Listener Feedback #194

Description: Leo and I discuss the week's major security events and discuss questions and comments from listeners of previous episodes. We tie up loose ends, explore a wide range of topics that are too small to fill their own episode, clarify any confusion from previous installments, and present real world 'application notes' for any of the security technologies and issues we have previously discussed.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-468.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-468-lq.mp3

SHOW TEASE: It's time for Security Now!, a great big landmark episode. I'll let Steve share that news with you. He'll talk about the 1.2 billion password exploit; BadUSB; and we'll answer your questions, too. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 468, recorded August 12th, 2014: Your questions, Steve's answers, #194.

It's time for Security Now!, the show that protects you, your loved ones, your privacy online with the man in charge, the Explainer in Chief, the head honcho at the Gibson Research Corporation, Mr. Steve Gibson. Hi, Steve.

Steve Gibson: Hey, Leo. It's great to be with you again for the final episode of Year Nine.

Leo: Holy cow. You mean we're completing our ninth year?

Steve: Just today. By some strange coincidence, it was actually Elaine who, in our weekly conversation where we're exchanging the transcripts, she mentioned a few weeks ago that because of the fact that we had shifted to Tuesday, and the phase of the moon and leap years and all that all coming together, our very first podcast we ever recorded was August 19th of 2005.

Leo: Wow.

Steve: So August 19th of 2014 is next Tuesday.

Leo: So there you go. There you have it.

Steve: So next Tuesday starts Year 10.

Leo: That's kind of amazing that we have been doing this that long. I just - does not feel that long.

Steve: It really is.

Leo: What do you give somebody for the ninth anniversary? Is that Bakelite? I think so.

Steve: Resuscitation, I think.

Leo: Nine years. Well, congratulations, and thank you.

Steve: Fibrillation. Hey, thank you, it's been the best thing I've ever done. It's been absolutely...

Leo: Oh, no, it hasn't. Don't say that. But it's the best thing...

Steve: One of.

Leo: It's one of the absolute best things I've ever done.

Steve: Well, I tweeted to Jen - Jen's up at a yoga retreat at the moment. She does that every summer. And yoga is like oxygen for her. And I told her that today was the end of the ninth year. And I said, "Nine years," I said. "That's something."

Leo: That's it. The whole story went to that, huh? "That's something." That's something. Yeah, it is. It is something, all right.

Steve: As they say these days, "That's a thing." I didn't know that was a thing. Oh, yeah, that's a thing.

Leo: It's a thing, yeah.

Steve: That's a thing.

Leo: I think Tom Merritt and Molly Wood had a podcast briefly called "It's a Thing," that they talked about it's a thing. Things that were things.

Steve: Those things that become things.

Leo: It doesn't - it sounds like a tautology, but it is not.

Steve: I think I might have first heard it used on "The West Wing." It was the kind of smart language that they used there.

Leo: Sorkin type, yeah.

Steve: Yeah. And Josh said, "That's a thing?" And it was like, oh, yeah.

Leo: Oh, that's a thing. You're right.

Steve: You can do that. You can buy one of those, whatever it was, yeah. So we've got a great podcast. It's a Q&A, our 194th Q&A.

Leo: Holy moly.

Steve: We've had a Patch Tuesday, a little BadUSB follow-up. We've got to talk about the discovery of the CryptoLocker private key repository which has allowed the creation of an unlocking capability. And of course the news of the Russians, a Russian group of 20-something hackers who have 1.2 billion passwords and usernames.

Leo: Yeah. I wanted to know a little bit about that. That seemed odd to me.

Steve: Yup. I have some back story on that. And then I wanted to talk to you about Google's security biasing their website ranking, which I think is interesting.

Leo: Ah, yes.

Steve: And LastPass had a problem this morning that they seem to have recovered from, but it upset people. And then the question of whether your potato chips might be spying on you. So lots to talk about. And of course then Q&A.

Leo: My potato chips?

Steve: Yes.

Leo: Not computer chips?

Steve: Turns out that a bag of potato chips might be leaking critical information.

Leo: Oh, crap. I can't believe it. Oh, lord.

Steve: Not a big-newsy Patch Tuesday. There were nine bulletins in all. Seven of them were sort of important, but not critical. The most severe affected all versions of Internet Explorer, and the second most - oh, and that was a remote code execution vulnerability. Now, what I saw said that it was IE on Windows 8. But when I looked at the Microsoft background, it looked like it was just across the board everything. And then the second most important one was also remote execution, and that one was interesting. That was Windows 8 and 8.1 and Media Center TV pack for Vista, of all things. And apparently there's some sort of a graphics processing pipeline vulnerability that can allow code to get executed in your machine on those platforms. So, and I updated my Win7 box, said it had 10 things it needed to fix. So it was probably these things from this morning. So just update Windows, as always.

We didn't learn a lot more detail from the actual presentation of our two German security guys at Black Hat last Thursday when they gave their BadUSB presentation. So, and I don't know that there is a lot more to learn. I've been listening to you talking about it. I wasn't home, unfortunately, for TWiT, but I was able to watch it a little bit on Sunday. So I'm sorry that I missed the opportunity to be there, Leo.

Leo: Yeah, that's okay. But Father Robert Ballecer interviewed the guys because he was at Black Hat. And so, if you haven't watched This Week in Enterprise Tech, his interview with the guys is there.

Steve: Oh, good.

Leo: And it is very - to me it's scary. They answered a number of the questions we've been asking, like why don't they just make it with unburnable ROM? And it's really scary. Turns out in their Black Hat presentation they demonstrated the exploit. And you can do it from software on a computer. You don't need any special hardware. And that's really - that means, if a computer is infected with malware, that malware can write to the USB drive - or the USB device, I should say, because it could be a phone, keyboard, mass storage device of some kind - modify the firmware. And imperceptibly, you take it out and give it to somebody. No antivirus can detect it. I mean, I guess you could write one that would maybe detect it, but nothing right now. And there's no way to fix it.

Steve: Well, yes. I've done a little more research since. And there actually is, in the USB specification, the formal USB spec has something called DFU, which is Device Firmware Upgrade. So that is an in-band, sanctioned, formal technology for allowing a USB-connected device to have its firmware upgraded. So some devices may support that. Or they may just use like their own non-spec style, which you could determine from reverse engineering. And then the other thing that we talked about was like, and this has been something that you and I have discussed, why is the firmware writeable?

Well, in thinking about it a little bit further, and again I did some more research, imagine that you're a thumb drive, and you've got this huge, like, 16GB, 16 billion bytes of read/write nonvolatile grazing land. Of course you're going to take a little corner of it and put your firmware there. Why have a whole separate region of memory? Because it's going to be more expensive, and these things are certainly cost-sensitive. So nothing is more natural than just taking a little tiny edge of the memory map of 16-plus gigs and just saying, oh, we'll load ourselves from there when we start up. I mean, you can see why the firmware would be writeable.

Leo: They also said nobody wants to manufacture something they can't fix if they find a bug.

Steve: Right.

Leo: You'd have to throw them all out.

Steve: And then there was some conversation about why not use a security certificate. Well, first of all, the way you would apply that would only be useful for the device protecting itself from abuse. That is, and it would have to have a non-writeable portion in addition to a writeable portion so that the non-writeable portion couldn't obviously get overwritten, and that it would have to check the signature of the writeable portion. This is what Apple has gone through with the iPhone, and we've talked about this kind of approach.

So that's where the device is protecting itself against malicious firmware. It makes total sense for an iPhone to do that because Apple wants to protect themselves from that. But we're talking about thumb drives that virtually have no profit margin in them, being made in vast quantities in China. And they don't care. They just want to sell them. So it just ends up not happening. And as you said, Leo, and as they said, it really is something to concern ourselves about.

Leo: They said it's undetectable. It's easy to do. It turns out you don't need - I thought you might need some special hardware. You can do it from any PC. And they said, though, don't panic. There's no examples of this in the wild. I had to add a "yet."

Steve: Uh-huh.

Leo: Because I think that we're not far from that happening. And I imagine that this

will create a business for some USB device manufacturers to make safe USB devices.

Steve: Yes. If they're able to start claiming that their firmware is not rewriteable, then that's a buying point...

Leo: You bet.

Steve: ...for security-conscious people. Oh, and by the way, Stina shot me a note immediately after this news happened, I think it was hours after the podcast, saying, just so that I knew, and actually one of the Yubico guys blogged immediately about it, that the YubiKey is absolutely safe. Though parameters can be rewritten, the firmware cannot be. So they cannot have...

Leo: There's probably very little storage on those things. But that's just one. There are several different kinds of YubiKeys, and some can be modified; right?

Steve: Stina said no.

Leo: None of them.

Steve: Again, parameters, yes; configuration, yes; but not the underlying firmware. What she said was that was in ROM, that the firmware itself is protected.

Leo: Good. That's great.

Steve: So two companies, FireEye and Fox-IT, teamed up. They were both doing research in the whole CryptoLocker virus deal, and they were involved in that much-publicized infrastructure takedown of CryptoLocker, which we heard about. Well, it turns out that they were able to track down the servers where the private keys were being stored.

Now, to remind our listeners, when someone gets infected by CryptoLocker, their instance of CryptoLocker per machine, that is, per instance of infection, contacts the server at a wacky, made-up DNS name which is algorithmically derived so every day there are new DNS servers which the virus is able to calculate. And so the bad guys are putting them online, sort of in a moving forward fashion. They look up that DNS address in order to get the IP of a control server. When they establish that connection, they ask it for a public key for this instance of infection. So the server remotely generates a public and private key pair, keeps the private key, sends the requesting malware the public key. It then, as it's encrypting each file, it generates a random number, a 256-bit encryption key, which it encrypts with the public key that it's received from the foreign server. And then it appends that, or prepends that key in a special header to the encrypted file.

So what the victim ends up with is all of these different types of files, encrypted with a

special header containing an encrypted symmetric key which was encrypted with the public key, but can only be, as we know from the way public key crypto works, can only be decrypted with the matching private key, which has never left the master evil control server. So at that point the victim's sole recourse, I mean, and this is industry-grade crypto, I mean, this is the good crypto that's giving us TNO security for cloud file protection and all the kinds of beneficial crypto that we talk about, turned to a bad purpose. But it doesn't make it any less effective. It can't be cracked.

So the only recourse is for the victim to then pay X number of bitcoin, or MoneyPaks while that was initially being supported. And we're coming up now on a year. This was a year ago next month, a year ago September this CryptoLocker hit the news, and we started covering it on the podcast. So it's been quite an experience. So only by paying the bad guys is the private key then sent to the victim, allowing them to run the decryption process, decrypt the headers to get the key that was used to individually encrypt each of the files.

So these guys found the repository of private keys. I mean, that's what you need. They're not generated algorithmically. There's no way to compute them. You had to actually find them. And they did. And so decryptcryptolocker.com is now online, and it is a UI, a user interface to the database of the master database of individual private keys for every victim that CryptoLocker has attacked. And anybody who has been a victim, you go to decryptcryptolocker.com, give them your email address, so that they can email you the private key, and a sample encrypted file.

And I'm not exactly sure of the process they go through. I mean, they could run the sample file against their key database. Or maybe there is some identifier in the header that identifies through a serial number, for example, which private key is the correct one. In any event, you give them a sample of any file of yours that is encrypted, and they will send you back the master private key and their own decryption executable, which you then run on your machine to get your files back.

Leo: Wow. Such good news.

Steve: So, yeah.

Leo: That's amazing.

Steve: Very, very cool. And you know what I mean, it's nice that we have this sort of conclusion to this interesting drama. I'm not sure how many people this will help. If, for example, people went to a backup that was old, but had the wisdom to keep their encrypted files, and the old backup caused them to lose some data, now they have the ability to get that data back. So if anybody, for example, had to fall back to a backup or suffered a catastrophic loss of data - and we were hearing stories about, what was it, police stations, where all the machines got it?

Leo: Yeah, in Massachusetts or somewhere, yeah.

Steve: Yeah. They just wiped them out.

Leo: Now, let me ask you. Andy German asks a good question, or Adam German. Didn't they say, if you don't pay us in 72 hours, we delete the key?

Steve: They did, but apparently...

Leo: Apparently they didn't.

Steve: ...that is not the case.

Leo: That's good.

Steve: Because they found all the keys.

Leo: That's awesome. So they don't charge for this service at decryptcryptolocker.com; right? It's free.

Steve: No. Absolutely free.

Leo: That's great.

Steve: Absolutely free. The FireEye guys said: "To help solve the problem of victims' files still being encrypted, we leveraged our close partnership with Fox-IT. We developed a decryption assistance website and corresponding tool designed to help those afflicted with the original CryptoLocker malware. Through various partnerships" - they're a little coy about this. They said: "Through various partnerships and reverse-engineering engagements, Fox-IT and FireEye have ascertained many of the private keys associated with CryptoLocker. Having these private keys allows for decryption of files that are encrypted by CryptoLocker."

And, in fact, I have some links in the show notes. There was a really great write-up and sort of a backgrounder about CryptoLocker's last year of history that Fox-IT.com blogged, let's see, six days ago. So you could go blog.fox-it.com and probably find it that way, too. It was a CryptoLocker Ransomeware Intelligence Report. Had neat charts and graphs and geographical information and all kinds of stuff. So if anyone's interested, that's a great resource.

Meanwhile, The New York Times reported and a bunch of other news outlets picked up on the news that a Milwaukee-based security firm named Hold IT released the news that there was a Russian crime ring that had amassed the largest known collection of stolen Internet credentials, which upon analysis reduced to 1.2 billion username-and-password combinations, encompassing more than 500 million email addresses. This Hold IT, or the Hold Security guys said that, by July, which is last month, the criminals had collected 4.5 billion records, each a username and password, though many overlapped. After sorting through the data, Hold Security found that 1.2 billion of those records were unique. Because people tend to use multiple emails, they filtered further and found that the

criminals' database included about 542 million unique email addresses.

So here's what's going on because they've managed to figure out a lot about how this happened. First of all, it's a small hacking ring based in a little city in South Central Russia, flanked by Kazakhstan and Mongolia, with fewer than a dozen kids in their 20s, who have personal relationships, not virtual. They know each other personally, so it's a tight-knit group. They've created a series of PC-based botnets which infect people's computers. And as those infected users visit websites, the bot running in the machine performs a background SQL injection vulnerability test. And if a site responds to a SQL database injection, it's flagged and reported back to headquarters. And then the team goes in and sucks out the entire database.

So it's a very clever sort of two-phase strategy. They've got just widespread malware in a botnet which is probing based on the habits of the users whose machine it has infected. It sits there and, as the individuals visit websites, that's the source of potential compromised web servers. So background SQL injections are performed to see if the database, if the back - I've forgotten the term. The back room? Not back room. The back...

Leo: Backend?

Steve: Backend, thank you. Yes, the backend...

Leo: Okay. We're getting old, Steve. Nine years we've been doing this.

Steve: If the backend database is accessible through the web UI, that information gets sent back. Then, and then this is what's so cool, is that simple test makes the site a candidate. And then humans, rather than any kind of automated system, each one is an opportunity and a challenge. So they go there and use human-driven analysis to figure out the table name and all the record names and so forth, in order to suck this thing dry. As a consequence, the profile of the identities in this 1.2 billion username-and-password collection is very different. It's not just Fortune 500 big name sites. It's tiny websites. And when you think about it, that's beginning to be the bigger Achilles heel because, even though the big guys have been compromised, they've got budgets and security teams and all that. But all these mom-and-pop sites tend to just use drop-in packages which are much less secure, turn the key, up, and everyone's excited when it starts to work, and they don't worry about security.

But the problem, of course, is that to the degree that the typical Internet user reuses their logon credentials, the vulnerability is that they've used their login credentials on a site that is insecure, some small little site on the Internet, but also use them for their banking site or for other major sites that really matter. And so, as a consequence, we have essentially credentials, email addresses, usernames and passwords, 1.2 billion of them, that might be repurposed and reused. So that's what's going on behind this news of this massive 1.2 billion large credential database.

Leo: I'll be honest, I was a little skeptical when I first heard the story because Holden Security is charging people a fairly hefty fee to see if they're in the database, right, 250 bucks?

Steve: Oh, no kidding?

Leo: Yeah.

Steve: $250?

Leo: Yeah.

Steve: Wow.

Leo: Unless I'm - I may be completely wrong. But that's what I - that was my understanding. And so I was skeptical. Brian Krebs did say, "I've seen it; it's real." I know whatever his name - his name is Hold. But so Brian Krebs has vouched for him. And I think that's enough for me.

Steve: Absolutely, yes, for me, too.

Leo: But I thought that was odd.

Steve: Boy, that's disappointing, yeah.

Leo: Seems like that's how they make their living, I guess, but...

Steve: Yeah. And I did, I ran across some references to this guy, the Hold Security guy, sort of operating the way Brian does. We've talked about how Brian really does spend time, he invests time in the underground, I mean, dealing with this. And I got the sense from the research I did that the Hold Security guy is doing the same thing. He's cultivated relationships over time, no doubt using a pseudonym, of course. But he's gotten himself into this and been able to pull out this intelligence. Again, it's a mixed blessing that he's charging that kind of money.

Leo: Well, he says he's charging for the Hold Security Electronic Identity Monitoring and Protection Service. But I believe you have to pay for it to find out if you're on that list. So...

Steve: Yeah. So there.

Leo: I have mixed feelings about it. It's his business. In the past when this kind of stuff has been uncovered, people have published, you know - but that costs money. You put it online, you know, type in your email address, and we'll let you know if you're in the database, that kind of thing. But that costs money, so...

Steve: Yeah. Well, I guess it's his right to do that, if he chooses to. Now, one of the things that came up, of course, is in the wake of this, sites were saying, were advising listeners or, you know, readers, the only thing you could do now is change all your passwords on all of the websites that you visit. That's like, oh, my lord, okay. And I've had people asking me, "Do I have to do that?" It's like, oh, I don't know.

Leo: You know, I'm looking at it now. Apparently, for individuals, it's quite a few hoops you have to jump through, but Hold Security is saying that they will check free of charge for an individual. So I'm going to try it, and I'll let you know. I'll probably have a - yeah. But you have to register and give them your email and blah de blah blah. And blah de blah blah blah.

Steve: But except you have to listen to this because we want to talk about Google. So they've blogged that they're going to - and for our listeners who don't already know, this is interesting, I think, that they're going to start adding weighting - w-e-i-g-h-t-i-n-g - of search results based on whether sites are supporting HTTPS connections. So what they blogged was: "For now it's only a very lightweight signal, affecting fewer than 1% of global queries and carrying less weight than the many other signals such as high-quality content, while we give webmasters time - isn't that gracious of Google - to switch to HTTPS. But over time, we may decide to strengthen it because we'd like to encourage all website owners to switch from HTTP to HTTPS to keep everyone safe on the web."

So I'm of two minds about this, Leo, as I imagine any sophisticated person would be. It's like, well, I mean, this really is Google using their formidable search strength and position in the industry to sort of push a policy onto the web. Not a bad policy, but still one that's not free. I mean, HTTPS is not free. You know? You've got to buy a credential, and you've got to renew it constantly. And you'd better not forget, or suddenly no one can get to your site after your certificate expires.

Leo: Yeah, we talked about this on TWiT on Sunday, as well. And my issue is I, for instance, TWiT.tv, there's no reason to encrypt traffic. You're coming there, it's read-only, to see what's on, or to get a link, or to find out who Steve Gibson is. Is there a reason I should make that HTTPS?

Steve: No.

Leo: There's nothing to protect.

Steve: I completely agree.

Leo: And, now, Google says it only slightly impacts the weighting, like 1%.

Steve: Seems to me you either do it or you don't. I mean, you either weight pages so that they're going to, I mean, but the idea is they're going to appear ahead if, secure pages are going to appear ahead of less secure. Now, I don't like the tone of this blog post, frankly. I mean, that's what sort of annoys me is that, like, this sense of entitlement they have. Obviously they can use whatever algorithms they want. But I

liked it better when I first heard this, before I found out what they were saying about it, is it made sense to me from the standpoint of maybe a weak quality metric.

Maybe, I mean, because here's Google. The reason they've achieved the prominence they have is that the original incredible quality of links inbound to a site, the original Google search-ranking algorithm that was brilliant and immediately gave Google global domination on search, was that they figured out how to do this. So it seemed to me that maybe one of the metrics that you could mix into a valid search quality result would be is the site secure, arguing that better sites will - whatever that means.

And that's the problem as you start getting into this. Spend the money to add security, and so for a bot that has no other way, I mean, this is all automated. So the bots are looking at links and trying to not be spammed and looking at keywords and all the different things that the Google system uses. So it made sense to me that whether the pages are being delivered over a secure connection could be one more "signal," as they use the term, to help in ranking a site. But the idea that they're saying, well, we want to encourage, like, this is literally being done because we're going to force security on the web by down-ranking sites that don't have it. It's like, okay.

Leo: We support HTTPS Everywhere. We've talked about that, and we've talked about it before, and we think it's a good thing. But to change search rankings based on that seems odd.

Steve: I know, yeah. It seems a little bit like they're pushing a policy.

Leo: You know, one thing - yeah, which they shouldn't be doing.

Steve: Right.

Leo: One thing that could maybe make this better is if they became a certificate authority and started issuing free TLS certificates. How about that?

Steve: Say that they didn't like colored backgrounds.

Leo: Right.

Steve: You know, it's like...

Leo: Right. Oh, you've got an animated GIF on that front page. We're going to downgrade you.

Steve: And, you know, we'd really - we think we need white. We really don't want any color. So it's like, okay. I mean, again, in general, seems like it's a good thing.

Leo: What could be wrong; right.

Steve: But security is not free. It's not free. And so we're saying you've got to pay money, now you've got to pay money in order to compete in Google rankings is what this comes down to.

Leo: How much is a - now, you get an extended cert, which is very expensive. But how much is...

Steve: It's several hundred dollars.

Leo: Yeah, I mean, if I'm just a blogger...

Steve: Every three years.

Leo: ...I'm not going to pay several hundred dollars every few years. I'm not making that much money. I'm just doing this for fun.

Steve: And most don't. It's why we're still in a world where security is there when you need it, when you want some credential protection, but otherwise it's not. And remember, from time to time, when these certs expire, I mean, there's...

Leo: It'll break things, yeah.

Steve: ...significant overhead that never goes away, where you've got to remember to pay your money every couple years. Anyway, so, I mean, there are, what is it, STARTTLS, I think it is, is a - no. That's an email protocol. There are a couple free cert providers. They don't do any checking. Actually, I'm not even sure that it's recognized now by web browsers.

Leo: No better than a self-signed cert.

Steve: Exactly. Exactly. So that's not going to help you. No, I mean, it really does say we're going to make you pay in order to have comparable page ranks. And I don't know, that seems bad.

Leo: Yeah. It kind of bugged us on the panel, too, on Sunday. Hey, so I got my official membership at Hold Security. And so now here's what happens. You have to enter in passwords, and they'll see...

Steve: What?

Leo: Yeah. And they'll see whether the password is in the database. It says, "We'll never ask you for passwords," except that's what it looks like they're doing in the form. "The form below provides you with the ability to encrypt your passwords using SHA-512, which makes it impossible to decode. Once you receive the hashes of your passwords, we'll compare them with the hashes we have and notify you." Well, that's okay, I guess.

Steve: Yeah, except I don't like this guy.

Leo: It feels funny. I'm not going to give him any passwords. So that was - I thought the whole thing felt a little odd. And by the way, Brian Krebs is on his web page as an advisor. So I just - the whole thing is...

Steve: Yeah, I mean, I think he's legit. But it just seems like a lot of hoop to jump through. And now he wants to, I mean, because he's managed to leverage this information, which he has the right to do with what he chooses, he wants to make money.

Leo: And he's now got my email address. I had to give it to him.

Steve: Oh, that's right, in order to get...

Leo: Yeah.

Steve: Uh-huh. Okay. So there was an interesting piece of news that I just thought was - that I was led to do some research on, that indicated that Apple's iOS 8 change, where they were going to start randomizing WiFi MAC addresses that we talked about, resulted in a third of a company's employees being terminated. And it's like, what?

And Recode picked up on this. It's a company called Nomi, N-o-m-i, which is kind of an interesting, well, because what they are is they're a tracking intelligence company. And Recode said Nomi, a startup that has raised 13 million in venture capital, has laid off at least 20 of its 60 or so employees, in part because of the forthcoming changes, according to sources. And so I dug around a little bit, you know. And we talked about how a neat feature that would be incorporated into iOS 8 was that, before you had an association, a WiFi access point association between your phone typically, or pad, and that access point, it turns out that it is an industry practice to track people based on their WiFi MAC addresses as they roam around. I mean, that's, you know, with their WiFi turned on. And so there are businesses that are doing this.

And it turns out that's absolutely the case. This Nomi company is installing WiFi systems in retail establishments and selling the service of letting retailers know when people revisit their store, when someone - and then generating reports for which they pay. And so when someone comes back in the store, they'll say, hey, you're getting a certain percentage of traffic, like every two days, every three days, every four days, and so forth. And they provide that intelligence by seeing that the same phone comes back into the store. So clearly this is what Apple is saying no to. Well, what Apple is saying yes to

is iBeacon. And so when I actually dug down, it turns out that Nomi has switched to deploying Bluetooth tracking, which is easier to set up and requires fewer employees. So it's not that they're...

Leo: Awww. Just maybe not the best business model. How about that?

Steve: Eh, that, too, yes, exactly. Yeah. Anyway, okay. Potato chips, Leo.

Leo: Love 'em. Mmm, they're delicious.

Steve: Yeah, they are. That's the salt. So...

Leo: And the fat.

Steve: And the fat.

Leo: Together at last. And the carbs.

Steve: Oh, my god.

Leo: Oh, my favorite three things.

Steve: So MIT has teamed up with Microsoft and Adobe. And at the upcoming Siggraph conference, which is the biggest, I mean, it's been going on forever. Siggraph is I don't know how many decades of years it's been happening. But, I mean, there was Siggraph back when I was programming the PDP-8.

Leo: Yeah, 20 years, at least, yeah.

Steve: Yeah. Or 40. I mean, like, really, it's been there forever.

Leo: Well, it stands for the graphics special interest group.

Steve: Exactly. Okay. So this freaked people out, but there's good news here. So the headline was - it was in a set of experiments, these researchers at MIT, Microsoft, and Adobe developed an algorithm that can reconstruct an audio signal by analyzing minute vibrations of objects depicted in video. In one set of experiments, they were able to recover intelligible speech from the vibrations of a potato chip bag photographed 15 feet away through soundproof glass.

Leo: Wow.

Steve: In other experiments they extracted usable audio from vibrations of aluminum foil, the surface of a glass of water, like the vibrations we've all seen on, you know, like when Godzilla's footstep lands, and we get that - or actually I guess it was "Jurassic Park" that we saw the water vibrating.

Leo: Here's an audio example. This is sound played for the speaker in the room. And then they took video of the plant leaf, which also vibrates. And they recreated it. This is - so they were playing "Mary Had a Little Lamb." Let me see if I can jump ahead to the actual - that's the sound recovered from the video. There's a little noise, but it's intelligible. I heard speech. You can actually detect speech.

Steve: Okay. Now, here's the good news. Nyquist is our friend.

Leo: Ah, thank god.

Steve: Harry Nyquist, yes. Don't think I've ever mentioned him before, name-dropped old Harry. You know how disturbing it is when the wagon wheels appear to be spinning backwards...

Leo: Yes.

Steve: ...in the Westerns?

Leo: Yes.

Steve: Well, that's of course caused by the fact, by the interaction between the frame rate of the camera and the spinning spokes. What Nyquist observed was that you had to sample at least twice the rate of the frequency that you wanted to resolve. So another way to visualize this, imagine a disk, like a black disk with a white dot out on one edge, and it's spinning. And if you are going to take a snapshot of it, how often do you have to get a snapshot in order to have a sense, a true sense for the rotation of the disk.

Turns out you have to do it at at least twice the frequency, or the speed of the disk rotating. Otherwise, you get an aliasing effect, as it's called in sampling theory. So what they failed to mention in the headline is that they had to use high-speed video photography at 2,000 to 6,000 frames per second. And the moment you hear that, you think, of course. I mean, speech is in that frequency range.

Leo: Like 1,000 Hz or somewhere, yeah.

Steve: Exactly. And so...

Leo: 1,000 KHz. No, 1 KHz.

Steve: Yeah, we have our vocal cords, which is generating a high harmonic content; and then our formants, as they're called, through our throat, which form bandpass filters to create speech. So you need to have - so standard webcams or surveillance cameras or anything is 60 frames maximum, often lower. You see, like, still frames with people moving through them because they're trying to minimize the amount of storage that they require. So the point is no normal cameras can be used to eavesdrop on you. The normal cameras, 60 fps, might be able to maybe identify the gender of someone's speech, maybe the number of speakers by doing speaker discrimination within a room. But nothing like eavesdropping on you. It's worth knowing that they can do it with high-speed video.

So again, it might be that, for a surveillance application, this would make sense, although frankly we have much better ways to do that. We would bounce a laser beam off the glass, and we'd use interferometry on the roundtrip distance of the laser beam. And there you get instant real-time speech recovery. So this was a little more of a stunt. Interesting that they did this, but it's not like all of our privacy has immediately been breached by webcams and surveillance cameras.

Leo: Here's what it sounds like with a 60 fps DSLR. Play back the recovered sound. It's the same, "Mary Had a Little Lamb." You get a little bit of something, but not...

Steve: Name that tune. For $400, Alex, yeah.

Leo: It's interesting, though. I mean, it's a fun little hack.

Steve: Yeah. Oh, no, again, I think it's neat that they did it. But the headlines all said, "Your potato chips are spying on you." It's like, okay, you know, just reality check time here on the Security Now! podcast.

LastPass had an outage this morning. They blogged - the blog entry that I saw from LastPass said that at 3:00 something, I think it was like 3:53 a.m., and maybe - I think that was Eastern time, so for us on the West Coast around 12:00 or almost 1:00 a.m.

Leo: I think it was 3:00 a.m. Eastern, yeah, midnight our time, yeah.

Steve: Yeah, that something happened. What they said was that one of their datacenters, which they clearly depend upon more than I think they should, died. And that unfortunately took out, I'm sure, everybody in the datacenter, one customer of whom was LastPass. They're back up. And I'm not sure if, when I used my LastPass, they were already up. It seemed like their website had a harder problem than their background data sourcing. We'll have to see what the upshot is. I'll see if I can drop a note to Joe and ask him for the whole story. Maybe we can talk about it more next week. There was a lot of Twitter traffic...

Leo: Oh, I saw it. Ooph, wow.

Steve: ...chatting about this, yeah, people not at all happy that their cloud service was out. And so I hope that the LastPass folks will strengthen their cloud provisioning in order to make this less important. Of course, and this is a lesson about the cloud. And this is one that we've discussed here all the time is TNO encryption, such as LastPass employs, prevents your data from being decrypted and stolen. But it doesn't prevent the cloud from disappearing.

I mean, so backing up is a great application where - so long as the data is there when you need to recover it. But it's a little more problematical. I mean, the whole cloud model is a little dicey when you need really robust, real-time access, like documents online, like editing documents. I've had like spreadsheets sort of lock up on me in Google Sheets, where it says "trying to reconnect," and I think, oh, please do because I'm using the cloud. And so there's a mixed blessing aspect to it.

Leo: Should you choose to, I should point out, you can back up, and maybe people should do this, your LastPass database. You can export it to a LastPass CSV, which I presume is unencrypted, a LastPass encrypted file. And that you can use in offline mode.

Steve: Yes. And there's also LastPass Pocket, which is an alternative, which is the same thing. And so, yes, I think if we're going to take a lesson from this, it's use the cloud synchronization, I mean, and we've never had a problem before. So this is, you know, this was a multi-hour outage. And that can happen. And obviously logging into websites is very crucial for people. I'll note that SQRL doesn't have that problem because it doesn't use any third-party connection at all. It's just you and the website you want to log into. So we're getting there.

And Tech News Today had an interesting piece I just wanted to call our listeners' attention to. And that was there was another Comcast fiasco that you guys covered this morning, Leo, where only because the caller had recorded his original conversation with Comcast, where Comcast promised that, if he did his own relocation of his service when he was moving, there would be no charge. And so he did that. And everything was fine for a while. And then there were problems.

And so somebody came out from Comcast and fixed, like, outside problems. Nothing to do with his own work and his relocation and so forth. And then he got hit with like, I don't know, don't remember the number, $181 or something on his next bill. And so he contested that, and they refused, absolutely refused to take it off and said there was nothing that they could do and blah blah blah. And he said, "Look, I have a recording of you telling me there will be no charge. Now, you cannot then reverse yourself and charge me. I have you recorded telling me there will be no charge."

So the takeaway, I think, is that you really should record your conversations with Comcast. I mean, that's just, like, for safety. They're so bad, you should record them. And one of the guests on the show said in 18 states in the U.S. there is a law called "All Party Consent," which requires everybody who's a party to a recording to know that. And so I think you should start your Comcast dialogue with the declaration. You should say, "Hey, I just want to let you know, Comcast person, that I am recording this conversation. So you've been notified that this is being recorded. I know you're recording me, so I just

want to let you know, I'm recording you, too, just in case it should ever be necessary."

Leo: Yeah. It's California is what they call a "two-party state." And so I do know the law for our state because of being in radio, you have to know that. The way the rule is, you need only notify them. You need not get them to assent. Their assent is inferred from the fact that they stay on the line. So as long as it's clear, if you just say, "Hey, I'm recording this, now here's what I'd like to do," that's all they have to do. In fact, Comcast doesn't get your assent either. They say "This call will be recorded for monitoring purposes." In fact most calls, Apple does, everybody does this. So it's fine for you to do it. If you live in a two-party state which, as you mentioned, many of us do, it's just prudent to say "I'm recording this" and continue on. You don't need them to say okay.

Steve: Yeah. And I think from the experience that we're having, that we're seeing, it makes sense because, I mean, it can end up being handy to have a recording of their original commitments and promises.

Leo: Even if you don't record it, just say "I'm recording this." It might work.

Steve: And you know, don't we know from the first instance, was it Ryan who had this happen to him?

Leo: Yeah, Ryan Block, yeah.

Steve: Yeah. Don't we know that the employee on the other end has a financial stake in the outcome?

Leo: Yes. Yes.

Steve: And that's what I think is so slimy is that, I mean, they're on the hook for the money they cost Comcast. So of course they don't care about, I mean, can you imagine, it's like a police officer who starts his career, and he's going to go out there and do good, and he's just a neat cop. And after 30 years of on the job, he's just ruined. I mean, his spirit is broken. His faith in humanity is destroyed. And that just must happen to Comcast employees after some length of time.

Leo: Can you imagine? What a horrible job.

Steve: They just, oh, god.

Leo: You remember the original Ryan Block call, the rep tried to get him to cancel at the Comcast Store. So he's really, his one and only purpose was "not on my watch."

Steve: Don't do it on this call, yes. Yeah. So I announced last week that I had just turned the first SQRL code over to the newsgroup gang the day or the evening before the podcast. It went very smoothly. They found a bunch of little UI things, like I had back buttons that were enabled when you should not be able to go back, that kind of thing. Those are all fixed. I just dropped another piece of code yesterday that has the identity export stuff just in file format done. I'm now going to work on the QR code encoding so that I can display the identity on the screen so you can pick it up with your smartphone in order to transfer the identity to your other devices. So we're just moving ahead beautifully with that.

And I did find in my mailbag a nice note from Joe White. And he helped me pronounce where he lives. It's Honea Path [honey-ah-path]. And believe me, from H-o-n-e-a Path, I would not have gotten Honea Path, so in South Carolina. Thank you, Joe. He said: "A resurrection at a funeral home." Okay. And so he said: "Hey, Steve and Leo. Thanks for the show. I've been through the archives and digested them *all,*" he has in asterisks. "I'm a funeral director at a small family funeral home, and being in a family business means wearing many hats. IT is one of my hats. One of our main workstations went down a couple of weeks ago, refusing to recognize its hard drive any longer." And of course we know where this is going.

"I brought in my personal copy of SpinRite, let it do its thing at Level 2, and, presto, workstation began working again. Our massive list of Outlook contacts has been saved from destruction. I've imaged the drive just to be safe, but it appears that the Level 2 scan has restored it to useful service. Our thanks to you, Steve, for turning our mourning into gladness." And then he says: "I think a site license purchase is in order." Which, Joe, I would certainly appreciate.

Leo: Yay.

Steve: So thanks very much for sharing that.

Leo: Yes. All right. I'm ready with some questions for you, Mr. Steverino.

Steve: Yeah, we've got a good mix-up today. I think they're good.

Leo: This one comes from Twitter @NickGustavsson. He tweets: What do you use for a password complexity formula in SQRL? Huh? He's gone beyond my limited means. I need something similar, and I cannot find anything great. What does he mean, a formula to validate a password?

Steve: Well, sort of. The next question we'll talk about how we feel about that. So here I was, and I mentioned this briefly last week, I wanted some - SQRL needs a password, one password, in the same way that LastPass does, which you use to authenticate yourself to SQRL. SQRL will then authenticate you globally and securely. But we need at this point in time, where we don't have, like on a regular PC, a way of making sure that the client knows it's us, we need to log into SQRL, essentially. So I prompt the user, after they create an identity, for a password, which they will use when they want to unlock SQRL, essentially, to let it stand in for them and do their authentication.

So I think password strength meters are currently the best solution. The problem is, what's a strong password? How do we define that? And when I was thinking about this, it's a little bit about that famous quote from the Supreme Court judge who was commenting on pornography, where he said, "Well, I can't define it, but I know it when I see it." And so a password is - originally I was thinking that I would include, for example, a dictionary of a thousand of the most common passwords. The problem is SQRL is going to be supported in 60 different languages. And, boy, that's a project that I don't want is generating the most common, finding the most common passwords in all of those different languages.

So I thought, okay, I need something simple and algorithmic. I have to satisfy myself with doing the best job that a simple algorithm can do. So I have a strength meter, and I run a rather simple formula, which I like because I think it does, as users are putting in a password and experimenting, very much the way the Password Haystacks site operates, it shows you what you've got.

And so here's what I thought of. It's a simple algorithm anyone could apply. When you're all able to play with the SQRL client, I mean, they're playing with it right now over in the newsgroup, so it's working, and they've been playing with this aspect of it. I take the number of characters, so just the overall length of the password. That's one parameter. Then I take the number of unique characters. So having more different characters is better than many of the same. I mean, it's not just black and white, but that's another parameter is how many unique characters you have.

Then I take the characters and divide them into five classes: digits, lowercase "a" through "z," uppercase "A" through "Z," then anything remaining that's less than 128, which will be all the other special characters and control characters, and then if it's greater than 128. That will generally be all of the - everything else in unicode space. So five classes.

Then I count the character class transitions. That is, as you go through the password, how many changes in class? So uppercase to lowercase. Alphabetic to numeric. Alphabetic to special character and so forth. So every boundary between classes gets counted. So that's going to, heuristically - which is what this is. This is a heuristic. Heuristically, staying in a class doesn't buy you as much strength as changing classes, whether it's case or number to special character to alpha and so forth.

And then I have a formula: the total count plus the total number of unique characters plus two times the number of class transitions. And that's my complexity metric, which is shown on a bar graph as the user is entering their password. And it seems to be pretty effective. It's not perfect. I recognize that. But I needed something to encourage people to put in a strong enough password. And in fact you've got to push that bar up to a certain point in order - it changes temperature. As you make a stronger password, it goes from red through yellow and orange into green, changing color and growing in length. And finally, when you get enough, it enables the Submit button. So it won't let you use a really bad password. There is no formal, it must be this long, or it must contain these number of things. That's, as we know, that's a problem because it provides an attacker with an attack pattern if they know what your criteria is. This doesn't have that. This just says, eh, in general, these are the characteristics of a strong password.

And so I implemented that with a rather simple and easy-to-use heuristic, which anyone could do, for example, in JavaScript, if they wanted to help people develop a good password.

Leo: You know what, we've never talked about, I don't think, the famous xkcd horse stapler password...

Steve: Great cartoon.

Leo: ...cartoon.

Steve: Yeah.

Leo: You know, the premise of it is that there's more entropy in four random common words than there is in a shorter but more random password. Is that accurate? Is that good advice?

Steve: No. I mean, it's not a bad thing except that it certainly subjects you to dictionary attack, where you have all those words in a dictionary, and you would try combinations of them. There's a lot of them. But it's still fewer than, I mean, again, it's a tradeoff.

Leo: Would it be better if you added some numbers at the end or used random punctuation between the four words or...

Steve: Yes. Surprise is a good thing, Leo. Think of it as surprise is always useful in a password. So add some surprise.

Leo: I guess xkcd's point is at least you can remember "correct horse battery staple," whereas a random password might not be so easy to remember. But that's what we have LastPass for.

Steve: Right. Exactly. And the problem of course is, once you acclimate yourself to that phrase, you will tend to use it on all your different sites. And that's not safe, either, because the Russians, among their 1.2 billion passwords is that one, with your email address. And if you reused it, they could get into your accounts somewhere else. In fact, xkcd just did something where they referred to haystacks.

Leo: Oh, good. Yeah, because, you know, I think everybody knows that comic. And it's become like lore. It's become urban legend. It's a received truth. And it isn't exactly right.

Steve: But it's better than "monkey." But it's just basically four monkeys in a row.

Leo: Karl in Chicago argues that password, as we mentioned, password strength meters aren't such a good idea: All I'll say in terms of praise is I've been a listener

since Episode 1. That means, Karl, you are at least nine years old.

Steve: Yeah.

Leo: When discussing passwords in the past and their use in SQRL - you know, we refer to SQRL a lot. This is Steve's very clever and now implemented method of logging into a website, not via password, but by something else, a QR code or the like; right?

Steve: Yep. We will, as soon as this thing is done we will do another podcast on it, when I'll actually have it, and you'll be able to do it. You'll be able to...

Leo: It solves this whole password issue.

Steve: Completely. It's over. It's just not a problem anymore.

Leo: The only negative is the sites have to adopt it. We've got to get everybody to do it.

Steve: Correct, correct.

Leo: And that's another challenge, given there's so many sites where they say it can only be eight letters, and there has to be one number. But no punctuation. Anyway, when discussing passwords in the past and their use in SQRL, it seems like you've generally settled on the idea of having a meter to evaluate password strength. That's one of those onscreen little bars that goes green when you've got a good enough password.

Generally, I'm not a fan of these for the following reasons: One, I think it probably gives users, especially uninformed ones, a false sense of security and focuses their attention on getting a green bar rather than understanding what a strong password is. I'll actually vouch for that. If I don't get a green bar, I keep going. I think it's extraordinarily difficult to determine if a specific password is actually strong. What may seem like a good password, for example, "1000_Elm_Street#501," might be easily crackable if it turns out to be based on your address. Or maybe it was generated using a heuristic that could easily be deduced or has been seen or used elsewhere. The simple passage of time may also weaken a password as hardware continues to advance and crackers continue to refine their knowledge from compromised password databases. That's the chief issue with that billion password database.

Steve: Yes. Oh, what a trove of intelligence.

Leo: Yeah. And it's not so much that they got your password, it's that they understand much better how passwords...

Steve: They've got everybody's passwords.

Leo: They've got everybody's. So while we know the qualities of a strong password, it's a difficult thing to quantify. As you said, I'll know it when I see it. Ultimately, the strength of a specific password boils down to one thing: How long does it take to crack it? And, obviously, we can't know that unless we try. Consequently, unless we're prepared to run a user's password through today's current password-cracking programs, and perhaps continue to do so periodically, I think any certification of password strength will ultimately be just too simplistic to be effective. So here's what I've settled on. One, enforce a minimum character set - upper and lowercase, for example, numbers and symbols. Enforce a minimum password length - 12 characters or more, for example. Provide simple guidance on what makes a good strong password - length, randomness, and uniqueness. Your thoughts?

Steve: So I don't disagree with any of that. And I recognize that, as we've discussed, and as I was just talking about, a password strength meter is a tradeoff. Essentially what Karl is suggesting here is the only way to test the strength of a password is to try to crack it, that is, use what a bad guy would use in order to crack a password and see whether it stands up to attack. Which would be nice, if it were practical. But it's not. And in many places I don't think it's actually effective to try to teach somebody, on the fly, what a good password is. I mean, so I think a site that enforces a strategy probably does that, exactly as you said, Leo, where you focus on getting the bar green. It's like you just, you know, you do enough until it's satisfied.

Leo: Just keep typing "monkey" till it goes green.

Steve: Exactly. So it's like I completely agree that it would be possible, for example, to arrange a weak - well, actually I'm not sure that it's possible to use what I would consider a weak password with my algorithm because a weak password would have to be really long in order for only the length to get you to strength because changing character classes, that is worth two points. And unique characters themselves are worth another point, in addition to length. So I think I came up with a good formula. I mean, again, it is a heuristic. But you're going to have a tradeoff because you're trying to help the user protect themselves, but they're not going to stand there and read a book about the science of password generation in order to do that. They just want to get done. And so it's a tradeoff.

Leo: I really like, and people should rewind and listen to your suggested algorithm. I think you nailed it. Nothing's perfect.

Steve: I think it's good.

Leo: But I think you got it. And you can tie that to a green bar.

Steve: Absolutely. Oh, yeah. And I have. I mean, as you're doing it, the bar is growing in length until it's long enough. And you can experiment with it, and you see the length change as you go.

Leo: You've been typing letters, you suddenly type a percent sign, the bar's going to jump.

Steve: It does, yes, and go, ooh, ooh, we like that.

Leo: Try another percent sign. Not as good. Hmm. How about a number? Ooh, that worked.

Steve: Yup, exactly.

Leo: Or just get LastPass, for crying out loud. I just, you know, it's all solved if people just get a password manager that generates password. That's the whole thing.

Steve: For now, that's the solution. And the reason I don't feel badly for, like, LastPass relative to SQRL is, as you said, Leo, SQRL's not going to take over the world overnight.

Leo: You'll still need it, yeah.

Steve: It needs to exist. It will start getting adopted. And the key, the cool thing is users only need one copy. You just get it. And as you encounter sites that support SQRL, you can use it. And if they don't, then you fall back to LastPass or whatever password manager you've been using. So it can happen incrementally.

Leo: I wish you could incorporate - I guess you don't need to - SQRL into LastPass. Somehow you need a flag in LastPass to say, oh, it's a SQRL site.

Steve: Yeah. And Joe might very well build it in.

Leo: I bet Joe will.

Steve: It's simple to do.

Leo: Why wouldn't he?

Steve: Yeah.

Leo: Phil M. in Los Angeles, the City of Angels, wonders about best practices for password management: Steve and Leo, love the show. My question is about the frequency with which passwords should be changed. After that announcement about the billion passwords, many news outlets recommended that we change our passwords on all our accounts. I think they had the boilerplate left over, and the type in the teleprompter. They just kind of pasted it back in. Since no specific companies have been named in this breach, what's your recommended action? How often should passwords be changed as a best practice? That's a good question, Phil.

Steve: You know, it is a good question. And we've talked about it in different contexts through the years. I've never really understood the logic behind enforced password change. That suggests that there's some long delay between the time that a password escapes your control and it's used. Now, I've seen this in spam. So I don't think the same model applies. I've noticed that old email addresses, which I often still have forward to new addresses, they get spam. But the new ones don't. So there does seem to be some weird multiyear delay between where it takes things to finally filter out into spam lists. And so if you are changing your email address occasionally, you can stay ahead of that. That seems to be effective.

Changing passwords just - that seems so burdensome that I don't understand the logic behind it. I know that we've talked about it, like many companies have a, oh, every six months you must change your password. And it annoys their employees so much that the employees go to great lengths to circumvent that enforced password change because it doesn't make sense to them, either. And they've memorized their password, and now they're being forced to change it for no good reason. So I don't know.

Leo: It actually encourages bad passwords, I think.

Steve: It does. I agree.

Leo: You add a one to your password. And we've discussed this ad infinitum. We don't need to continue on. But there are certain circumstances where you would do that. And I think people just saw that and said, oh, well, a bank does it, I should do it. Not necessary.

Steve: No.

Leo: But he asked two questions. One was do you recommend changing your password because of this breach.

Steve: Wow. You know, I feel a little bit like the attorney who always gives advice that

errs in the direction of caution because why not?

Leo: It couldn't hurt. It couldn't hurt.

Steve: It's not your time the attorney is spending. I mean, it's not his time he's spending, it's your time.

Leo: With Heartbleed we did recommend people change their passwords; right? On Heartbleed-affected sites?

Steve: Yeah. And certainly, if this was the typical database compromise of a given site like Twitter, it's like, oh, change your Twitter password. I mean, that seems like clearly there the amount of effort you go to is going to return safety in measure. But telling people, "Change every password that you have on the Internet?" Oh, wow. You know? Okay, yes, do.

Leo: That's what it would have to be.

Steve: If you want to be safe from this, if they might have your passwords - and we know nothing about which ones they have. I guess you could go to Hold Security and give them all your passwords.

Leo: No.

Steve: Yeah, I know.

Leo: I'm not going to do it.

Steve: No.

Leo: By the way, they only have room for, like, 20. I have one for every site I go to. What am I supposed to do with that? That's a useless form.

Steve: I know, yeah, yeah.

Leo: You know, it doesn't hurt to change your bank password every once in a while, though really their advice always comes back to use a password vault. Use KeePass or Dashlane or LastPass, which is what we recommend, or 1Password. Then changing it is kind of trivial. So you could...

Steve: Still it's a pain.

Leo: It's a pain. But, I mean, if you're worried, change your bank. You don't have to change everything. I'm not going to change my New York Times password. But you might want to change your bank password.

Steve: I think, there you go, that's it. Perfect. Change the critical ones. and especially if you have not adopted the process or the practice of using separate passwords for important accounts. If you're sharing them across sites, so that when you bought a toothbrush at Joe's Wooden Sticks site, and you used your same credentials that you use for banking, and Joe's Wooden Sticks site might have a SQL injection vulnerability because they're just not that security conscious, the problem is with this massive breach and the nature of the way these passwords were exfiltrated, the lower security sites are leaking potentially high security credentials. So exactly as you said, Leo, do it for the important sites, for the crucial sites.

Leo: And another good reason to use a vault is so that you don't use the same password. That's a nice feature of LastPass. It'll audit and say you're using the same password on this site as other sites. Do you want to change it? That's great. And sometimes I don't care. But most of the time when it says that I change it. And I go, okay, yeah, that's a good idea. Not a bad idea. Omega Project actually has the best solution. He says change your bank every 60 days.

Steve: Sorry, no money there.

Leo: That'll work.

Steve: You're welcome to log in. You won't get anything.

Leo: Confuse the criminals. Just keep moving. DJ James in Maryland with a personal anecdote about USB infirm-ware: Your report reminded me of problems several years ago when buying memory devices from eBay. Unscrupulous sellers would change the firmware on MP3 players or USB memory products. For example, they would take a 2GB MP3 player and have the firmware report 4GB or 8GB. Then they'd sell it at a premium price. The player works fine until you load enough music to cross the 2GB boundary. And then, of course, files get corrupted as new files overwrite pieces of old ones, and the whole thing's a mess. I was burned by one of these players, he says, and by the time I discovered the problem, the seller was long gone with my cash. I dismantled the unit and discovered the fraud by noting that, sure enough, the part numbers on the memory chips allowed for only 2GB of memory. I found a firmware modification tool on the web and changed the size back to 2GB so I could use the device as intended. Wow.

Steve: Yeah.

Leo: So this guy would have known about this exploit.

Steve: Yup. And I thought - so there were some interesting things. First of all, it has been done. It's been done to abuse people, and there's a firmware mod tool floating around that allows that to be changed for some devices. So there it all is.

Leo: Kind of is my point last week, which is that, if we had but thought about it, we would have realized BadUSB was always a problem. I mean, it was just like, oh, you mean...

Steve: Fundamentally a problem.

Leo: Yeah.

Steve: Because USB is so powerful. We are just trusting something, we're trusting a computer that we're plugging into a port with lots of privileges.

Leo: Right.

Steve: Duh.

Leo: Duh. And it's got writeable firmware. Software writeable firmware. Phil Zeman's ports are no longer stealth, he's glad to report from Wisconsin Rapids, Wisconsin: I recently moved and had to switch from a cable ISP to DSL. All of my ports except 113 were stealthed with the cable provider. Almost all ports are closed with the DSL provider. I'm using the same router as it was set up for the cable ISP. Anything I can do to get back to stealth, or is closed good enough?

Steve: Okay. So here's what's going on. We've talked about port 113 in years past. That's the funky port which your IP address needs to at least respond to in order for some old protocol - some FTP servers and, I think, IRC servers will try to do a lookup on 113 when you're attempting to log into them. So they just sort of do a reverse ping to see if you're there. If that is stealthed, you can have problems logging into those servers. So that's why that single port wasn't. That was actually probably Phil's router that was smart enough to handle that correctly.

And I remember back in the old days ZoneAlarm was smart enough to handle the 113 non-stealthing. Also, actually, they did adaptive stealthing. They would stealth port 113 unless the firewall saw that you had an open dialogue with the IP where the 113 test packets were coming from, and then it would respond, which I thought was technical ZoneAlarm or Zone Lab's brilliance back then.

What's happening here with the switch from cable to DSL is that, when you had a cable connection, the packets coming in were actually hitting your router. If not changing your router has switched you from stealth to closed ports, those packets are not actually getting to you, Phil. Your ISP, this DSL ISP has some equipment which is responding to those packets on your behalf. So there's - who knows what they're doing. They might have a NAT router. You might be NATed, like in the DSL network, so that there's another router between the Internet and you. That's actually a good thing.

So I would imagine, for example, when you're using ShieldsUP!, that ShieldsUP! is not actually seeing your IP, the IP of your home router. It's seeing the IP of a NAT router where the packets are hitting. It's that NAT router which is not stealthed, it's closed. So, eh, you're fine. I mean, it really doesn't - it's not you whose ports are closed, it's some piece of equipment upstream of you that is responding to GRC's ShieldsUP! probe packets. And so it's just a different configuration.

Leo: And nothing to worry about.

Steve: Yeah.

Leo: Ashley Black in Reading, Berkshire, United Kingdom - or is it "Barkshire"? - knows why VPNs get Netflix working: Hi, guys. I've been thinking about this for a week or two. Following on from the Level 3/Verizon spat, and why users say that Netflix is being throttled because, when they try it with a VPN, it works better, here's what I think is actually happening: As everyone now knows and agrees, for whatever political reasons, the Verizon/Level 3 peering connection is congested, maxed out. So Netflix is degraded. Then a user starts up a VPN, which connects to the VPN service provider over a DIFFERENT peering link that is not congested, and Netflix works from their VPN provider's uncongested connection. So the upshot is the user thinks that the ISP is throttling Netflix, when in reality you've just basically rerouted Netflix traffic to another provider without a congested path. Regards, Ashley Black, IT security consultant and long-time listener, SpinRite evangelist, et cetera, et cetera. That actually makes sense. That's great.

Steve: And I think - I'm sure he's true. Many people have asked, and I've just never had an opportunity to mention that, if you can change the routing of your packets so that you're not going through that pinch point, you'll be fine. And many people have noted that, when they use their VPN, Netflix works fine. And I'm sure this is why. You've connected to - you've gotten out of the network through a different peering connection. Then your traffic turns around and wants to go back into Netflix, and it doesn't do it through that Verizon Netflix pinch point. So problem solved.

Leo: Yeah. Yeah. That makes sense.

Steve: Yup.

Leo: But it could also be, and this is with the original presumption that your ISP is throttling intentionally Netflix traffic, as opposed to congestion, that they're actually saying, oh, we're going to turn the knob down on this.

Steve: True. And the VPN tunnel would prevent them from knowing that's what you were doing.

Leo: That's what people are presuming is that, oh, the deep packet inspection's not happening because I'm in a tunnel. And so whatever they're using, Sandvine or whatever, isn't working anymore.

Steve: Right.

Leo: It could be either one.

Steve: Yup.

Leo: Gregg in the United Kingdom wonders about TNO cloud storage: I've been a listener for a few months, and the show has been a breath of fresh air away from the media hype of the sky is falling, burn your computers, and live in a cave to be safe. I'm interested...

Steve: Well, that might still be a good idea.

Leo: Actually, I think, if you listen to this show and you really understand what Steve's saying, you might feel that way. I'm interested in encrypted cloud storage solutions. I'm currently using Google Drive with Boxcryptor 2.0, free, as it suits most of my needs and seems to be TNO. However, the free solution doesn't encrypt filenames. I'd be more than happy to pay for the product if it saves me from stress. 28 a year? No, thanks. I'm really looking for either a one-time purchase or a cheaper subscription cost.

I was thinking about using TrueCrypt for a while. But now I've started to use Boxcryptor I'm looking for something similar. I have been thinking about this, and I wonder, does it even matter? Filename encryption would be nice, but isn't that similar to LastPass not encrypting the names of the sites you've stored in your vault? Yeah, an attacker can get the names of the files. But provided you have strong credentials, they have no way of accessing the info. I'm going away to university next month to study computering. Computering. So it'd be great to have a solution by then so I have something in place to store my coursework. Any ideas?

Steve: So I actually do have an idea. And it relates to Boxcryptor. I continue to look at Boxcryptor and come away very impressed by the breadth of compatibility and the features. But I feel exactly the way Gregg does about having to pay an annual fee for something that I purchased one time or want to use. The whole software as a service model just chafes when I'm just using a piece of software. Turns out Boxcryptor 1.0, which they now call "Boxcryptor Classic," is available; and, with a one-time purchase, you can get filename encryption. The free version works and does not do filename encryption. So that's their way of like pushing you to pay them once. I'm coming around to thinking that's sort of the sweet spot. There are all sorts of very cool features that Boxcryptor 2.0 will do for an annual fee. But unless you really need them, then I think this makes a lot of sense.

So I would say take a look at Boxcryptor Classic one-time purchase if you want filename

encryption. And again, I agree with Gregg that, if your data - you have to decide. If the filenames of what you're storing you feel are sensitive, then buy Boxcryptor Classic for a one-time fee. If you don't care about your filenames being leaked, but the contents is safe, you can use the Boxcryptor Classic for absolutely free. But take a look at Boxcryptor. I need to make time to do a survey. I'll probably do that as soon as I've got SQRL running so we can talk about these sorts of solutions. But I'm impressed with everything I see from Boxcryptor.

Leo: This is was you call Pre-Internet Encryption, or PIE.

Steve: Yes.

Leo: And it would work with any cloud-syncing solution because you encrypt it, and then it syncs the encrypted file, not the...

Steve: Oh, exactly. In fact, they explicitly support, like, everybody. I mean, Google Drive, Amazon, OneDrive, TwoDrive, ThreeDrive, all the drives.

Leo: And what about SpiderOak? I mean, isn't that Trust No One encryption?

Steve: It is, but it's a service.

Leo: Oh, you pay for it.

Steve: So they're providing the - yes, exactly. I just like - I like the separation. I like the idea of separating - oh, and also Boxcryptor is all cross-platform - Mac, iOS, Android, PC. And so you have access to your cloud hosting from all of your platforms. So that's becoming another important thing.

Leo: That's nice, very nice.

Steve: Yes.

Leo: BoxCryptor. All right. Moving along, I hear the trucks, but they haven't been around in a while.

Steve: No, it's interesting, they've been coming later, after the podcast. And when that happens, I always think, whew, we made it.

Leo: Phil Forrest, Auburn University, Alabama, wondering about BadUSB, asks: Why no signed firmware? Steve, I've been reading about the BadUSB firmware attack.

Seems like this would all be prevented by a signed firmware infrastructure. Furthermore, wouldn't signed firmware also thwart hardware-based USB keystroke loggers? Thanks, love the show. Phil Forrest, IT Manager, College of Sciences & Mathematics, Auburn University, Alabama, "War Eagle."

Steve: So we kind of talked about this before at the top of the show. And I guess, yes, it's important to understand that the history of USB predates this concern about security. It was all designed 20 years ago, and it was - what was it replacing? I guess serial and parallel ports was the only thing that we had then. And USB came along, and it was like - I remember, you probably do, too, Leo, that I think it was at Comdex where they had on a huge table 256...

Leo: All at once, yeah.

Steve: ...like all in hubs and keyboards and all kinds of...

Leo: I do remember that, yeah.

Steve: ...cameras and everything. And it just stunned us all. It was like, my god, this is fabulous.

Leo: We were thrilled, yeah.

Steve: And so worrying about firmware signing was like the furthest thing from our mind. The fact that it worked at all was, you know, where can I get it? So now we have a standard, vastly supported, and unfortunately it's not secure. I think the solution is non-writeable firmware and companies explicitly making it clear, the way Yubico has, that our stuff cannot be written to, so you're safe from that. And if this takes hold, it's going to end up being a competitive advantage. Unfortunately, we'll have to take their word for it because they could have proprietary means for writing to their firmware rather than using the device firmware upgrade spec. So it's a mess.

Leo: Yeah, it's kind of sad.

Steve: Yeah. But it's the nature of - that's why the cave does sound like a good alternative.

Leo: I think it was Windows 98 didn't support USB natively, but SE did. Right? Something like that.

Steve: You're right. I think there was like an update.

Leo: Service pack? Oh, okay.

Steve: Yes. There was an update that added that. And it was like, oh, this is great. That's one of the reasons I moved from NT to 2000 was that I was still on NT4, and 2000 had better support for USB.

Leo: Right, USB.

Steve: [Growling] Okay, fine.

Leo: So IronKey, by the way, is saying we do require that firmware be signed.

Steve: Good.

Leo: But I'm trying to remember from this BadUSB presentation at Black Hat, seemed to me that that wasn't enough to make it secure.

Steve: I think it's not enough. I mean, I'm glad that they've given it some attention, but...

Leo: The problem is that the firmware itself reports whether it's signed.

Steve: Exactly. Exactly.

Leo: So if you modify the firmware, among other things, what you're going to say is, oh, and if anybody asks, we're signed.

Steve: Yeah.

Leo: So that's the real problem is that the communication is managed by the firmware. So if you've compromised the firmware, all bets are off. That's why an antivirus can't detect it, because the firmware says "No viruses here."

Steve: Well, and an antivirus doesn't have access...

Leo: Can't even see it, yes.

Steve: ...through the USB interface. It sees a hard drive or a keyboard or something. It doesn't know how to look behind the curtain.

Leo: If the device itself has a certificate, and the firmware you're trying to install on it is not signed, will that block it?

Steve: There would have to be - you'd have to have a non-writeable core.

Leo: That's it.

Steve: Yes. You'd have something that cannot be changed. I mean, the model is exactly like iOS, the notion of bootstrapping. You have a bootloader which is in ROM or absolutely non-accessible, non-rewriteable volatile storage or nonvolatile storage, but cannot be changed.

Leo: Otherwise you just zap the key.

Steve: Yes, exactly. So it cannot be changed, and then that bootloader looks at the firmware and checks the signature and verifies that it's been signed. Then the bootloader jumps into the relatively volatile code and executes it.

Leo: Now, IronKey is saying, oh, no, our keys are in hardware. But that's not sufficient. The software that does the checking has to also be in hardware.

Steve: Yeah, otherwise, you're right...

Leo: It's modifiable, and it doesn't matter.

Steve: Yeah.

Leo: So this is the problem is these companies may not be responding fully candidly.

Steve: I know. And it's...

Leo: For obvious reasons.

Steve: Exactly.

Leo: Harrison Ward in Flower Mound, Texas offers his Home Guest network/router solution. We were talking about the fact that EFF and others, in fact Steve Jobs had this idea, all routers should have a guest mode. Why not just use a system like pfSense [pfsense.org] with dedicated NICs and networks? I've been using this

forever and love it. Have myself multiple networks so I can monitor, set up separate rules, one for my home network systems, network of things - thermostats, light bulbs, et cetera. Oh, I get it, the idea of having separate networks, VLANs I guess, for different kinds of traffic - media networks, and then a guest network with capture portal and port blocking and a DMZ network for anything shared with the world. I know this is a much more complicated setup, but it does allow for segmentation and high security. We use that here at the Brick House, but more to avoid collisions because Ethernet is collision based. You have VLANs for all the different kinds of traffic.

Steve: Right. So I just wanted to acknowledge pfSense. Many of our listeners at the higher end, Linux hack-y sort of listeners, have said, "Hey, Steve, what about pfSense?" And it is a beautiful firewall. You take a PC which you're no longer using because you've outgrown it, it's only got one core instead of 25, so you give it a couple NICs and basically build yourself a little hardware firewall router appliance. The reason I don't normally talk about it, the reason we were talking about knitting together the blue plastic boxes to sort of cobble together a solution like this is that this is much more high-end network expertise, firewall rules and port ranges, and as you said, Leo, VLANs and so forth.

So by all means, if a higher end listener wants really a power solution, pfsense.org, load that onto, like I said, a PC that's no longer your primary, just sits in the corner and routes your traffic, you can have, I mean, if you're into network technology, you can have never-ending amounts of fun with something like this.

Leo: Oh, yeah. Oh, yeah. And remember our first sponsor, Astaro, I mean, you could have set up all of this by downloading the free Astaro software.

Steve: Precisely, yes.

Leo: Putting it on an old PC, creating all of this stuff. And I'm sure people do. But that's a lot of work.

Steve: Yeah.

Leo: And expertise.

Steve: It's a whole different deal. And so I just wanted to say, yes, pfsense.org, absolutely, for the high-end networking guy.

Leo: Right. Here you go. Joe Rodricks, our last question, in Massachusetts wonders about home infrastructure for the future: Steve, this is a bit out-of-band, but I was hoping to get your thoughts. I'm working on building a house. It's 400 feet from my parents' home, and I'm looking into linking the two with fiber. It's surprisingly affordable and doable for a single gigabit link. But what would you do for the actual

LAN inside the home? Do you think it's worth running Cat, or will radio and future radio specs keep up with future needs? I assume 4K TV over the Internet will happen eventually, in fact Netflix is already doing it, and who knows what else is coming. Security aside, is wired still better than wireless?

Steve: You know, I'm feeling my age because I am amazed by the bandwidth that we're getting now with wireless. I mean, I just - I remember when we were at one megabit, thinking, okay, yeah, they can't get a megabit through the air, no. That's just not - that's not going to happen. And it's just crazy. Where are we now, 300Mb, Leo, on...

Leo: AC, yeah, I think that's 300Mb.

Steve: Wow. That's just - I just - I'm stunned.

Leo: And it's got beamforming, so that's really cool, too. It aims at whatever's talking to it.

Steve: Yeah, it's got the whole - the MIMO, the multiple-antennaed beamforming deal, phasing the individual members of the antenna so the nodes cancel out, and it's strengthened only in a certain direction. It's just like, oh, it's amazing to me. So I think, Joe, my feeling is I'm past the point of being stunned by radio. And I just think the convenience of it, we now have security protocols that make it as secure, secure enough, given a strong password. I'm old-school. I've got wire running through. If I were building a home, for example, Mark Thompson did, and he plumbed the entire place for Ethernet cable.

So, I mean, the idea that you're a building a home, and you have the opportunity to run wire, boy, I'd at least run wire to the important areas. Put down Cat6 from, like, a closet to where you think your entertainment system will be, where you think your computer will be. So you take advantage of wireless for where you need mobility, but take advantage of wired where you're able to know in advance where your major bandwidth, your non-mobile bandwidth consumption will be, like your entertainment center, your home theater, your office and things. You really can't pass up the chance of wiring those. But, boy, wireless is a great fallback.

Leo: Well, the good news is you can always add wireless later. If you're building a house, you don't have to do anything to prepare for wireless except maybe not put too much metal in the wall. But you're not going to add Cat6 later. We use Cat6 for 10Gb here, by the way.

Steve: Wow.

Leo: So Cat6 is still relatively economical, even compared to fiber. And fiber, remember, you've got to get switches. You have to have more than just the fiber. You've got to get the expensive glass switches.

Steve: Yeah. He's talking about just a fiber backbone between the houses.

Leo: Well, but you still have to have a fiber switch at each end.

Steve: At each end, yes. True.

Leo: And that's lots of money. So I would do Cat6. In fact, what I really would do - here's what I would do. Conduit. Conduit with Cat6 in it. But if the day comes when there's Cat16, you can just attach to the end of Cat6, pull it out, and rewire. So I would put conduit everywhere, and I would put Cat6 for now.

Steve: And then also leave some nylon cord...

Leo: Fishing line, right yeah, to pull it through.

Steve: ...through the conduit, coming out of each end, so you're able to pull things.

Leo: Yeah, conduit is really the only way to future-proof it. The other thing is, no matter how fast wireless gets, wired will still be faster, no matter what; right? I mean, the same - I think.

Steve: I have to think, I have to believe that. I just can't believe what is going through the air. But also just integrity. I mean, wireless is fundamentally jammable, for example, and just I have a hard time believing what they're able to do. But wires, wires I can believe in.

Leo: Yeah. Yeah. But I was really thrilled to learn that the Cat6 we put in here three years ago can carry 10Gb. We're abandoning our SAN and the fiber that was leading to the SAN and replacing it all with copper using fast NICs and fast storage, and it's great. 10GB is great.

Steve: Wow, that's...

Leo: It's plenty for our editors.

Steve: Listen to us. Listen to us. 10Gb.

Leo: 10Gb. We did a - Russell, of course, wanted to validate it before we did it. He transferred a gigabyte file in a second. One second.

Steve: Unbelievable.

Leo: That's good. And that's hundreds of feet, too. I mean, that's a pretty big throw. So we wanted to make sure that the longest - I don't remember what the longest throw was. But we wanted to make sure the longest throw could handle 10Gb. And yes, in fact, we could.

Steve: So I wanted to wrap this with just a quick comment. It's been on my mind after looking through the list of presentations at Black Hat and Def Con, and in the wake of the Snowden revelations, and knowing what it means for today's actually delivered security to be as soft as it is. I've talked about, I've used the term "porous" before, where you look really carefully at OpenSSL, which so many sites are using, and you find mistakes. You look really close at pretty much anything which is sufficiently complex, and you find mistakes. I mean, that's the history of this. They look at automotive security systems and the networks now that are operating our cars, and they find mistakes that allow them to open the doors and roll down the windows and apply the brakes and deploy the air bags, unfortunately. Mistakes.

And then you look at the NSA and their budget and their interest in finding these mistakes. And I think we're fooling ourselves if we believe, despite the fact that the math is absolutely perfect, it's the implementation that is the problem. Mistakes. Little things like, ooh, look, that random number generator we're not so sure about. And they may have helped that get chosen to be the standard. Over the last few years we've seen example after example.

And so I just wanted to wrap this ninth year of the Security Now! podcast where, over the course of these years, we've really sort of developed a much, I think, deeper sense of reality. Some people want to go hide in a cave. Others of us are going to go crusading around with SQRLs and keep trying to solve these problems in a simple and secure way. But at the same time, I'm just betting that, inside the depths of the NSA, they're not worried because they understand that their ability to focus on the porosity of the security we're actually implementing in the field gives them an edge. They may very well have already found the mistakes, for example, in BadUSB. There's some supposition that Stuxnet may have been propagated that way. I mean, they could certainly have arranged to have compromised firmware in drives that everyone just assumes they're fine, if they're able to physically get them into the channel somehow so that they drift into those factories.

So, I mean, I don't think this is gloom and doom. I think this is a little, maybe a dose of constructive reality. It's that we should not imagine that we can really raise barriers that a super well-funded, determined attacker with good intentions, arguably, like the NSA, are unable to penetrate. I think they probably can. I think it's worth us continuing to try to close the pores in security. Clearly, we're finding mistakes, and we're fixing them. We are learning. We are way better today than we were nine years ago at the beginning of the podcast, we as a society, we as a network of people.

Unfortunately, we're seeing new things, like the Internet of Things, where people are launching products with half-baked security. But notice how quickly we're now catching up and saying, whoa, whoa, whoa, we need some standards for light bulbs and pasta makers because you can get up to mischief if you have an unsecured pasta maker in your kitchen. So overall I'm encouraged. But at the same time I'm having to acknowledge, given the presentations we see at conferences like Black Hat and Def Con, security is hard. And there really is, there's an inherent porosity to big, complex systems. There generally is a way in.

Leo: It's funny you should say that because I've been talking a little bit about that on other shows, that there is no such thing as a hundred percent security or a hundred percent privacy, and particularly on the Internet. If you insist on either, you're probably going to break the Internet.

Steve: Right.

Leo: In order to do that you'd have to kind of fundamentally undermine what makes the Internet work.

Steve: Well, and look at the cost that we are putting the typical user through by requiring a 20-character random debris password uniquely for every site they go to. I mean, that is the only way we have today to be really safe. And, wow, that's, I mean, I'm completely crippled now without LastPass. I don't know any of my passwords.

Leo: Right, right. Yikes. Yoiks.

Steve: And - I know.

Leo: You know, as we found out with Mat Honan, the Wired editor who got hacked, the truth is, for most of us, if we do get hacked, it won't be because we had insecure passwords.

Steve: Right.

Leo: So have at it. Have great and secure passwords. He did. And it didn't - had he had second-factor authentication on his Google, that might have worked. That might have been enough. So there are things you can do. I don't worry, you know? I don't really worry. And I'm, as you are, I mean, I think we're targets. Anybody who does a show called Security Now! is kind of asking for it. We've got a big red bulls-eye on our back. And knock on wood, I haven't been hacked. You haven't been hacked. I think it's possible, if you operate fairly sanely, and you don't really have a determined hacker going after you...

Steve: I think, see, and that's exactly my point is the determined hacker is like the determined NSA.

Leo: Right.

Steve: We have to accept the fact that we're in an imperfect security world. And we need to decide where's the right tradeoff between the cost and the vulnerability. And so I do things like I've talked about, like having an absolute firebreak in my electronic funds transfer, where I force the physical writing of checks across those barriers. It's a pain for

Sue. But it's like, I just don't want that. I want a firebreak there. And so, but as people have noted, it's inconvenient, and they can't do their electronic banking the way they want to. It's like, yes, I know. But if something gets in your computer and transfers the funds away, then if nothing else, that's a bigger - that's also an inconvenience. So, yeah, I think here we are, wrapping up Year Nine, and we'll have plenty of content for Year 10.

Leo: Amazing. Well, we're wrapping it up. Next week our 10th year begins. Geez, Louise.

Steve: Yup.

Leo: And we've both gone gray doing it.

Steve: We have.

Leo: Steve Gibson is at GRC.com, the Gibson Research Corporation. That's where he sells SpinRite - you must buy it -- the world's best hard drive maintenance and recovery utility. It's like 20 years old.

Steve: Yes.

Leo: I mean, this version isn't, but you've been doing this for a long time. You can also find all sorts of great stuff Steve has accreted on his website over the past 10 years.

Steve: It does accrete.

Leo: Like the pearl in an oyster. Every little thing that irritates him becomes a pearl.

Steve: That's right.

Leo: And you'll find them all at GRC.com, all free, including 16Kb versions of this show in audio, and Elaine Farris's wonderful handwritten, handcrafted transcriptions. On our site, TWiT.tv/sn, we have full-quality audio, video, in a variety of formats. You can go there or anywhere podcasts are aggregated. iTunes, all the best places have Security Now!.

Steve: Or accreted.

Leo: Accreted. Aggregated or accreted. What is the difference between aggregating and accreting? I'll have to think about that. Since we've been around nine years,

that makes us one of the oldest podcasts in the world.

Steve: Still surviving.

Leo: Still-surviving podcasts, that's right. And that means we are everywhere. Everybody knows about us. Stitcher. We've got great apps on iOS and Android and Windows and Roku. And thanks to our app developers for making that possible, all independent, hardworking boys and girls deserving of your support. Steve, we'll be back here when we do this again next time, it's Tuesday, 1:00 p.m. Pacific, 4:00 p.m. Eastern time, 2000 UTC on TWiT.tv. We'll see you New Year's Eve, I hear. That's wonderful news.

Steve: Yup. I'll be back.

Leo: We're doing our 24 hours of 2015 once again. We had so much fun doing a 24-hour marathon last year on New Year's Eve that...

Steve: And we've now recovered.

Leo: Yeah. It took me a week or two or five or a hundred. But, yeah, no, I feel fairly rested after about eight months later. We aren't just...

Steve: Yeah, and in fact you're going to do another whole 24-hour cycle.

Leo: We are. And we've started planning and everything, and we're going to do it for charity this time, which we should have done the first time. So it'll be almost like a Jerry Lewis Telethon. We're going to raise money for Child's Play and some local charities, too. So that'll be fun. And we'll give you more details. But just schedule New Year's Day, New Year's Eve day with us because it'll be almost - it is virtually all day New Year's Eve day.

Steve: It's a lot of fun.

Leo: Thank you, Steve. We'll see you all...

Steve: Okay, my friend. Talk to you next week for the beginning of Year 10.