Listener Feedback #150

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

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-368.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-368-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We're going to talk about the latest security news, including that leaked FBI laptop, and, of course, your questions and Steve's answers. We've got some great questions for you, coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 368, recorded September 5th, 2012: Your questions, Steve's answers, #150.

It's time for Security Now!, the show that protects you online. Steve Gibson, the Explainer in Chief, is here. He is from GRC.com, the creator of SpinRite, world's finest hard drive maintenance and recovery utility, and joins us every week to talk about security. And a good day to you, Steve.

Steve Gibson: Hey, Leo, great to be with you again, as always. After some delays for important news...

Leo: We actually have a Q&A?

Steve: We actually have a Q&A this week, yeah.

Leo: Wow. Well, you know, I almost thought, because there was that FBI database that was leaked yesterday, I almost thought, oh, crap, we're not going to do questions against this week.

Steve: Well, we're definitely going to be talking about that. So that's happening for sure. I have a fun quote from the hackers who posted the list up on Pastebin. And as we know, the FBI is denying that they ever had that data or that it came from that laptop. We don't know.

Leo: Well, who are you going to trust, a bunch of anonymous hackers or the FBI? That's a tough question, actually.

Steve: Neither. None of the above. Is there a third choice?

Leo: I mean, I trust the FBI. But on the other hand, there's a lot of incentive to deny this breach.

Steve: Yes, yes. Strong need to deny it. And although at the same time the hackers could simply be trying to give the FBI a black eye.

Leo: I'm kind of in that - well, we'll talk about it. That's good.

Steve: Yeah, yeah.

Leo: Yeah. And we have a Q&A?

Steve: We have news. We have news that we discussed at length last week with Java. And many of our listeners, probably every one of our listeners know that immediately, I would say the ink was hardly dry, but we don't use ink on the podcast, so the files were still freshly dated when the Java update...

Leo: The bits were newly minted.

Steve: They hadn't really settled down yet.

Leo: No. The electrons were still buzzing.

Steve: And we got another one, which is not right, either. So we'll talk about that and a bunch of other cool things.

Leo: Steve Gibson, let's get the security news.

Steve: Let's do that. So we spoke extensively last week of Java, yet again. And I was explaining, I think, what I liked best about the way I posed things last week was the idea that Java being invokable by web browsers was inherently a problem because it was a

full-strength language, not designed for web clients; whereas JavaScript was designed for web clients. So JavaScript is safer.

Leo: Er.

Steve: Certainly not without its problems. Thus we have NoScript, in order to even control scripting. But Java is very dangerous. And the problem with Java is that, in order to try to control it, they have attempted to sandbox it. And we talked about sandboxing off and on through the years because it's very much like a firewall, where you've got goodies behind the firewall that you want to keep bad guys from getting. Well, that creates an inherent tension. There's a problem because then you're depending upon your firewall to limit access to something you don't want people to have. It's much safer not to have it in the first place. So JavaScript is better in that way.

So what happened was, almost immediately after last week's podcast, the day after, Oracle did release their emergency out-of-cycle update, which wasn't scheduled until the middle of next month, I think October 18th or 16th, if I remember correctly. But they did it immediately because this was a zero-day, two different vulnerabilities that had been reported, I think, more than a year ago, and they had not gotten around to fixing them. So they pushed that out immediately.

And at that point Adam Gowdiak of Security Explorations, who had some dialogue with Oracle as recently as April of this year about those and other problems, he posted: "Hello, all. Yesterday, an out-of-band patch was released by Oracle, which among other things incorporated fixes for the issues exploited by the recent Java SE 7 attack code." And then he says in parentheses, "(ClassFinder/MethodFinder bugs)," which is what we discussed last week.

He said, "One of the fixes incorporated in the released update addressed the exploitation vector with the use of the sun.awt.SunToolkit class," which is where those two methods are. He said, "Removing getField and getMethod methods from the implementation of the aforementioned class caused all of our full sandbox bypass Proof of Concept codes not to work anymore." And so that's the good news.

He says, "Please note that not all security issues that were reported in April of 2012," so about four months ago, "got addressed by the recent Java update. [So] today we sent a security vulnerability report, along with Proof of Concept code, to Oracle. The code successfully demonstrates a complete JVM," which is Java Virtual Machine, "sandbox bypass in the environment of the latest Java SE software, [which is] version 7 Update 7 released on August 30," which was Thursday of last week, "2012. The reason for it is a new security issue discovered that made exploitation of some of our not-yet-addressed bugs possible to exploit again."

So Oracle chose not to fix all the problems that they knew about. They only fixed the two that were being actively exploited, left the other ones alone. And so these guys designed yet another proof of concept to show to Oracle yet again that there's still trouble. Now, this of course exposes us to the same problem, which is further indication that we really do need to start listening to all the security people in the entire industry who are now saying get rid of it. If you don't need it, remove it. I did see some disturbing tweets from people in, I'm thinking Sweden, maybe Switzerland, somewhere like that, saying that the banks over there depend upon Java for all of the online banking.

Leo: Yeah.

Steve: It's like, okay. That's not going to be a problem.

Leo: I think there's a lot of places where you do need Java.

Steve: Yeah. Unfortunately. I mean, it is powerful. It's multiplatform. We need to consider it to be a transitional technology in the same way that Flash was. And where the evolution of the native capabilities of our browsers, represented by the movement to capabilities available in HTML5 from HTML4, and things like those extra services which will be available in JavaScript will increasingly replace these sorts of third-party solutions. And the good news is there is certainly pressure on everyone. In the same way that Adobe and Flash has been in the doghouse because they were not in the sandbox for so long, the same thing is happening with Java. I mean, you can't have the industry advising everyone to uninstall it without that creating some pressure on it.

Leo: Is it as good - people keep asking me this. We talked about it last week. But is it as good to disable it in the browser as to uninstall it? It certainly is better than nothing.

Steve: Okay. If you really need it, then I think the better - a workable compromise is to use a different browser. For example, when you click a link, you'll have your system default browser, whatever you choose for it to be - Chrome, IE, Firefox, whatever. Maybe do not install the plugin for Java in that browser so that your random link-clicking and surfing just cannot invoke Java, but have a browser which you use for playing games or talking to your Swiss bank or whatever it is you're doing. And that browser has the Java plugin so that it's able to invoke Java. That way you've created on your end - although you have responsibility for doing this correctly. The idea would be, though, that the default browser you're using doesn't have Java plugin, cannot invoke Java through the browser. And then you need to, I mean, many of us do that, for example, with IE. I don't use IE for anything except Windows Update that wants to use IE under XP. So that's an example of the idea of multiple browsers, sad as this is to suggest the need for, but multiple browsers to create some isolation.

Leo: And modern browsers like Chrome and Safari will warn you before they run Java. They don't run Java ever without warning. They say I want to run Java, allow or disallow. Right?

Steve: Oh, okay.

Leo: Safari does. I'm pretty sure Chrome does.

Steve: Apple has been making some good inroads from a security standpoint. As we know, it keeps it disabled. It times it out and redisables it. And so, I mean, they've been very proactive on behalf of their users after the catastrophe of hundreds of thousands of

machines getting infected from the prior Java zero-day exploit.

Leo: So tell people also, pay attention. It's one thing if you're running a game, and it says "May I launch Java." It's another thing if you're just at a regular website and it says "May I launch Java."

Steve: Yeah. And so here we are, we're looking at the difference between the common, the typical user, and our audience. Our audience...

Leo: Right, they know.

Steve: ...I can say "Two browsers, one with Java, one without." Not a problem. That same audience will say, ooh, wait a minute, why am I being prompted to run Java? Steve and Leo have told me over and over and over, I could get in trouble when I do that, and here I'm going somewhere where I'm not expecting to run Java. So this audience understands that. But there's just no way that anybody who's just your typical Internet user is not going to get in trouble.

So, I mean, the industry needs to protect users from the dangers of common actions. And, unfortunately, running Java is a common action. Certainly at some point many people have had their browsers say, "Oh, in order to do this, you need to click this button." Most people do. And so, like, it downloads and installs Java. And annoyingly, it also installs extra stuff you don't want to. I can't even believe that, Leo. When I was updating my Java there was an opportunity to install some - was it a security, I think McAfee security scan.

Leo: Oh, no.

Steve: Oh, yeah.

Leo: Thank you, Oracle, you piece of junk.

Steve: Unbelievable.

Leo: Jiminy.

Steve: I know.

Leo: Ugh. Blame Oracle.

Steve: Of course I do. And that's just wrong.

Leo: Yeah, blame Oracle. That's just wrong.

Steve: I mean, they're getting revenue from that. They're upselling their...

Leo: Java's free, but it's always been free. This is part of the deal when you bought Java, you bought Sun. Open source it or something; you know? If you're not going to keep - it's one thing if they would keep it up to date. Then maybe I wouldn't mind giving a little something. So if I go to Free Java Download, oh, it won't let me because I'm on a Mac. I don't even know if...

Steve: Yeah, my most recent Java update, I'm very sure that it was checked by default, and it was "Get a free McAfee security scan." It's like, ohhhh.

Leo: It's so not okay.

Steve: Bad people.

Leo: Bad people.

Steve: So, okay. The big news is, of the week, new news instead of new old news, or old new news or something, anything, new new news is that 12,367,232 Apple iOS unique device identifiers, UDIDs, along with usernames, the name of the device, the type of the device, something called Apple Push Notification Service Tokens, users' zip codes, their cell phone numbers, their physical addresses, et cetera, and personal detail fields referring to people, were leaked onto, well, were obtained by a group of hackers.

So I tweeted, if anyone is curious to see this themselves, I tweeted at SGgrc, so Twitter.com/sggrc, and my very first - my most recent couple tweets, I tweeted just this morning for the podcast. There's a link to the Pastebin page where 1,000,001 of these 12 million-plus UDIDs have been posted. And excerpting a piece from the long dialogue that's there, it says:

"During the second week of March 2012, a Dell Vostro notebook used by Supervisor Special Agent Christopher K. Stangl from FBI Regional Cyber Action Team and New York FBI Office Evidence Response Team was breached using the AtomicReferenceArray vulnerability in" - [cough] wait for it - "Java. During the shell session some files were downloaded from his desktop folder, one of them with the name of 'NCFTA_iOS_devices_intel.csv'" - so a comma separated values file. It "turned to be a list of 12,367,232 Apple iOS devices, including Unique Device Identifiers, user names, name of device, type of device, service tokens, zip codes, cell phone numbers, addresses, et cetera. The personal details fields referring to people appears many times empty, leaving the whole list incomplete on many parts. No other file on the same folder makes mention about this list or its purpose."

Leo: Now, this is all allegedly. I mean, these guys...

Steve: Yes.

Leo: Who knows? These guys, we can't trust them.

Steve: So we don't know, this is true.

Leo: We don't even know who they are, yeah.

Steve: We don't know this is true. This sounds plausible. So does the FBI's denial that they ever had such a list.

Leo: And Apple just minutes ago denied that they gave the FBI such a list.

Steve: Okay.

Leo: Not that Apple would have to give the FBI a list to have such a thing.

Steve: A security consultant in Dunedin, New Zealand, Aldo Cortesi, has been worried about the whole issue of UDIDs for years. In his little "about me" he says, "I hack on things that interest me and run Nullcube, a small security consultancy that provides services worldwide." And I liked his description, so I'll share it. And this is a post from him a year ago, 9th of September, 2011. He said, "A UDID is a 'Unique Device Identifier.' You can think of it as a serial number burned permanently into every iPhone, iPad, and iPod Touch. Any installed app...." Now, this is a year ago, and we'll talk about what Apple has done since and then current state. So a year ago.

"Any installed app can access the UDID without requiring the user's knowledge or consent. We know that UDIDs are very widely used. In a sample of 94 apps I tested, 74 percent silently sent the UDID to one or more servers on the Internet, often without encryption. This means that UDIDs are not secret values. If you use an Apple device regularly, it's certain that your UDID has found its way into scores of databases you're entirely unaware of. Developers often assume UDIDs are anonymous values, and routinely use them to aggregate detailed and sensitive user behavioral information.

"One example is Flurry, a mobile analytics firm used by 15 percent of apps I tested," he writes, "which can monitor application startup, shutdown, scores achieved, and a host of other application-specific events, all linked to the user's UDID. I recently showed that it was possible to use OpenFeint, a large mobile social gaming network, to de-anonymize UDIDs, linking them to usernames, email addresses, GPS locations, and even Facebook profiles.

"This post looks at the way UDIDs are used in the broader social gaming ecosystem. The work is based on a simple question: What happens if we swap our UDID for another while communicating with the network? There are a number of ways to do this. In my case, I used mitm" - which of course we know is an acronym for "man-in-the-middle" - "mitmproxy, an intercepting HTTP/S proxy I developed which lets me re-write the traffic leaving a device on the fly. In most cases this was a simple matter of replacing one string

with another. But two networks, Scoreloop and Crystal, prevented UDID substitution using cryptography. Unfortunately, both networks relied on the secrecy of key material distributed in the application binaries to every device. I have verified that it is possible to reverse-engineer the application binaries to extract the key material and circumvent the cryptographic protection." Okay. But that's a little detail. So he's saying, in general, swapping UDIDs swaps your identity.

"The outcome of this experiment shows," he writes, "that social gaming networks systematically misuse UDIDs, resulting in serious privacy breaches for their users. All the networks I tested allowed UDIDs to be linked to potentially identifying user information, ranging from usernames to email addresses, friends lists, and private messages. Furthermore, five of the seven networks allow an attacker to log in as a user using only their UDID, giving the attacker complete control of the user's account." Full impersonation. "Two networks had further problems that compromised a user's Facebook and Twitter accounts. Crystal lets an attacker take control of user accounts by leaking API keys, while Scoreloop partially discloses users' friends lists, even if they are private."

Okay. So we have a situation where, from day one, our iOS devices have all had a unique ID. There have been apps on the App Store, I've downloaded them, one called "UDID," which you run it, and it shows you your UDID.

Leo: You can get that in the About box, too.

Steve: Exactly.

Leo: You don't need an app for that.

Steve: Well, in this case it makes it simple because what I was doing was I was beta testing apps that were not yet public, and so I needed to provide the app developer with the UDID of my device so he could...

Leo: That's the most commonly used purpose for UDIDs.

Steve: Correct. And so this made it easy because I was able just to put in that person's email address, and it sent a preformatted piece of email, including the UDID, so it was all done for me.

Leo: Otherwise you have to type it in, yeah, yeah.

Steve: Yeah. So what we have is something very much like an Internet or, sorry, Ethernet MAC address. It is a globally unique ID that uniquely identifies the device.

Leo: But, I mean, the phone has at least that, has a MAC address, UDID, and it has an IMEI number. It has numerous unique identifiers.

Steve: Right. And so what's happened is 12 million...

Leo: All phones have this, by the way.

Steve: We presume 12 million of these are, well, and remember these are also non-phone devices, iPods and iPads.

Leo: iPads, too.

Steve: In addition to the iPhone. So anyway, so we have 12 million ostensibly leaked with a ton of user information. There is a site that we've referred to in the past, ShouldIChangeMyPassword.com. And I also tweeted the link to that, where you can get your UDID, put it into here, and it will do a check to see whether yours is in the leaked data. And there are a number of anecdotal posts on the Internet of people who have been active iOS users finding three of their devices' UDIDs among this first million leaked IDs.

So where's Apple on this? A year ago Apple acknowledged that this was probably not a good idea. There was an API, an Application Programming Interface, that essentially allowed any application to ask for the device's UDID. And it was just a convenient token to represent the user. The problem is that it was common to all the applications on the device. So with the release of iOS 5, which is where we are today, which was about a year ago, Apple has formally said that the use of UDIDs is deprecated and will be removed from iOS.

Leo: And since March they've stopped approving any app that uses UDIDs.

Steve: Right. So they're saying "Unique Identifier, an alphanumeric string unique to each device based on various hardware details. Deprecated in iOS 5. Instead, create a unique identifier specific to your app." And so they're saying do this yourself, don't use something global. So this is, I mean, this is not the end of the world. This isn't, I mean, we'll see how this gets leveraged and exploited. These things tend to have some legs. So there may be - we may be talking about this again in future weeks or months, if bad things happen as a consequence of this. But for the time being, I just wanted to bring it to the attention of people who like to follow interesting events in the security world.

Leo: Yeah. And I guess we don't really know what happened.

Steve: We don't. We have absolutely...

Leo: We have no idea. This is all the assertion of a fairly discreditable group.

Steve: Yes. I completely agree. They're saying this is where it came from, with lots of information that sounds plausible. But we don't know. So I'm not...

Leo: Or even what to do with it; right? I mean...

Steve: Well, we'll see. I mean, depending upon what the information is.

Leo: Right.

Steve: If in fact there are networks and apps that simply - that identify and authenticate their user purely with the UDID, then this means anyone who's in this database can be impersonated. That's not good.

Leo: Right.

Steve: So what I would say is, any app developer who is still using this really, really needs to stop right now.

Leo: It would be, I mean, nobody's doing it, as I said, actively, because Apple won't approve you.

Steve: What about apps that have been?

Leo: That's the problem. There's probably a whole bunch of legacy apps that do that.

Steve: Yes, yes.

Leo: And but they have to be updated to stop doing that.

Steve: Yes. And so the takeaway from this for app developers is consider that your users are not safe any longer if you are giving a UDID too much power. And we've seen that many apps do. So developers need to immediately change that behavior, like yesterday.

Okay. This is a bizarre little tidbit that wraps up our security news for the week. A security researcher has figured out that it is possible to put an entire phishing web page in a URL.

Leo: The whole page.

Steve: The entire page.

Leo: It shows the whole page. [Laughing] That's good.

Steve: Yes. I posted a link to the - I think I posted a link to it, also in my Twitter feed. Anyway, so this is - there's an RFC, 14 years old, August of '98. It's RFC 2397. So because of its age, it is completely supported in all browsers. In the same way that you can have http: and then something, like // and domain name and so forth, you can have the word "data," data:, and you then specify the media type, like image/gif, and optionally whether what follows is base64 encoded, to allow you to encode anything which might otherwise have a problem being encoded in the character set. And then you follow it with the content of whatever it is. And in the RFC is an example of them encoding a jpeg image in an image tag.

So, for example, normally, for people who haven't done web pages before, normally a web page would contain an image tag where it would say SRC, for the source of the image, equals, and then http:// and then, you know, so forth. That is the URL for the browser to go fetch the image. It turns out you can use this data: approach instead and actually put the image data right there in the page, so that the browser isn't going and fetching this object, going back and fetching it from the server, but the page itself contains it.

Well, what the researcher recognized was that this would also be honored in a URL. Which means that you could use any of these link-shortening services to have a shortcut which expands to a full-blown web page, and it works. Now, browsers treat this a little differently. They all know about the data: option. Firefox and Opera work perfectly. They allocate as much memory as necessary and interpret this. You could have JavaScript. You can invoke Java exploits. You can do anything you want to.

So the point is, what's different about this is that, traditionally, phishers, p-h-i-s-h phishers, had to have a site somewhere. They had to maybe have it say PayPal with Pa1 instead of Pal, to sort of trip you up. They had to have a hosting server that your browser would be bounced to. Well, no longer. We don't need to have a site at all. We can bundle the entire fake web page into the URL of the browser using this approach. So the good news is IE limits the length of the data: element so that it inhibits what can be done. And Chrome is interesting. Chrome does it, but brings up a security kind of warning saying, um, this is sort of strange.

Leo: That's an awful long URL. Are you sure you meant that? How long can it be? 256 characters? I mean, is there...

Steve: No, 23K.

Leo: 23K.

Steve: Yeah. And, I mean...

Leo: But what does IE limit it to?

Steve: I didn't look...

Leo: Not 23K.

Steve: Yeah, a lot shorter. So that in...

Leo: That's an awful lot.

Steve: In this PDF, in the appendix, they show a complete base64-encoded page. Anybody who is curious could, like, copy and paste it into their URL, and up comes a well-formed, ready-to-go phishing page. It's just incredible. So I thought, okay, that's...

Leo: I'm looking at it. It's a pretty long - because it's base64-encoded, it's...

Steve: Yeah, but, I mean, that's the URL. And Firefox and Opera go, wow, okay.

Leo: All right. If you insist. Here it is. Wow. Look at that.

Steve: Isn't that something.

Leo: Yeah.

Steve: Yeah. And Chrome, you are able to click through Chrome's questioning of this and go, yeah, that's what I meant. And then you get the page.

Leo: Well, I'm going to try it now. Now you've got my curiosity. Wonder what Safari does. It's quite a bit of copy and pasting here. All right.

Steve: Yeah. And it's across page breaks and so forth.

Leo: Well, and I hope I didn't get page numbers in there. Doesn't look like it. It looks like they made it so that you could do that. Let me try it in Safari here. Yup, it opened. Oh. Login to create account. Wikipedia. Looks just like the page.

Steve: That's the page.

Leo: That's it? Holy cow.

Steve: Yes.

Leo: So that it didn't go out and get any of this. This is all in that base64-encoded text. And no warning from Safari at all.

Steve: Nope.

Leo: It just did it.

Steve: And so you use a link shortener and click. That's in email, or it's wherever. And your browser just says, oh, I'm going to display all of this right from the address bar.

Leo: Now you've got me going here. Let's try Chrome. Paste that in. Oh, did I get it? Maybe Chrome won't even let me. Yeah, see, maybe I don't have it anymore. Huh. Well, that's - oh, yeah. You know what, I bet you that, among other things, it cleared my clipboard. Does it? With everything else it did?

Steve: It's running a script.

Leo: Yeah, it's running a script. It could have cleared the clipboard so that you can't investigate. Wow. Nice job.

Steve: Wow. Isn't that cool? I was just like, okay, well, this is unintended consequences. Allow us to put embedded data in anything. So instead of http:, we have data:, and anything you want. Yeah, it's cool.

Leo: Love it.

Steve: So before we began recording the podcast, I mentioned to you, we discussed a little bit, or maybe it was even when you were doing the Audible mention, that Mark Russinovich's book "Trojan Horse" that I referred to a couple months ago and read a couple months ago - Mark was kind enough to provide me with a galley of the book - is now out. I looked over on Amazon, and it's in hardcover, paperback, and Kindle versions. And a great read. It follows on from his first novel with the same characters that he developed there. And what was chilling as I'm reading it, I was thinking about our listeners the whole time, as I mentioned a couple months ago, because everything he was talking about is exactly how this stuff works. And it gives you sort of this queasy feeling in your stomach about, wow, the world really is crazy right now, for lack of a better word. Anyway, I enjoyed the read, and I wanted to let everybody know that it exists.

I also wanted to clarify an acronym collision. We were talking about the Google Authenticator and its support of OAUTH. Well, there's two "o-auths." There's OATH, which is the Initiative for Open Authentication, and that's what I meant when I said "o-auth." But of course I have often also spoken of OAUTH, where it's OAUTH, which is the technology that we have done a full podcast on, where you log in using your Twitter credentials, log in using Facebook. That's the whole using-one-site-to-authenticate-

another technology, which is controversially now at version 2.

There was an interesting posting a while ago from someone who had been really deep in the management of the project, who gave up and pulled up stakes and left the project out of disgust over the nature of sort of committees designing things, and he was completely unhappy with how OAUTH was evolving at that point. So anyway, I guess O-ATH versus O-AUTH?

Leo: Yeah, because I would pronounce it O-AUTH, both. Yeah, that...

Steve: Yeah, I know. It's like...

Leo: It's confusing. I actually didn't know there was a difference.

Steve: Yes, and so I wanted to make clear that these are very different things. There's the Google Authenticator, which I - and the reason I wanted to bring it up, make sure people understand it, is that I think this is going to win. I think the idea of individual accounts in a software-based time authenticator that runs in all smartphones and PCs and so forth and is open source and, I mean, this is a great way of strengthening login. So...

Leo: Yeah, I use the Google Authenticator. I like it.

Steve: Yeah, me, too.

Leo: You have to use it for Google, and LastPass uses it, too.

Steve: Right. LastPass has, yes.

Leo: Love it.

Steve: Yeah. So OATH versus OAUTH.

Leo: I guess it's "oath" versus "o-auth."

Steve: Oh, OATH.

Leo: OATH.

Steve: OATH, yeah, you're right.

Leo: But, see, what's confusing, it's the Initiative for Open Authentication.

Steve: Exactly.

Leo: Which would be OAUTH, but it's not.

Steve: Okay. So many of our listeners have been excited about the Raspberry Pi project. And I think I've heard you mention...

Leo: We've interviewed them, yeah. We have one, yeah.

Steve: Yeah. And of course instantly sold out the first batch and scrambled to make more. It's a very inexpensive, lightweight, very capable platform for messing around with an ARM-based RISC chip. Well, Cambridge University has a very nice-looking set of free open courseware for taking someone from the very beginning and learning low-level coding in the ARM's native RISC assembly language to build a small operating system. Which is really neat.

And so one of the shortcuts in the tweet that I tweeted this morning is the Ras-Pi RISC coding tutorial. Anyone who's interested go to, again, Twitter.com/sggrc. You'll see my tweet there. That'll take you over to the University of Cambridge page. And it's a few levels in because I couldn't really find a way to get in there through navigating. So I thought I would drop you in because you can easily back yourself out using their little "You Are Here" hierarchy menu up at the top of the page because there's lots of stuff.

It looks like they're standardizing on the Raspberry Pi themselves. Every student who is enrolling in computer science gets one, and so they're assuming their students have them, and they're developing a bunch of courseware around it. And it's all free and on the 'Net. So I wanted to point our users to it: bit.ly/RLsoyj. Our listeners.

And I did just want to make a mention that - I meant to mention this last week, but I forgot, Leo, my discovery that 3D, that is, television in your home, is real. I had no idea that home 3D was real, but I was upgrading an old Sony, a big Sony glass bottle CRT, I'm trying to remember the name of it, it was XBR, a Sony XBR receiver that I had. And so I just - I went to Amazon and poked around, and there was an LG, it was about the size I wanted, 47". I said, oh, okay, that'll fit. And it said 3D. And it's like, okay. I just didn't believe that it was anything but a gimmick. But I also needed to update my Blu-ray player, so I got one that was 3D capable.

And sure enough, I mean, it uses the spiral polarization so that one eye is counterclockwise polarized, and the other is clockwise polarized. We talked about this some time ago. This is the real 3D technology. And so they're passive glasses, no active shutters, no batteries or anything. And, I mean, it's pretty good. It was like - it didn't cost anything, so I didn't pay much extra for it. But I was just surprised that it was real. And I was tickled by that.

I'm in the process of updating my creaky old home media approach. About a decade ago I set up a trio of Series I TiVos, and I have updated them. I've updated the Linux kernel in them. I've given them larger drives. I've sort of kept them going and alive. But they're

standard definition. And so I figured it was time to move. And I know that you and Paul have talked a lot about the Media Center, the Windows Media Center. And after poking around and looking at Sage, which was the one that Mark Thompson was loving, except that Google bought them, so that's sort of no longer an option, I decided to build myself a Windows Media Center box. So there's a...

Leo: Oh. You're really - welcome to the 21st Century, Steve. And some day you'll be in this decade.

Steve: Yeah, there's some things where I don't want to be on the bleeding edge. Because apparently it had a rough history. It used to be an app that you could get for XP. And I guess there was, what, like a Media Center edition for a while.

Leo: Yeah. They're offering it on Windows 8.

Steve: Yeah. And it's just there in some of the versions of Windows 7, which is what I'm running. But it's very smooth and slick. There's a card by a company called Ceton, which is - it's called the InfiniTV, and it's a PCIE card, a 1x PCIE card that you plug a cable card into that you get from your cable company, and it'll decode four streams. So I can record four different things at once. And it is a DLNA server. And it turns out that all the devices I've recently purchased, this LG flat panel is itself a DLNA client, as is the Blu-ray player. So then I get the whole watch-stuff-in-the-other-room sort of capability. So as you said, yes, I know, welcome to the 21st Century. But, okay.

Leo: I'm glad you're happy, Steve.

Steve: I'm having a good time. And speaking of having a good time, Anthony Pitcher wrote, on the 1st of September, on Saturday, with a little note. He said, "SpinRite saves me again." He said, "Hi, Steve. I just felt compelled to tell you that SpinRite saved me again. I was having frequent total computer freezes, i.e., the mouse would not move, nor the NUM LOCK lights change, and a full reset and hard reboot was the only resolution. After running CPU Burn-in tests, Memtest86 for over 24 hours, tests and drive SMART checks all reported okay. The freezing still occurred every day.

"I decided to run SpinRite as I know it's the only sure way of knowing the disks are okay. SpinRite did not report anything was fixed." How many times have we heard that? He said, "But I have not had a single freeze in over a week. Fingers crossed, but it looks cured. If SpinRite did fix the program, I didn't know a faulty hard drive would cause Windows to completely lock up. Is this common? Thank you again."

And actually, for example, one of the things that the drive is doing, might be doing a lot of, is swapping. And it's typically code which is swapped out and swapped back. So your memory could be fine. But if the drive channel was flaky, I mean, and not all errors are caught, which is a problem because these are just pretty simple checksums that are being done. It's very possible for errors to get by if they're recurring a lot. And so you might have, between the time the code went out and came back in, you could have some code altered. And when the CPU then attempts to execute that code, which it would do because that's why it was asking for it to be swapped back in, it could easily just lock up completely. So, yeah, this is another instance where SpinRite fixes things silently

because it's working with the drive to clean up sectors and swap them out and make sure that they're being read correctly. So thanks, Anthony, for the report.

Leo: I'm glad it worked.

Steve: Yay. Me, too.

Leo: Yeah, really. Of course it worked. All right. I have questions. Steve has answers. If you are ready, sir, I shall begin with our first question, from Berlin. Michael Walther - I wonder if his family invented the gun - paying close attention, caught a verbal typo, or a "verbo." Dear Steve, just a - and his isn't even his native language, and he caught it. Just a short aside: In Security Now! 365, you mentioned the Logo programming language with its Turtle Graphics. You then said Logo was a product of the mid-'60s. Well, not quite so. In the mid-'60s we used to have punch cards and hand-wired magnetic core memory, but not much more. I seem to remember it was shortly after I bought my Apple II, back in 1980, that the Logo language appeared. Love the show. Cheers, Michael.

Steve: And of course he's right. In the early '70s, in between, was when I was programming my PDP-8s, working in the afternoons after high school. And so if I said '60s, I wanted to thank Michael and correct that. Certainly it's 1980. That's when...

Leo: Nope, it's 1967, Steve. You were right.

Steve: What?

Leo: Seymour Papert created it in...

Steve: Oh, that's right, he was on the ground back East. Yeah, yeah, yeah, at MIT, was it?

Leo: MIT. And it's based on LISP. So, no, it's vintage. It is vintage.

Steve: That is right.

Leo: It was true that the first - a lot of people got exposed to it in school and so forth with Apple IIs.

Steve: Right, right.

Leo: But, no, no, Logo's been around a long, long time.

Steve: I did remember vaguely.

Leo: And I don't think it was originally designed to be teaching kids. In fact, I'm looking at the article on Wikipedia. The first implementation of Logo was called Ghost, written in LISP on an SDS 940 at Bolt, Beranek and Newman, BBN.

Steve: Yep, BBN. And I was going to say, you don't want to give kids - you don't want to let them even see the LISP programming language.

Leo: Now, well, actually, "The goal was to create a math land where kids could play with words and sentences." So, and they did use turtles right from the beginning. But the first turtle was a physical, as you say, on the ground, a physical turtle.

Steve: Right.

Leo: That moved around. So Michael, it may have been your first experience, but in fact they are that old.

Steve: I was right from what I remembered. Cool.

Leo: Yup. Question No. 2, soon as I can find it. I keep closing these. Here we go. From Max Cohen. He's considering the plight of Mat Honan, our friend who's the writer for Wired.com and lost all his data, and obtaining change: I understand how the people who "wreaked havoc" for Mat might face criminal charges, but how would one have gone about telling Amazon and/or Apple about a security flaw unless someone with access to a large website, in this case Wired, published it? If a person found a security issue, they could contact the company. But let's be realistic. Would that company even do anything without pressure similar to that which Wired created? It would seem the one reporting the problem could face prosecution for having that knowledge. What are hackers supposed to do? Where's the safe haven or whistleblowing protection for reporting issues?

Steve: And this perfectly leverages against what we were just talking about, the Oracle problems that had been reported quite some time ago, but because they weren't critical...

Leo: It's tricky.

Steve: I mean, it is, exactly, it is tricky. I mean, you can understand, on the side of the developer, their not wanting to respond to every flaw, unfortunately, because there are so many of them. So they would like to aggregate them. And on some level they must be holding their breath that, like, ooh, we've been privately informed...

Leo: We know it's there.

Steve: ...of, like, really bad - yeah, it's like, oh, crap.

Leo: Hope nobody finds it.

Steve: Yeah. We've been privately informed of some really bad stuff, but we would really like to stick to our release schedule. And we hope we make it to October 16th. I mean, maybe that explains why they were able to do it in a day, is it was ready to go. But they thought, well, we don't want to confuse people, we don't want to push it out unless we have to. And but I think Max Cohen, who asked this question, raises a very good point because we have many situations where people report, I mean, we're talking about all the time where these disasters befall companies, they knew about the problem, but famously were not fixing it. And unfortunately it only is when it really becomes a problem or when enough press gets behind it that, I mean, essentially that resources get allocated in order to handle it. Until then, everybody's busy doing other things. So it's probably just competition for resources. And unfortunately, not everybody is Mat who writes for Wired, and it may even be difficult for someone who found a problem to be able to generate the press required to make something happen.

Leo: Well, that, and there's some risk, too, perhaps, that you could be arrested. I mean, I could think of people like Adrian Lamo who...

Steve: Yes, yes, we see that all the time, too.

Leo: Yeah, I mean, he says I was pentesting, I was just showing The New York Times - because he broke into The New York Times and started changing data there. Not maliciously, he says, but just to show them their flaw. But he was prosecuted.

Steve: Well, and the Digital Millennium Copyright Act, the infamous DMCA, really is overbroad in the protection that it provides to copyright holders, allowing them, for example, I mean, there are university professors who are now afraid to publish their discoveries about crypto that we're using because it's DMCA protected. And so that's hurting academia. It's hurting the development of the crypto industry and the security for all of us.

Leo: Let's move on. Double-click. Thank you. It's funny, it's open, but it's not showing that it's open. I don't know why. There's something weird going on with my computer. Ethan Stone in Berkeley, California suggests a prevention of Honan's Bad Weekend: Hi, Steve. So here's my current thought about how to minimize the problem. Oh, boy. I set up 10 to 20 free "feeder" Gmail accounts with strong passwords, different for each account, and two-factor authentication. I set them all up to forward to an additional "aggregator" account. I'm not sure that it should be Gmail. Further discussion below. I also set that one up with a separate strong password and, I think, two-factor authentication. I then assign feeder emails at

random to everyone who needs an email address. I never post the aggregator email address anywhere and never give it as a direct email address to anyone for any reason. I then stay logged into the aggregator, possibly running Chrome for that purpose only, since I usually use Firefox.

I know this isn't a complete protection, but it will stymie lots of Honan-style attacks, since any one feeder account is likely to be different from the one used on the account that the attacker is trying to get into, effectively putting a firewall between those two accounts.

I'd be curious what you think. I'd also be curious what your thoughts are about using Gmail account for the aggregator versus an account under a custom domain through my hosted Exchange provider. As I see it, Gmail gives me two-factor authentication and some additional separation from my other accounts. My "real" email accounts are all hosted Exchange accounts on my own domains. Then again, it makes Google's security procedures a single point of failure. Ethan Stone.

Steve: So this sort of represents - I chose this as a proxy for many people who proposed similar ideas. I think our Matthew Stone - Stone. Our Mat Hogan...

Leo: Honan.

Steve: Honan. Right. It gave a lot of people pause because they realized, oh, that's what I'm doing. That could happen to me. And clearly, what we saw in the scenario that we covered with Mat was the problem of guessability of his email accounts and the multiuse of an email account. And we discussed this a few more times during the remainder of these Q&A. So I'll go into this as we encounter them because we wrap up with perhaps a superior approach that I like a lot for what it offers. But essentially I wanted to sort of highlight this as the thinking that many people have done and the fact that it certainly begins to solve the problem.

I agree with Ethan that I don't think having an aggregator that's in Gmail is a problem. Google offers strong authentication. I talked a couple weeks ago about wanting to assemble sort of a formal statement. And the news overrode my ability to do that. So I'll just take this moment to say that it seems to me the source of most of this is the whole issue of password recovery. And, for example, as we have mentioned, Leo, it is crazy for me to use a multifactor authentication device to log into PayPal, and underneath the field where I am to provide the code from my hardware football there's a link that says "I don't have it with me." Okay. Wait a minute. What is wrong with this picture? Because the hacker doesn't have my football with him, either.

Leo: Right.

Steve: And so what I think we need to - the next step in this, what we need from websites, and this is not a hard problem to solve, is a checkbox which can be enabled by default, that's okay. And what that says is, when we're creating our account, we're in our security settings, it is "Enable password recovery." And it can be on. Fine. We need to be able to turn it off. We need to be able to say, we're adults. We care more about security than we do ease of being hacked. I mean, the reason I want this device is I'm saying I

will take responsibility for arranging to have it with me one way or the other. And now that we have OATH for smartphone-based Google Authentication style, it's easy to have it with us. And if you can grab, as I talked about last time, the key, and put it in your various iOS devices so that you're not stuck with it only in one, then we've created much stronger authentication. But then, if that's the case, don't ask me the name of my first girlfriend or of my dog or my favorite teacher in high school. Turn that stuff off.

Leo: That's just more stuff you know. That's not really another factor.

Steve: Oh, exactly. And obviously, remember that in the story that Mat related...

Leo: Not stuff just you know.

Steve: ...the hacker was asked those questions.

Leo: I don't know. And he said, "I don't know. I don't know. Here, but I have the last four digits of my credit card and my address. Isn't that enough?" Oh, yeah, sure, that's enough.

Steve: So what we need to move forward is now we have multifactor. Now we have to turn off the bypass. Turn off - give us the option. Have it on by default. That's fine. Most people, that's probably okay. But every time I'm logging into PayPal, and I'm being asked the six-digit code, and I'm holding my football right here, and right underneath that I see the link, "I don't have it with me," it's like, okay, why am I holding my football? I'll just click that link. It's crazy.

So to move forward, this is what we have to ask for. We have to say, give me the option to take responsibility. And whatever it is. Or maybe have something way more onerous than something as convenient as what's my mother's maiden name, something that's not like that. Or absolutely no excuses. No-excuse login, whatever. That's the only way we're going to move forward is for us to be able to take responsibility for not forgetting our credentials. Because otherwise, almost by default, I mean, almost by definition you can say, unless we're given responsibility, we don't have responsibility. And if we don't, then we don't have authentication because anybody can impersonate us.

Leo: Somehow I just doubt it's going to happen because the companies will say, but, yeah, but then we're going to get calls from customers who've lost their dongle and have lost access as a result to their accounts.

Steve: And that would be inconvenient, yeah.

Leo: Yeah, yeah. Well, it could be more than inconvenient for that company. They could get...

Steve: Oh, and look at No. 4. What do you know? The next question.

Leo: The next, which takes us to Tim in Sydney, Australia, who proposes perhaps we don't need immediate gratification on password resets: Hi, Steve and Leo. I say DITTO! At the end of the last show you pretty much concluded that the weakest point in our online security system these days is the password recovery system. I completely agree. I do wonder, though, why we need instant gratification when we lose a password. Why do we need to be able to access our account straight away? Why not simply deactivate the account for a couple of days after an "I forgot my password" reset? After all, it is my fault, and I should be made to suffer a little in the interests of security, especially for these free services. If this were enabled in Mat's case, he would have quickly realized something was up and contacted the service provider before any damage could be done.

More generally, I figure that there is a business opportunity for a trusted organization like the post office to offer an in-person identification service. If, for example, you were to lose an important password or otherwise be hacked, you could call Google tech support and, after supplying the "usual" weak identifying information, put a "hold" on your account. Then Google could get you to print out a prefilled PDF form with reference numbers and bar codes and everything. You take that to the post office, where they could verify that you have what we here in Australia call "100 points of ID." Through the post office computer system, Google would learn that you are the real deal, then release the account.

Sure, this would not be a free solution, with the post office charging a fee for each verification. But the onus is on me to maintain password security; and, if I fail at that, I would be very happy to pay for my mistake. Similarly, if I were subject to a hack attack that would otherwise succeed, I would be happy to pay, as well. If such a feature were a option in Gmail security settings, I would enable it in an instant. Tim.

Steve: It's interesting to invoke the post office because this is not the first time the post office has come up in the context of security. You may remember, Leo, that there was some discussion for a while about the idea of some sort of personal IDs, personal certificates. And the question was, well, how would that be managed? How would people prove their identity, and who would issue them and so forth? And the answer that was proposed was the post office. At least in the U.S., and apparently in Australia, they're ubiquitous. They're staffed with people. If you have a package that you need to pick up that's registered, you have to provide some proof of identity in order to pick it up. So it's sort of - that's an interesting idea.

And also the notion that, well, maybe we don't need an instant bypass of our identity. The notion of sending out information to the registered email address and putting a hold for some length of time on the account would allow - would certainly shut down the hackers' immediate need to do all these things. One of the things that we saw in the Mat Honan story was that, remember, there was a timeline where it was like, at 5:45 this happened, at 5:50 this happened, at 5:55 that happened, at 6:00 o'clock that happened. And so it was like, bang bang bang bang bang, where they were in lockstep, sort of spreading through his own electronic identity, all because they were able to do this very quickly. And so the notion of some sort of a refractory period after a password recovery is an interesting one.

Leo: Hmm. Eh.

Steve: I know, I know.

Leo: You guys are living in a dream world. Dream on. Ed Reilly in Springfield, Massachusetts wonders how web-based WiFi authentication works: Steve and Leo, love the show. Using SpinRite since 1993 - wow - when we would use it to change the interleave - he says "interweave," but it's interleave - on the old drives to speed up them up.

Steve: Yup.

Leo: I remember that. Did SpinRite do that? You could change the interleave from 1:1 or 4 to - what, it was originally at 4:1 to 1:1? Make it faster?

Steve: It's an amazing - it's amazing. IBM released the XT, which was the PC with a hard drive in it. And the interleave was wrong. They didn't know what it was supposed to be. It was 6:1.

Leo: 6:1.

Steve: 6:1, which meant that every - that successive logical sectors were spaced six sectors ahead, so that the...

Leo: The idea was that you wouldn't have to wait for the drive to spin all the way around to...

Steve: Well, yeah. The problem was the bus at the time was so slow that, if you read a sector, then you asked for the next one, that was already - and if the next one was physically adjacent, it was already too late to get it. So you'd have to go a whole revolution around to get sector 2. Then you got that, and you'd ask for No. 3, and you'd have to go all the way around again. So they would be spaced out and interleaved among each other to give the software time to ask for the next sector.

Well, then what happened was Western Digital came along with the famous 1003 controller that was in all the clone, all the PC clone machines. And they thought, well, we're going to have our controller be twice as fast. We'll set the interleave to three. And so what happened was, with an interleave of six, naturally it would take six revs to read all 17 MFM formatted sectors. Western Digital could do it in three revs because you would get every third sector each time around. The problem was the clones were not fast enough to do an interleave of three. They needed four. And so the other problem is that all this interleaving is in the low-level format. It's the actual sector headers that contain the interleaving. So you can't fix this in software.

So SpinRite started out to be a nondestructive re-low-level formatter. It would low-level format the drive and determine, by building a table of performance versus interleave, what the optimal interleave setting was for your particular computer in your environment. And then it would fix it, and it would re-low level format the drive, tuning the interleave to be optimal, and people would get a 400, in the case of Western Digital,

all the clone computers, 425 percent performance improvement after running SpinRite just once on the drive. So, and it turns out that the IBM XT, it could also run an interleave of four, so it would speed it up 50 percent, from six revs to four revs, after just running SpinRite once. So those were the days, 1993, yup. What were we talking about?

Leo: Oh, I'm sorry, Steve. I'm online with my bank. I have a fraudulent charge. I'll get back to you in just a second. [On telephone] Okay. Thank you. I don't, I don't. Thanks so much. All right. Bye bye. Apparently just a few minutes ago somebody charged $5 in an Arkansas Hyatt on a card that I have right here. and then, like, 50 cents on something in England. And I asked them, I said, I definitely didn't do those. And I said, what? He said, well, sometimes they'll do these small charges to test...

Steve: Those are test charges, yup.

Leo: Arkansas and England. And they immediately flagged it because I don't normally buy things in England and Arkansas within a few minutes of itself. And I do have the card right here. It's my business card. So they caught it. So now I'm without a card for three days. But better that than more charges.

Steve: Yup. That happens. I think I've mentioned, when I'm traveling home for the holidays, which I do, well, the holidays, annually, my travel agent will say, okay, Steve, do we still have the same card as last year? Because they tend to be a little slippery. It's like, ah, yeah.

Leo: That's right, yeah. No, and by the way, I don't answer the phone normally. In fact, it normally never rings, I was kind of surprised, during the show. But it said "Bank of America Fraud." And I thought, oh, I'm on Security Now!, I should probably answer that call.

Steve: And I'm busy talking anyway, so you'll probably just be able to slip that one in.

Leo: Snuck it in there. Question No. 6, James Gilbert in the woods near Micanopy, Florida questions backup: Good show. I've been listening for about a year. I remember watching Leo back in the ZDTV days. You, too, Steve. You were on with us.

Steve: Yeah.

Leo: I got to thinking about the whole subject of backing up your computers. Yes, it's terrible to lose important pictures, or financial records, or that great novel, or in my case my great music composition that I've been working on. But if you're old enough, think about it. When we got our 35mm film developed, we rarely had more than one print made. No backup. Many people tended to store negatives and prints together - no offsite backup. I still get a paper statement from my financial institution. Nobody makes a backup of that. In spite of not making local or offsite

backups of these important things in the pre-personal computer age, we managed to keep the important photos and important papers. I've got a file cabinet full of music arrangements from the 1980s that were performed once then and, given the way the music business is going, the world may never hear again. But I've still got them.

This suggests a couple of possibilities to me. One, we've become a bit paranoid about backups; two, we don't treat our data with the same respect and concern we had with those old photos; or, three, computers have a long way to go before they can be considered as reliable as paper. I lean toward No. 1, but I suspect all three are true to some degree or another. Any thoughts? James.

Steve: We've sort of talked about this in different contexts from time to time. And it interests me because I'm on No. 3, computers have a long way to go before they can be considered as reliable as paper. I'll bet that, if I were to somehow do a survey of the reason most people purchased SpinRite initially, it would be because photos were lost.

Leo: Right.

Steve: I think photos uniquely are, I mean, that's certainly not the only thing that people have to recover. But music most people don't create themselves. They got them from CDs, or they downloaded them from online or something. Movies they're not creating themselves typically, again. Those came from somewhere else. But photos are uniquely media they created themselves. And as we moved to digital cameras and away from the 35mm film, they've been in digital format. And I've just - I was just seeing another SpinRite story this morning when I was going through the mailbag about SpinRite recovering absolutely precious photos.

And so what's different, I think, about this digital domain, as James notes, or asks us, is that this stuff still, I mean, computers are failure prone. That's my business is helping people recover from the fact that hard drives are always being pushed to store probably more data than they really should to offer the reliability that people would like. I don't think I've mentioned, Leo, I don't know if you're aware, you may have seen the news go by that the warranty periods of hard drives have been snuck down also?

Leo: They're, like, three months now; right?

Steve: They used to be multiple years. And they've said we're going to kind of shorten those warranty periods. It's like, ooh.

Leo: I think part of that is people are buying OEM drives now. They want the cheap, cheap, cheap drives. Remember OEM warranties were always 90 days, and then you could buy the inbox shelf version, and they were...

Steve: Yes, I think they were like, like Western Digital used to be three years. And I think it's brought it down now.

Leo: Yeah. Five years on the Western Digital RAID edition. Five years. Nobody's going to warranty a drive for five years. That's crazy.

Steve: So a shoebox of photos, unless your house burns down, but they don't spontaneously cease to exist. It really is the case, though, that some percentage of data stored on computer media, it's not only hard drives, although they're physical and the most temperamental. People have problems with thumb drives where it just - it breaks. Maybe a little static zap goes out and touches one of the pins. And suddenly it's all gone. And without backup or data recovery, you're in trouble. So it's a different problem. But I think people really do need to maturely understand that, if it's really precious, if it's irreplaceable, make copies because the computer may not give it back to you.

Leo: Somebody in the chatroom, Web1038, said - and the other thing, the other side is there's zero cost in space and time to making a backup of digital, or very little cost. So it's one thing to say I'm going to make a backup of this shoebox of photos. That's nontrivial.

Steve: Or to Xerox a whole pile of papers.

Leo: Right, right.

Steve: And, I mean, people do lose photos in fires all the time. And it's a tragedy. But it's just so much difficulty to back it up. Whereas backing up your hard drive, if you just do it, is not a lot of difficulty.

Leo: So I think Web1038 in the chatroom also has a point there, that it is easier. That's one of the advantages of the digital. You know, when we - I had - we had one set of slides in the family. They were our family slides, and they were in my mom's attic. And I felt like that was risky; right?

Steve: Yeah.

Leo: So my sister brought them in, got them scanned, and each member of the family got a CD. Now I feel much better about those pictures, and I uploaded them to SmugMug. So those pictures are preserved. There isn't just one copy. And that made me nervous. So in a way I think this is a new capability that we didn't have before that makes it better to save stuff. Not that anybody's going to want to see those pictures in a few years, but - you know, you wonder. You save all this stuff. Who's going to look at it? My grandson? I don't know. Maybe. I think if I had pictures - wouldn't it be interesting if you had pictures of your great-great-great-grandfather?

Steve: Yes, yes.

Leo: That would be of some interest.

Steve: Yeah, and there really is a notion of the world before the Internet not being online.

Leo: It's not recorded.

Steve: Everything since is, like, going to be archived forever. And everything before, it's like, oh, it's kind of creaky and dusty and just never going to be online.

Leo: Yeah. I wish we had archival footage of Abe Lincoln and George Washington. Those things would be really, really intensely valuable now. So it is a different era.

Steve: Yeah.

Leo: Mark in Melbourne, Australia wonders about secure usernames: Steve and Leo, we know what makes a password more or less secure - length, size of character set, randomness, unique to website/service, et cetera. But I don't think yet I've heard you talk about what makes for a more secure username. How do you choose your usernames, Steve? And when the website/service uses email addresses, should you use your main email address? This is actually germane because one of the things that made it easy for the bad guys to get Mat Honan's stuff is he always used mathonan@service.com.

Steve: Yes, gmail.com and @me.com.

Leo: So they could guess his email address. And that would turn out to be crucial in the success of the hack.

Steve: Yes. And so this is a really good point that we'll leverage into our closing question also. And this is a problem, is that email addresses are our usernames. The common practice now is you create an account with your email address and your username, I mean, sorry, your email address as your username, and then a password. And from the account service side, from the web service side, there's an advantage to that. You don't need a username separate from email address. The username is their way of contacting you for verifying it, for unfortunately bugging you with free offers and random garbage that you may or may not want. But also importantly, think about it, they don't have to worry about username collisions because two different people don't have the same email account.

Leo: Which is great.

Steve: Yes. And so it's like, oh, sorry, this username is taken. Well, we have that with

Twitter because there we really want a separate username, our Twitter handle, separate from our email address. But that creates a chunk of overhead. And if you use an email account as your username, you solve the uniqueness problem of usernames because email accounts are sort of inherently unique. So, but we'll talk a little bit more in a second about what makes a good one. Clearly, though, as you said, Leo, you don't want it to be...

Leo: First name, last name, yeah.

Steve: ...the same on all of your accounts, yeah.

Leo: I'm going to have to really look at that because I use that everywhere. I shouldn't have said that. Alejandro - but nobody knows my middle name. Alejandro in Chicago has encountered "non-routable 5.x.x.x" addresses: Steve, this is my very first time contacting someone over the 'Net. Wow. Welcome, Alejandro. But I thought it would be a great opportunity to get a little more information about IP addresses in the 5-dot range. I remember these from Hamachi.

Steve: Exactly.

Leo: I'm currently seeding Ubuntu and other open source distributions and noticed that on some of the peers I'm connected to, the IP address begins with 5. As I recall, you once mentioned that this range is non-routable and used only in virtual mode by Hamachi, now LogMeIn. If I can see these IP ranges, does that mean they are public or these hosts are simply disguising their real IP? Every one of them, by the way, in Russia. If it helps, I was able to ping them, too. Thanks. I'm hopeful you'll choose my question for one of your episodes. Wow, that is a mystery. This is worthy of Sherlock Holmes.

Steve: Well, and this is what we would predict with the depletion of the IPv4 address space: 5 has been allocated.

Leo: Oh. It's no longer unallocated.

Steve: No, because remember we're running out. And 5 is - the whole 5.x.x.x or *.*.*, that's 16 million IPs very - or, wait, 16? 2^24.

Leo: 16 million.

Steve: Yeah, 16 million. So there's 16 million IPs that for the longest time was just sitting around. There was no 5 anything. And that's why Hamachi was able to very cleverly at the time use it because there was no danger of it colliding with anything public. Well, now...

Leo: But it wasn't guaranteed unused. It was just unused.

Steve: Absolutely not.

Leo: It wasn't like 198.162 or...

Steve: Right, those have been set aside on purpose. Like 10.*.*.*, that is still - there's 16 million IPs nobody ever gets to use because that's been deliberately set aside for local networks. But 5 just had - no one had gotten around to allocating it. From what Alejandro has said, that's no longer the case, which I thought was really interesting. I haven't verified, haven't seen who got them. But if he's seeing peers connecting - I'm assuming he's talking about, like, torrent peers. If he's seeing torrent peers connecting, and he can ping, then, I mean, there's no way those can be disguised. They have to be real. So that means 5 is now public, which I thought was an interesting development.

Leo: Yeah, it's kind of depressing. The end of Hamachi. Somebody asked on the radio show this week, he said he wanted to make - he had two houses at different locations, wanted to have them be on a LAN. And I said, well, Hamachi, but I don't know what the status of that is now that LogMeIn...

Steve: I don't know now either. They had to have changed something because they can't be using 5.

Leo: Can't be using 5-dot. Alan M. in Manchester, U.K. suggests a strategy for safely emailing passwords: If your military/DoD guy can run portable versions of programs-

Steve: And you'll remember that he's referring to a Q&A we covered where somebody was - he was in a sensitive environment where he couldn't bring any devices in, but he wanted to know how to transport his password lists from place to place.

Leo: Right, so he was able to use a portable USB key - as you would need to do for AxCrypt, then Keepass (the open source LastPass, basically) might be a good idea for him. There is a PortableApps.com version of the software. It fully encrypts the password database, which is easily emailable, as would be the program itself. It's less than 5MB uncompressed. It may not be LastPass, but it's quicker to use than Excel, which is what the guy was using. Enter your passwords, then double-click the username or a bunch of asterisks to copy to clipboard. Then it clears it after 12 seconds, in case you forget to clear the clipboard. Hey, that's a good idea.

Steve: Isn't that nice, yeah.

Leo: I thought this was slightly easier to use than AxCrypt due to the auto-clearing clipboard and that someone over your shoulder never actually sees the passwords

because they're just stars unless you wish to open the details about that specific item, then click it to uncover it. It also will remember URLs, notes and all that good stuff, if you want it to. It's kind of like LastPass, but it's open source and free: Keepass, K-e-e-p-a-s-s.

Steve: Right, and its little database you are able to carry around with you. So I wanted to share that with our listeners.

Leo: Yeah. Keepass is great.

Steve: And JD's final question.

Leo: Our last question of the day, JD in Memphis. He has a solution for completely unique, per site and service, email accounts: Great show, been listening for a while, LastPass, SpinRite, yada yada yada. He's got a post on his blog, jd7.org: In it I talk about how to make a different email address for every online service but manage it all with only one email account. Basically I quickly talk about buying a domain, using Google Apps free with that domain, and creating a mailbox that's a catchall. You keep the primary email account secret, don't use that anywhere, and choose a secure eMail address. Security Now! listeners would know what such a thing should look like, basically random characters and letters and numbers.

Steve: Exactly.

Leo: But then, when you sign up for Amazon or Pinterest or anything else, use something like pinterest@domain.com that will go to your primary account because you own the domain. You use LastPass to keep up with all the usernames - per-site email addresses - and passwords. My question is would this be strong enough to prevent the @mat problem? Would love to hear how to strengthen this. Currently I am using this, and that's why I suggest it in my blog post. I have a short, three-character Twitter name - that's what they went after Mat for - and do not want to become a target, as well. But I don't use my me.com email for anything, either. Thanks for all you and Leo do. JD, Memphis, Tennessee.

Steve: So, okay. This is the remaining link that I shortcutted in my Twitter.com posting, so anyone who wants to read what JD wrote...

Leo: Seems like a great idea.

Steve: He has, essentially, a step-by-step how-to.

Leo: And it's free.

Steve: Which is very nice, yes. Well, almost. What he suggests is you start by getting your own domain. So Hover or...

Leo: That's not free. That's 10 bucks or something, yeah.

Steve: Yeah, 10 bucks a year. But you have the coolness factor of your own domain.

Leo: You should do that. And you control it that way. You don't, if you're changing your email server, you don't have to tell everybody, you just control it. That's better.

Steve: Exactly. It'll never change. And he does mention making sure that you use a service that allows you to mask your WHOIS data. We've talked about that.

Leo: Hover does that for free.

Steve: Yes. So Hover looks like it would be a great choice. So get a domain name. You'll have your own domain name, which is a cool factor anyway. And using Google Apps you can have, essentially, wildcard email, which is - this is the coolest thing about what JD has suggested is he calls it a "catchall mailbox." But the idea is that anything coming to that domain, so it's anything@domain.com, whatever your domain is, will go into there. So what's cool is this gives you instantly per-site, per-service email accounts.

Now, I would not - one improvement I would make, I would not use pinterest@domain.com because then we're back to guessability. Someone might say, oh, well, then, he's probably using amazon@domain.com and twitter@domain.com and so forth. So you want unguessable names. But you can make them up on the fly. You can, as he said, use LastPass as your userID, which would be your - we were just, before this, we were talking about how our email addresses have become our userIDs, or our usernames. So you really want them to be unguessable. You want them to be something, someone who sees one can't guess what the other one is.

And I have to say also that common names are spam bait. For the longest time I was just steve@grc.com. And I couldn't figure out how my name got out. Well, it didn't. They were just - I once looked at the traffic on my own email server, and servers were hooking up and just running through name lists. They were just trying to send email to every first name at GRC.com that they could. And I finally got a clue and made my email address something else from that. So you don't want to be "Amazon" or "twitter" or something common. You want to do something to it that will not have it just automatically and easily guessable.

But anyway, so I wanted to point people to this because it is a cool solution. It will cost you, I think it's 10 bucks a year grand total because Google Apps is free, and you're only having to pay for the domain, but it's neat to have your own domain, too. And then you get this wildcard concept with email. And if you're willing to, I mean, it's not as easy as using the same email address for everything, but our email addresses are becoming our userID. And it would really be nice if those were unique, too, in addition to our passwords being unique.

Leo: So much to do to secure ourselves. At least the information is out there, and it's all here at Security Now! thanks to Steve Gibson. Steve, you're the best. If you want to...

Steve: Well, I'm all we've got.

Leo: You're all I've got. If you want to ask a question of Steve, he does these feedback episodes every other episode in general, and you can go to GRC.com/feedback. Don't email him, just GRC.com/feedback. And then you can handle that without any trouble, just fill that form out. You can also find, once you're there, SpinRite, the world's best hard drive maintenance and recovery utility, and all of Steve's freebies. You've got a ton of them, ton of stuff.

Steve: We should remind our listeners of a scheduling note.

Leo: Oh. Yes. We won't be here next Wednesday. We will be here next Tuesday. We'll flip-flop with MacBreak Weekly because of the Apple announcement on the 12th. So if you're planning to listen to Security Now! live, it will be September 11th at 11:00 a.m. Pacific, 2:00 p.m. Eastern time.

Steve: And if you're not live, it'll probably be available a day early.

Leo: Yeah, we'll just put it out a little bit earlier. Yeah, we make on-demand versions available after the fact at TWiT.tv/sn. Steve has special online, on-demand versions, both 16Kb audio for very small file sizes and the transcripts, even smaller, which are done by humans so they're very good, done by Elaine, and...

Steve: One good human.

Leo: One good human at GRC.com. You'll find all of that. I think that's it. Yeah, see you next Tuesday, not next Wednesday.

Steve: Yup.

Leo: We'll do the Apple iPhone event at 10:00 a.m. on Wednesday, next Wednesday. We're going to move TNT up an hour, and then we'll do the iPhone event, then we'll do MacBreak Weekly. It's going to be a crazy, jam-packed day next Wednesday.

Steve: Cool.

Leo: Thank you, Steve.

Steve: My pleasure, Leo, always fun.

Leo: See you next time on Security Now!.