stupid. Thank you for saying that because it's been driving me crazy.

Steve: Something very worthwhile is when sites offer you advice because this is a teaching opportunity. My mom needs help. And so no one's really told her, when she started logging into, as she calls it, AWOL, she didn't - it's like, oh, I have to create an account. Hmm. Oh, I know. I'll use my children's names, or who knows what Mom uses. I don't know. But I'm sure it's not a high-quality password. And if right then there had been a few lines to explain to her some of the things she should consider, then she'd have not made a mistake. And but that was missing.

So to a greater degree than - actually to a surprising degree this is one of the things that is ranked, and you can see it on the spreadsheet. In that column there's a lot of Y's under "Is password advice being offered right there." Now, one of the things that's annoying is when they have a lot of criteria, and they don't tell you what those are right upfront. That's another gripe of mine is where you're wanting to create a high-quality password, but you don't know are they going to allow me to use special characters? Can I put digits in this or not? Because it's not until it fails that it then says, oh, sorry, you can't, you know, you have to use only this or that. Well, why not tell me right there upfront? So sometimes they do, sometimes they don't.

Another interesting aspect that they checked was whether mixed case was required. Clearly a good thing because that increases the size of the alphabet and makes brute-forcing the password more difficult. Not a perfect solution, but we see over and over and over many people use all lowercase because it's easier for them to type. They don't have to do the shift key. And again, especially in a mobile platform, where shift is something you have to more explicitly and deliberately do.

Now, the onscreen password strength meter is one of the best things you can do because

there is a teachable moment right there. Also, if you're not just showing them a meter, but telling them why they're not doing so well, or have a series of checkboxes, sort of like what I did over on the Password Haystacks page, where as they were typing things, when they, like, added uppercase, then blink, light up a light saying, ah, you've got some uppercase. When they hit a digit, whoop, another light lights up. Oh, good, you got a digit there. And so forth.

Now, to do that interactively you need scripting on. There's no way you're going to do this otherwise. The way to do it without scripting would require them to submit a bad password and then come back to them and explain why this password wasn't up to snuff. So not nearly as good as scripting. But we're seeing more and more instances where you really can't operate without scripting. And you could make your password entry page just tell them, in order to handle password entry, you need to turn scripting on, just for this page, please. And then give them a little built-in tutorial.

So the notion of an interactive password strength meter, that's really great. Yet today only 7 of the top 100 retail sites do it. So we see it. I know that LastPass does it, although they weren't on this list. But Apple actually is the No. 1 site in the survey. And we'll talk about some of the reasons why in a second. But that's clearly a useful thing to do.

Now, the final two, I mean, there are some other criteria worth looking at, but do they block login after four missed attempts? We see instance after instance where you can just guess forever. And we know why - first, of all, maybe they're not doing it because they don't care. Or maybe they're not doing it because they don't want the support burden of somebody who now cannot log in after four failed attempts. But Apple blocks you after four misses.

Leo: Good, good. This prevents brute-forcing it.

Steve: Yes. Newegg blocks you. Nordstrom blocks you. Hayneedle, whoever they are. Foot Locker, Costco, Staples, ShopNBC. Now, I noticed that ShopNBC blocks you after four tries, but also allows a four-character password. So I guess...

Leo: That's okay, though; right? Because the chances, well, I guess if it's "qwer," that's the first one I'd guess. But if you have four random characters...

Steve: Right, right.

Leo: If you have four random characters it would take you a few times to guess it. So in some ways that makes up for weak policies in other...

Steve: Really, blocking after four is, I mean, I get it that it's going to be a problem. I mean, you're going to have - you have to have online tech support. It's going to be a problem. But as you said, Leo, I can see the necessity for a site like Apple, which is a high-value...

Leo: You don't want robots attacking the login.

Steve: Yeah. Now, again, because it's online, you're going to have a delay loop. And in fact, if you could afford it, and server architectures don't make this easy, but to be able to hold up the response to mistaken passwords, for example, UNIX has famously done this, where as you keep making a mistake...

Leo: It slows it down, yeah.

Steve: ...it waits longer and longer and longer to come back. That's beautiful. But that works because you're on the computer. The computer you're trying to log into is right there in front of you, typically, or remote login and so forth. But a web server typically has an architectural problem of holding up a response for a long time. It could be designed so that it's not a problem to do that. But that would be the nice thing.

So these guys, Dashlane, test blocking after four and blocking after 10, those two. Microsoft blocks after 10, but not after four. Target, same thing, 10. CDW, Amway, Musician's Friend, WW Grainger, Walgreens, CVS, and others. So there are major sites that do put a limit. But, for example, Amazon doesn't. You can sit there and guess someone's login password till the cows come home, just keep on going.

So overall summary of the stats from this analysis: 73%, so nearly three quarters of the current top 100 retail online websites, accept passwords having six or fewer characters. So many are just ridiculously, like not checking your length. Or again, I can't understand, oh, four is enough. Okay. 62% do not require a mixture of letters and numbers. So nearly two thirds don't care if it's all alpha. They won't reject it saying, oh, sorry, please add some numbers to your password. But frankly, the fact that a third do, I guess I'm surprised it's that many, and I'm glad that it's that many.

Leo: That seems to be one of the most commonly adopted security measures is to - you've got to have a number. Kind of drives me crazy because I have a long alphanumeric that I use. I mean alpha-alphic, alphabetic.

Steve: Right. I've run across that being annoying, too.

Leo: I should point out these are retail sites. These are retailers, not...

Steve: Well, yeah, but, I mean...

Leo: So they have a strong economic interest in you...

Steve: ...[indiscernible] traffic.

Leo: Yes, understand. But they have a strong economic interest in you creating an account. So they have to balance appropriate security with inconveniencing the customer too much.

Steve: Yeah. I forgot to mention, I had it in my notes, but with SQRL there's none of this. This all goes away.

Leo: It goes away, yeah.

Steve: Of course passwords go away. And breaches go away. SQRL gives the website nothing that it has to protect. Its database can get stolen. It doesn't matter. I mean, there's no way to impersonate someone, even if they get the database for a given site. So this fixes all of this. And so fingers crossed that this will end up happening.

Okay. Now, I was surprised about this number, too. 55% accept 10 of the most commonly used weak passwords - 123456, 111111, or the word "password" and so forth. So they test, and the spreadsheet shows, the 10 most commonly used passwords. 55% accept 10 of those, which is to say they're not checking. But I was impressed that that means 45% are actually checking to see, hopefully before they hash it, whether it's one of these really dumb passwords, and just say no. No, sorry, come on. You try harder. You could do better than that. So I'm somewhat encouraged by that.

Half the sites, 51%, make no attempt to block entry after 10 incorrect password entries. And those 51% include Amazon, Dell, Best Buy, Macy's, and Williams-Sonoma. No attempt. You can guess forever. So that's a little disturbing because, as you said, Leo, I mean, 10 is a lot. 10 it's like, okay, sorry. Call us. Click here. Do something.

Leo: Ten seems to be enough. And it's a high enough number that it should give you plenty of opportunities. Because I understand people go, oh - and I do this a lot - which password did I use, or that kind of thing.

Steve: Yeah, well, exactly. That's the failure model is you can't remember which one of your collection of common passwords that you used there. 61% don't provide any advice on how to create a strong password during signup. Although, again, I'm encouraged that that means 40 or 39% do provide advice because that's something clearly useful.

93% do not, because remember we said that only 7% did, that's 7 out of those 100, do provide onscreen strength assessment, which is really, I mean, that's just a win. That is, like, the No. 1 way to get people just to sort of passively encourage them to create a good password, show them that they're in the red, and give them some clues about what they ought to add to the password they're building in order to get their score higher and generate a stronger result. I like that.

So overall, of the 100 that were checked, only 10% scored above the threshold for good password policies. That was, in this standard of measure, 45 points or more in the roundup. And you could go, you'd get a hundred, as far as a positive hundred, or as poor as a negative hundred. And only 10%, 10 out of that 100 pushed it over a score of 45. And then one of the other things they checked is, oh, I forgot my password, what is it? Eight sites, including Toys R Us, J. Crew, 1-800-Flowers, which actually no longer

surprises me because they accept one-character passwords, send passwords in plaintext via email.

Leo: Wow. Really, that many? Wow.

Steve: Yeah, yeah. Eight out of those top 100 just say, oh, here's your password. And I get tweets about...

Leo: Which means they have it in the clear, as well; right? They have it.

Steve: Yes, exactly. Actually, someone just tweeted me the other day, Yahoo! said, in response to them having to change their password, your new password is too similar to your previous password.

Leo: Yeah, I've seen that in places, yeah.

Steve: And what does that tell you?

Leo: They know your password.

Steve: Exactly.

Leo: Because a hash wouldn't reveal how close it was.

Steve: Exactly. A hash would have looked completely different if you'd just changed one character. So that means Yahoo! is also storing it in the clear.

Leo: Yeah, wow.

Steve: Which actually is no surprise, knowing how great Yahoo! security is.

Leo: Their policies are not great, yeah.

Steve: Apple received the highest rating and was the only retailer of those hundred to get a perfect score. Apple was perfect. Newegg, Microsoft, and Chegg - what is Chegg, C-h-e-g-g?

Leo: I don't know.

Steve: Anyway, whoever they are, they're big somewhere. They all tied for second place. And Target rounded out at third place. So their online password handling is pretty good, even if their in-store point-of-sale terminals have had some problems.

Leo: Chegg is a textbook store for students.

Steve: Ah. Also MLB.com, Karmaloop, and Dick's Sporting Goods received the three lowest scores. And I got a kick out of noticing, of course MLB, I did the math here in my head, I figured that had to be Major League Baseball.

Leo: Yeah.

Steve: And they allow the password "baseball" on their site.

Leo: [Laughing] I bet that's No. 1 on their site.

Steve: How many people do you think might be "protecting themselves," unquote, with the password "baseball" on MLB.com? Amazon, Walmart, Victoria's Secret, Toys R Us, were also among the lowest ranked sites, receiving scores of negative 35 or below. So again, Amazon, Walmart, Victoria's Secret, Toys R Us, very, very poor policies.

Leo: What this doesn't reflect, though - I mean, look, Target was, what, No. 3? - is other security policies. For instance, that people could hack in. And Apple, No. 1, quite notoriously gave up Mat Honan's account information to social engineering. They've changed that so it doesn't work, that technique, anyway, doesn't work anymore. But that's one of the things we've learned from the Mat Honan case and others is that having great passwords and great password security is not necessarily all that's required to securing your account.

Steve: Right, right. So these guys sum up their overview by distilling all of this to four simple password management strategies, or advice to websites: Require that passwords contain at least eight characters, and a combination of upper and lowercase, numbers, and symbols. Block account access after four failed login attempts. I think that's a little short.

Leo: That's low, yeah.

Steve: Yeah, I do. I think that's going to really cause some problems.

Leo: Ten's plenty.

Steve: Yeah, I think so.

Leo: Ten is enough to stop brute force.

Steve: I think so. Provide users with onscreen advice on how to choose a strong password during signup. I love the onscreen advice. And provide users with an onscreen assessment of password strength while they're choosing a password. That is, basically, look at the password. And the dictionary of most commonly used passwords is so short, which is really sad, but that you could easily have that as part of your JavaScript. So I'm immediately thinking of you don't want to be making a roundtrip to check the password entered so far against the database. And the point is you don't have to.

Leo: Just do it at the end, yeah.

Steve: The list of really bad passwords will easily fit in the JavaScript, which is also showing you, like, do you have characters, upper and lowercase and so forth. Add a simple dictionary to say, okay, yes, but this is something you can't use. And actually, most of those common passwords would fail the upper/lowercase, numbers and symbols tests because they don't have any of that. They're all numbers, or they're all lowercase alpha.

So anyway, that's it. I encourage listeners who are interested to go check out, just scroll around this web sheet, or the online web page spreadsheet at Google. It's really interesting to see what companies are doing. And with any luck, we will see evolution over time. Clearly, the best companies weren't always this good. They decided to take this seriously and tighten down the screws, batten down the hatches, make their online login tighter because, again, this isn't automatic that these things happen. You do need to write some code and make it so.

Leo: And here's one more password we want to add to the most common list. It would be 'w3Lc0m3!HERE' in leet speak. You probably shouldn't use that one. This is a video, I'm sure you saw this story.

Steve: Oh, my god, yeah.

Leo: Of the super secret Super Bowl security center that ran on CBS, and of course there's a big screen, a lot of big screens. By the way, a lot of these screens running it looks like Windows XP, I've got to tell you. When I look at that menu bar, that's a very XP-like menu bar right there. They also neglected to pull down or, I don't know, they had a big screen that said WiFi access login "marko," password "w3Lc0m3! HERE" in leet speak.

Steve: Yeah, m-a-r-k-o was the SSID for the WiFi, and then that's the shared...

Leo: Oh, that's the SSID, of course, not the login, right. It was apparently an internal WiFi access spot. But just, you know, just a little tip. Oh, it's so funny.

Hey, thank you, Steve Gibson. As always, a very interesting show. You get the show notes Steve now posts on his website, GRC.com, with the rest of the Security Now! stuff, including 16Kb versions of the audio for the bandwidth-impaired, people who still want to listen, but don't want to download a giant file. Transcripts, too, written by Elaine Farris, very nicely done there. You'll get full versions of the show on our website, TWiT.tv/sn, in higher quality video and audio. You can also find it wherever podcasts are aggregated - netcasts, I guess we call them - including iTunes and Stitcher and all those places. If you visit GRC.com, don't forget to check out all the great stuff, the freebies, ShieldsUP! and the Perfect Paper Passwords and all of that. And if you're so inclined, wouldn't be a bad thing to tip Steve by purchasing a copy of SpinRite, the world's best hard drive and maintenance recovery utility.

Steve: Even if you don't think you need it, it does actually prevent hard drives from failing, as many users have found. I get reports, it's like, well, you know, I've never had a failure. I run SpinRite every couple months.

Leo: It's a good idea.

Steve: And it keeps my drives from failing. It actually does work.

Leo: It's like an oil change for your hard drive.

Steve: Exactly.

Leo: Our show we do now at a new time, and I want to remind you that we do it Tuesdays, 1:00 p.m. Pacific, 4:00 p.m. Eastern time, that is 21:00 UTC, at TWiT.tv. Right after MacBreak Weekly, if you tune in and MacBreak Weekly is still going on, it'll be on soon. We also, as I said, make it available after the fact. Thank you, Steve Gibson, and everybody else who's joined us here today. See you next week on Security Now!.

Steve: Thanks, Leo.