Listener Feedback #154

Description: Steve and Tom 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-377.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-377-lq.mp3

SHOW TEASE: It's time for Security Now!. Hey, I'm Tom Merritt, filling in for Leo, who's off on a cruise. And I get to hang out with Steve Gibson and talk all about security. We're going to go over the new browser updates, both in Firefox and Chrome. There's some Do Not Track stuff to keep an eye on. But there's some cool stuff in that Chrome revision, as well. And we're still going to be talking about OAuth. Just because Leo's gone doesn't mean there's not more to discuss. We've got some great questions from folks in our Q&A section, coming up next.

TOM MERRITT: This is Security Now! with Steve Gibson, Episode 377, recorded November 7th, 2012: Your questions, Steve's answers, #154.

It's time for Security Now! with the man who tries to keep you as safe as you possibly can online, Mr. GRC.com himself, Steve Gibson. How's it going, Steve?

Steve Gibson: It's going great. And everyone is probably realizing you're not Leo.

TOM: I am obviously not Leo. Leo's off on his cruise. So I get the pleasure, I get the perk of filling in on Security Now! for three weeks. And this is going to be fun, man. I can't wait.

Steve: Well, you have a familiar voice to everyone who listens to you and your gang over on Tech News Tonight. Or I guess it's Today, isn't it, Tech News Today?

TOM: Yeah, I guess it's sort of Tech News This Morning because we do it in the morning now. But yes, I'm Tom Merritt, host of Tech News Today. And this is Security Now!. All right. Well, let's get into the security news, starting with the updates. We've got some Adobe updates to talk about.

Steve: Finally, Adobe is on schedule. We've had some famous emergency updates which

they've reluctantly released outside of their quarterly normal update schedule. But this is back on their planned schedule for actually yesterday, which was also Election Day in the U.S. So this is November 6th was they made available updates to Flash and Air, both of those runtimes. The updates pertain to Windows, Mac, Linux, and Android releases, so across the board. They fixed seven troublesome critical vulnerabilities, so these are worth doing.

I went to check my version under Firefox, wondering if Firefox would be alerting me to the fact that this version of Flash was out of date because there's been some plans to do that. And I know that in some cases it takes responsibility. In this case it didn't. Maybe I need to shut it down and start it up again or something. But you can go to www.adobe.com/software/flash/about. And that will run a little Flash app which shows you a bouncing red cube, given that you permit Flash to run in your browser, and it will show you your current version and also list the latest versions that are available. And in the case that I did this today, I was seeing that I was not running what they have just released. Now, on that page they do give you a "Download the latest" button. If you go there, you want to be careful to turn off the "Get your free McAfee virus scan."

TOM: Oh, man. Did they opt you in? Do you have to uncheck it?

Steve: Oh, yeah. That's so annoying. You can also go to www.adobe.com/products/flashplayer/distribution3 - that is, the numeral 3 - dot html. That just takes you directly to a clean page where you just get the download that you need for whatever platform you're running and no upsell nonsense installing things you didn't ask for.

TOM: Now, Chrome has its own Adobe Flash built into it. But a lot of Chrome users still need to do this; right?

Steve: I'm not sure where Chrome is. I haven't seen, I didn't see an announcement from Chrome in the last day or two. But they're now taking responsibility for keeping that up to date. So they ought to be doing it for you.

TOM: Yeah. So you might want to check on that. I know there are some features in Flash that I actually have to have the Adobe Flash plugin running to get them to work in Chrome because the Chrome Flash is incompatible for some reason. Every once in a while there's a page that'll do an auto-detect. So that's just a little extra thing to keep in mind.

Steve: And it is annoying that we cannot get away from it. I mean, I know that when I, I mean, I really love my iPad, and I use it all the time when I'm out and about. And you run across sites where it's like, okay, wait, I'll just have to go, you know, wait till I get home in order to do this because it's still - it provides enough extra, if you'll pardon the term, "flash" to the site that there are a lot of sites that still just assume that you're going to have it or be able to make it available.

TOM: Wordsworth in the chatroom points us to the Chrome blogspot, claims that their Chrome 23 stable release did include a Flash update.

Steve: Ah, good. And in fact we're about to talk about Chrome 23 stable, which is just out. But I wanted to mention Firefox first. We've got a new Firefox beta which is - they're continuing to move their click-to-play forward, which click-to-play is the technology which prevents vulnerable plugins from running without the user's permission. Now, I didn't have to click it just now when I was playing with Flash, and it was vulnerable and

obsolete. So it must not be in...

TOM: Not perfect.

Steve: Well, or not in the version that I've got yet. What's nice is that they're doing something else in this latest release that'll be out, for some reason I have the number "17" in my head. We're all running in crazy version numbers these days, instead of - we were on Firefox 3 forever, and then 4 forever, and suddenly Firefox version numbers took off, incrementing at the same rate as the national debt. So it's just gone crazy.

But they've added something which the Chromium Project has been talking about in their blogs, which is enforcement of strict transport security. So we did an episode on strict transport security. The idea is that what we really want is we want to us the web with SSL as much as possible. So if sites support ubiquitous secure socket connections, that is, HTTPS, then the way to make that even stronger is for the site to be able to declare to the browser that it supports strict transport security.

And what the browser will then do is, if any URLs are in the page which are not HTTPS, this permits the browser to upgrade them to SSL connections, knowing that the site is okay with that. So the benefit is that it prevents various types of man-in-the-middle and JavaScripting attacks where, by filtering the page coming in and removing the S's from the HTTPs, it would be possible, for example, to get cookies that are supposed to be protected by SSL security if they were not properly flagged as secure-only cookies, you could get them to leak out in the clear. And of course this is famously important for things like using laptops in open WiFi networks like Starbucks uses where there's no wireless security at the hotspot. And...

TOM: Should you use something like HTTPS Everywhere in addition to this?

Steve: Well, the two interact. What you want is, if the site says that it supports this, then the browser is able to itself update the security. Now, the problem has been, though, that if there was initially a man in the middle that prevented the server from making this declaration, that is, the way this is done is, when you initially connect, the server sends back a response header set, as it normally does in answering every browser request, but it adds a header that is "strict-transport-security:" and then an age for that to be enforced and some additional parameters, like whether that includes subdomains from the root domain of the site. So there has still been a problem with this, which is, if that initial declaration is filtered, then the user agent, or the browser, never gets the news that the site supports strict transport security.

So what Firefox is now experimenting with, and this is part of the Chromium Project, is automatically bringing up strict transport security preemptively for a large and growing list of previously known, built-in domains which support it. And this is a list which is large already and growing. So for all the servers that are known to support strict transport security, the browser itself would have knowledge of that and would never initiate a non-SSL connection to that domain or subdomain.

So this is really nice. This is, I mean, what we're seeing over the last couple years as sort of this awareness has been growing that the future is cloud and connected and more of our experience is through the browser, we're seeing real maturation of browser security and technology in really useful ways. So this is just neat, the idea that Firefox and Chrome will preemptively know which sites are able to support SSL, even if you've never visited them.

So this max age, the Max Age header allows the browser to be taught that a site supports strict transport security, but you still need to get it the first time. And so there's that little bit of a wedge that malware could exploit if the browser wasn't told, or if you were using a new machine, and you had never visited that site. So the browser, again, wouldn't have the information that that site is able to support STS. So this is just, you know, it's a small thing, but it's one fewer ways of malware getting into, getting hold of our browsers. And of course browsers are becoming increasingly important.

TOM: And who updates that whitelist of the servers? Is that administered by Mozilla?

Steve: It's just built in. It's built into the core browser code. So now we're seeing much more smooth and continuous updates in the background. So, I mean, Chrome is updating itself. Every time you launch it, it verifies that it's got the latest and greatest. And what we've seen in the last few iterations of Firefox is that they've been making their own update process increasingly transparent so that they're following that Chrome model of taking responsibility for keeping the browser current all the time.

TOM: And is it Mozilla that's making the list in the first place?

Steve: I don't know. I went to - I have the link in the show notes, src.chromium.org. And it's a JSON file: transport_security_state_static. And so that's built into the browser code base. And it's a pretty comprehensive list. It's down toward the end of the list.

TOM: It's probably the project, the Chromium Project, then, that's keeping it up to date.

Steve: Yes.

TOM: So you have a little HSTS guacamole and a little HTTPS Everywhere rice, and then you wrap it up in a VPN burrito, and you've got delicious security.

Steve: Right. Although speaking...

TOM: Yeah, let's talk about Chrome v23. We mentioned that briefly earlier.

Steve: Yup. Yeah. So Chrome with v23 has been the last browser, the last major browser to finally add support for the Do Not Track header, the DNT:. Microsoft, Mozilla, Apple, and even Opera have enabled it some time ago. Of course we've spoken about how Microsoft's enabling, well, I should say those have supported it some time ago. And they've all played by the official W3C rules of having it off by default, with the single exception of Microsoft, which in IE10 has enabled that header unless the user turns it off, which has created some controversy. Apparently the Apache web server group have stated that they're going to just ignore the Do Not Track header when it is expressed from IE10 because it breaks the rules.

Now, Google has done it in the standards-compliant fashion, as has Firefox and Apple and Opera, where it's off by default. But not only is it off by default, but it's quite well hidden by default. So I went looking for it. And you go to Settings in Chrome, and if you look there, it's not present. So you need to click the Show Advanced Settings at the bottom of that page. Then that expands another block of things, which then exposes the Privacy option. And under Privacy, under the Privacy section, the last item there is "Send a 'Do Not Track'," it has in quotes, "request with your browsing traffic." And if you click that, it pops up a notice that sort of explains it. And it's interesting because you have to then say Okay to that. The first time I just hit Cancel because I thought, okay, I'm just kind of canceling this notification. But it disabled the checkbox. So it's like, okay, it's

there, but they're not really encouraging you to click this.

So when you do click it, what pops up says "Enabling 'Do Not Track' means that a request will be included with your browsing traffic. Any effect depends on whether a website responds to the request and how the request is interpreted. For example, some websites may respond to this request by showing you ads that are not based on other websites you've visited. Many websites will still collect and use your browsing data, for example, to improve security, to provide content, services, ads and recommendations on their websites, and to generate reporting statistics." Then they have a "Learn more" at the end. And if you click that, then it takes you to a page that asks the question, "Does Chrome provide details of which websites and web services respect Do Not Track requests and how they interpret them?"

TOM: It's kind of a funny answer, too, isn't it.

Steve: Uh-huh. So they answer their own question, saying, "No. At this time most web services, including Google's, do not alter their behavior or change their services upon receiving Do Not Track requests."

TOM: So nobody uses it. But just so you know.

Steve: Well, and this is why lots of people just think it's nonsense. My position has always been it's better than nothing. And that once upon a time the web was regarded as nonsense. And there was this whole chicken-and-egg problem. People who were arguing against the Internet were saying, well, no one's going to go there if there aren't sites there. And no one's going to do sites because no one's going to go. And it's like, okay, well, that all worked out.

TOM: Yeah.

Steve: So I'm convinced that this is progress. And this is the way, this is exactly the way I've said it was going to happen. I've said...

TOM: If Microsoft doesn't ruin it, though; right? Because the idea is that you need to get most of the legitimate websites onboard to respect the Do Not Track, and then it's useful. But a bunch of them are now getting upset because Microsoft turned it on by default, as you mentioned, breaking the rules. That's starting to muddy the waters of whether it's going to catch on or not.

Steve: Well, this isn't going to be an easy path.

TOM: Yeah, yeah.

Steve: I mean, the big problem is that everyone argues against this, saying, well, it's voluntary. It's like, today it's voluntary. That's fine. Users are expressing their opinions, and so we'll get some sense for that. Now, you're right about Microsoft. But of course it's only in IE10 of that particular browser version, which they have said they're going to make available under Windows 7, as well. So sites know which user agent, which browser you're using, and which version. So, for example, Apache can ignore the Do Not Track header on IE10, but choose to make it available for other sites running on their services for other browsers, or even other versions of Internet Explorer.

TOM: But the sites have to get to critical mass. And if they don't have IE, then it starts to be a harder argument to get them to bother respecting it. That's what I worry about,

anyway.

Steve: What I expect we will see during Obama's presidency...

TOM: During the next four years.

Steve: ...so that gives us four years, we will see some legislation eventually. It'll be soft, and it'll get stronger, and ultimately it will end up being legally enforceable. And I don't think that's a bad thing. I think that this is the way we're going to get there, and I'm happy that I'm talking to you, Tom, and not Leo as we discuss this.

TOM: Well, let's talk about something fun: GPU accelerated video in Windows.

Steve: Well, yeah. That's another thing that's in Chrome v23, is in their experiments, what Chrome v23 adds is video playback in the browser now uses the graphics processing unit to essentially lower the power consumption. Since the GPU has got hardware specifically designed for rendering video, it's able to do that with a built-in hardware rather than lots of software. So you end up, they're saying, getting about a 25 percent increase in battery life playing video in Chrome 23 on a laptop versus Chrome 22 and presumably other non-GPU-accelerated video browsers. So that's just, again, this is all good because we're just sort of seeing browsers moving forward. And Google is paving some of these roads. Firefox is paving others. They're having to be cross-compatible and do the good things that the other one does. And so it's - and then Microsoft is sort of coming along, too, trying to keep up with the various standards that the other guys are pioneering.

TOM: Anything that gets me more battery life, I am very much in favor of, wherever I can get it. Now, this next thing, the per-site permissions, reminds me a little of NoScript.

Steve: Well, I'm very excited. That's one of the huge movements forward. In fact, NotScript was the extension that I have been running in Google, in Chrome so far.

TOM: In Chrome, yeah. I run that one, too, on Chrome.

Steve: And so I'm happy that now what we have is we have added, also with v23, is per-site permissions are built into Chrome.

TOM: That's fantastic.

Steve: Once again, they're buried. So they are essentially nonexistent in the same way that the DNT header option is nonexistent for non-expert users. But it's there. So once again you go into Settings and then Show Advanced Settings. And then under Privacy, oh, well, first I should say the way you see this is when you're looking at any site's page, you click on the little site icon to the left of the URL.

TOM: So where the lock goes?

Steve: Exactly, the little lock guy. That now pops open a dropdown showing you, for that site, a very comprehensive set of settings are in place for cookies, images, JavaScript, handlers, plugins, pop-ups, locations, notifications, full screen, mouse lock, and what they call "media." And so normally a site will default to the default settings for the browser, and then you're able to override either always-on or always-off for all of those things individually for that site. So I thought, okay, well, to make this work the way, like, for real safety, we need somehow to turn JavaScript off, disable JavaScript by default.

So then I went digging around, how do I change the defaults globally so that then I enable it for specific sites. And so that's where you go into Settings, Show Advanced Settings under Privacy, and then click the Content Settings button. And that pops up a dialogue with where all of these features can have their defaults changed, and it also explains what they are to some degree. For example, it has an item for handlers which allows sites to ask to become default handlers. And you can say no, don't bother me with that. Or plugins, you can choose "Run the plugins," "Click-to-play," or "Block" them. And there's something called "Mouse lock." And I thought, what the hell is mouse lock? Well, it allows sites to disable the mouse cursor.

TOM: Oh.

Steve: And you can either allow that; you can have, if the site requests it, you can have Chrome prompt you for whether you want to allow that or not; or you can block it outright. And then there's also two different options at the bottom which are nice to see, which is for Flash camera & microphone control, where sites can request it or sites can require it. And then you're able to either have Chrome ask you or block you. So, oh, and right there, all of those support per-site overrides. And there's management buttons that allow you to pop up and see what overrides you have in place on a per-site basis.

TOM: Well, that's great because something like Google Hangout I might say, I want it to have access to my camera and mic because I trust it. But I don't want anything else to access that thing.

Steve: Right. So what we're getting, and this is really nice, the only thing we don't yet have is side tabs, and so I can't use Chrome yet because I've got 75 tabs open over in Firefox right now. But we know that Google understands that some people organize their lives around tabs, and so they didn't like having the tabs on the side. But there has been blogging from Google saying, yes, we understand, we're going to come up with something, some sort of really cool tab management. Which reminds me that right now I'm using an add-on over on Firefox to move the tabs onto the side. Firefox will be getting native side column tabs. And of course this is all as a consequence of the fact that screens are becoming wider than they are tall. And so it's making more sense to get the tabs off from the top and move them over to the side. In which case you can just have so many of them. It's wonderful.

TOM: I like being able to move them around a lot, too, which is the one thing that Chrome has been nice to. But any of the extensions for Chrome that I've tried make that buggy for some reason. So having it native I think would be desirable for that reason, too.

Steve: And I want to see what they come up with. I'm really happy with the UI, with the Chrome UI. I'm still a Firefox user because I've got my tabs on the side and the add-ins that I like. But, boy, Chrome is maturing as a power user's browser with this v23 release, where we can change the global defaults to lock it down the way we want to; and then, on a site-by-site basis, just as with NoScript, we can flip these things back on again. It's getting close.

TOM: The only thing that bugs me about the per-site permissions is that I didn't have it up till now. It's like I spent so much time with NotScript, customizing it, teaching it. It's like, wah, why didn't you have this before? But it's great. I think that's awesome.

Steve: Yeah.

TOM: So tell me about this TNO cloud backup solution. What is this?

Steve: Well, I teased everyone about it last week because I received email from its author, telling me that the news was embargoed until yesterday. Well, I'm not sure what that meant because everybody else was tweeting me about it. And I thought, well, okay, wait a minute. How does everyone else know about this, and I do, too, but I'm not supposed to talk about it? Anyway, so back on May 2nd I famously tweeted, and this is @SGgrc, I said, quote, "'ARQ' Cloud backup for Mac to Amazon S3 w/TNO Crypto & iOS viewer. The deeper I dig, the more I wish I was a Mac user."

TOM: Wow.

Steve: And then I did a little shortened shortcut to HaystackSoftware.com. Now, what the news is, is that with v3 that was released, I guess yesterday, ARQ for the Mac now supports the Amazon Glacier service. Glacier is the - and we discussed it on the podcast a few weeks ago - is an interesting and very appealing long-term storage solution from Amazon. The idea is that, if you don't need rapid access to your cloud data, they will make it available to you eventually, but at a much lower cost. In fact, it's one penny per gigabyte per month. So, okay, let's do a little math.

TOM: Yeah, that's...

Steve: That's 100GB, that's $1 a month for 100GB, or 10 bucks a month for a terabyte. So that's really nice. ARQ, one of the reasons I like ARQ is that it is a flat fee purchase, $29. You buy it per computer, one time, and then you own it. You can upgrade from ARQ 2 for 15 bucks to ARQ 3. And I did look at the crypto. Anyone who's interested who is a Mac user, I know that Leo switched to it. The UI is very Mac-ish, and that is to say it's not super techie. It's very friendly, easy to use. So it's HaystackSoftware.com. And this, I mean, to me, the fit with S3 Glacier is perfect for long-term archival storage.

TOM: Yes, absolutely.

Steve: People might say, oh, well, but I've got 10TB of data, so that would be $100 a month. It's like, whoa, wait a minute. What have you got, 10TB? Sure, of, like, crap that you ripped from DVDs or downloaded from torrents or who knows what. My point is that that stuff isn't original content. Anybody who's, like, creating anything original, even photo collections are not going to be that massive. So, and this is a great deal. So I just think it makes a lot of sense.

TOM: Oh, yeah. This reminds me of the old Jungle Disk service, which did S3 for its backend. You paid for the - same sort of thing. You paid for the software, and then you just subscribed to S3. I stopped using that, I don't know, four or five years ago because the S3 fees were getting too big for what I had.

Steve: Right. And Jungle Disk...

TOM: So Glacier makes this work well again.

Steve: Yeah, Jungle Disk was our choice as a high-quality TNO, Trust No One, crypto. They did it right. And ARQ has done it right, too, so that your system knows the password. It's symmetric crypto. It never leaves your machine. All that's being sent up there is pseudorandom noise. And Amazon is doing nothing but just storing your random numbers, as far as they can tell. As far as we know, no force on Earth is able to decrypt that stuff. It's really good crypto. So I did vet it completely and looked at it carefully, and

that's why I recommended, for Mac users, HaystackSoftware.com was a great solution. So I really like the fact that you can put that much data up at Amazon. And, hey, for things like archival backup, you don't need instant access to it. So Glacier is a really nice tradeoff, I think.

TOM: That's one of the reasons I'm using this little ZaReason laptop now, even though it's only got a 64GB flash drive, is I get to use a flash drive. And I've got encrypted backups of the stuff I don't need every day, and I've got a Dropbox with the stuff that I go back and forth and I do need access all the time, if I need a little more space than what this thing has.

Steve: Yeah.

TOM: So it's perfect that way. So tell me, how do you pronounce this next thing [PortQry]? Is it "port query"?

Steve: I think so. I mean, I know that's what it's short for. It's nice, too, because Google only produces the proper results for it. It's not a new utility, but someone who tweets to me often, his Twitter handle is Captn_Caveman...

TOM: It does make me want to yell like the cartoon.

Steve: His real name is Joachim. I think he's over in Germany somewhere. He's been a follower and a frequent contributor to a Twitter stream for me for years. Anyway, he sent me a link to a page of free Microsoft security stuff. And in browsing through it, I mean, I saw the EMET stuff and random things that we know about. But there was this PortQry that I thought our users would get a kick out of. And it's free. It's old. It's back from 2003, so nine years ago. But it's small and lightweight.

So if you just Google "PortQry," what it is, and the reason I wanted to bring it up to our users' attention, or our listeners' attention, is that it is a local port scanner. So there's always been interest in, within a local area network, scanning your own machines. And this is very small and lightweight. It's a few hundred K and runs on anything post NT, so all the way back to Windows 2000 and on. So I just wanted to bring it to people's attention. It's a simple command line-oriented scanner and shows the status of ports on machines and supports a number of protocols, so it's able to check them for different protocols, and written by a Microsoftie some time ago.

TOM: Very cool. Captn_Caveman is good with that stuff.

Steve: Yeah.

TOM: There's a security movie up on Kickstarter that apparently is your fault.

Steve: I guess that's the case. I wanted to give a little shout-out and let our users know, there's a guy who's actually in my neck of the woods, in Costa Mesa, California, Jonathan Schiefer, who tweets @jschiefer. But he has the more memorable - and I was surprised he was able to get this domain. TheRootKit.com is the domain.

TOM: I'm surprised Sony didn't have it, yeah.

Steve: Yeah. If you go to TheRootKit.com, it bounces you over to his Kickstarter page. And so what he tweeted me is he says, "@SGgrc, I'm doing a Kickstarter for my movie TheRootKit.com about computer hackers. Most of my research came from Security Now!.

Thanks." So I went over to the page. And under his "Who Are You," he describes himself. He says, "My name is Jonathan Schiefer. I'm the writer, director, editor, and one of the producers on 'The Root Kit.' I've been writing scripts for 10 years. I've been making movies, short films, commercials, music videos, and industrial videos for the past three years. But that's what I do. Who am I? I'm a person like you, trying to make my dreams come true, trying to make a difference. 'The Root Kit' is my best attempt to date." So as of course is the case with Kickstarter, he's hoping to generate some critical mass of contributions in order to help produce this movie. Anyone who's interested, I encourage you to go to TheRootKit.com, check it out, and maybe consider supporting him. I am certainly going to do so.

TOM: Yeah. I'm going to maybe take a look at this. We could talk about it on Frame Rate, as well.

Steve: Yeah, that'd be great. So I did find a nice testimonial for SpinRite that I wanted to share with our listeners. This guy is rather formal. He said, "Dear Mr. Gibson." And then he says, in parentheses, (Steve). It's like, okay. He says...

TOM: Is that your father? Do you do that thing, "Mr. Gibson is my father"?

Steve: He says, "I just got off the phone with your helpful, knowledgeable, and nice sales/customer service lady. I had my SpinRite moment, a major catastrophic hard drive failure caused by power outages and surges from the fast-moving storm system here in the East." Whoops. So he says, "The surge protector did not work."

TOM: Ohhh.

Steve: "The drive was caught in an endless loop, and I knew there were disk errors from using other lame third-party utilities," he says, "three different ones. My OS files were there, but my efforts to repair did not work. One tool said I had, quote, 'disk errors,' unquote, and the restore points, of course, would not work, either. The drive's controller chip was possibly fried, and my drive seemed toasted, as well. Of course I had an image that was about three weeks old. But the drive had the image on it in a different partition, and both were now unreadable.

"So I purchased SpinRite 6 after hearing about it frequently on the Security Now! podcast. After figuring that it was my only option, as the malfunctioning disk could not be restored from my save image or backup, and the reinstallation was also inaccessible, I needed to get it working again so I could image or do a repair installation of the OS. I used your well-designed program to create a bootable SpinRite CD. I started it after following the easy-to-follow instructions and waited and watched. SpinRite automatically recovered and repaired 12 sectors, and I was then able to boot and had the disk detected in another machine as my motherboard was RMA'd to Intel." I guess it really did fry him.

TOM: Wow, yeah.

Steve: "After a repair installation, the machine is back to normal with no files lost. It boots fine, and SpinRite saved the day. It was well worth the cost and time saved, easy to use, and, after nearly 20 years of making systems, this was the first I needed to use it. But I know it will not be the last. Thanks for a great and easy-to-use product. David G. Speigner in Marlborough, Connecticut." So, David, thanks for sharing your happy story.

TOM: When I saw it was from Connecticut, I wondered if that was going to be a hurricane or storm-related story. So, yeah. I'm sure there's a lot of people in similar

situations out there with all of the havoc that that caused.

All right. It's one of my favorite things to do. It's time for a little listener feedback. This is #155. I was talking to Steve before the show. It's been almost a year since I filled in on Security Now!, and the last one I did was a Q&A.

Steve: Yup.

TOM: So let's start off with Sean O'Brien in Texas, tweeting you the following question: @SGgrc, please don't say it's hard to factor large primes. It's impossible to factor any prime.

Steve: Yeah, so, okay. To everyone who sent this - I ran across a number of similar people groaning at me in the mailbag. And, okay. So I guess I had been misspeaking. This is, okay, first of all, I would argue that it's not impossible to factor any prime. It's trivially easy.

TOM: Exactly. I was just thinking the same thing. I'm like, you can do it once.

Steve: Right. So, and we all know that what I have been meaning to say every time I say "It's hard to factor large primes" is that it's hard to factor the product of large primes. So I've been taking a shortcut. I apologize to all of our pedantic listeners. I want to, I mean, I can't say that they're wrong because I'm a perfectionist, as well. I mean, we know I am. And so I have been misspeaking. So I stand corrected. To all of you who said, Steve, you know, you keep talking about factoring primes, it's like, okay, factoring the product of large primes. So if I make the mistake again, we all know that we all know what I mean. I'm going to try not to make that mistake. Now I'm sure I'll be so self-conscious about it that I will say it correctly from now on.

TOM: No, you're absolutely - you're a good man, Steve, for owning up to it and doing that because I've done exactly what you're doing right now. It really does help you not make the mistake again, for sure.

Andrew in Arizona wonders about decrypting files. He says: Steve, I have a question about cracking an encrypted file or block of text. I understand how it's possible to use a dictionary attack to figure out a string of text, for example a password, that has been hashed. Hash the text. If the hashes match, viola [sic]. But I am wondering how the process works for encrypted files. Where would you start, and how would you know when you've guessed the right key?

Steve: Okay. So that's really a good question. There are two approaches for - so the idea being you've got a blob of what looks like just pseudorandom noise. And we assume that the reverse-engineering information is available to tell you what crypto algorithm was used to encrypt it. So you know what crypto algorithm you would use to decrypt it. Meaning that the only missing piece of information is the key, which is, what, 128 bits, 256 bits, whatever.

So the question is, how do you know, how do you go about doing a brute-force attack, essentially? Well, there's two solutions or two approaches. One of the things that normally comes along with encryption is authentication. That is, not only are you going to decrypt something, but the algorithm which is used or a secondary algorithm also authenticates that it has not been modified. So it's not enough just to apply the key and get something that appears good. How do you know it wasn't cleverly tampered with so that it looks fine, but it's actually different data than was originally encrypted?

And so normally most encrypted modes either authenticate at the same time, or there's an authentication pass that can be made. So you could apply keys and then, I mean, like, brute-force different keys and apply the authentication portion to see whether the block authenticates and, if it does, then perform the decryption. In some cases the authentication process is faster than decrypting it. So that works. But what you can normally do is only decrypt the beginning of the file, which is going to be faster than either of those two approaches.

For example, you might have a TrueCrypt volume which has a well-documented, publicly well-known format and header. And so the idea would be you successively guess what the key might be and apply that key to the beginning of the TrueCrypt volume and see if you get something that makes sense. See that the result of maybe the first 1K looks like a valid TrueCrypt header. It's extremely unlikely that you would guess a key which makes the beginning of the volume decrypt perfectly, but it's actually the wrong key, and the rest of the volume decrypts incorrectly. I mean, it's possible, but really unlikely. But if nothing else, only do the beginning till you get something that looks like it's good, and then apply the key to the balance and see if the whole thing makes sense.

But so anyway, the point is the way to try it is just guess. Begin the decryption of the file and go only far enough in till you get enough, till you can see enough of what might be the decrypted content to decide, oh, look, it's English. I mean, it's no longer gibberish. This looks like a document.

TOM: I'm seeing header stuff, sort of, yeah.

Steve: Exactly. Exactly. No need to go through the entire thing. It'll just take you several lifetimes. Just get enough to see if it makes sense. And in fact you could actually apply a clever decryption scheme which really does very minimal decryption, only goes in a few blocks' worth of deciphering and immediately backs out and rejects a guess if it's clearly not correct. And if it looks like, ooh, it might be correct, go a little bit further. So anyway, that's the approach to use if you're trying to just tackle something like that and don't have any idea where to start, but you just have to brute-force a file.

TOM: The header is like that little nub that you can start scratching away at, so to speak, and get in there and erode the rest of that encryption.

Steve: Right, well, yeah, it provides a structured, sort of a structured metadata for the file. And you can normally quickly see whether it's, I mean, most decryptions are going to just produce complete noise as opposed to something that, oh, look, that might be a header.

TOM: Yeah. By the way, he wrote "viola." Everyone in the chatroom was on me, and even Jammerbabe was like, "Viola, not voila?" No, that's what he wrote.

Steve: He did write - he did, you're right.

TOM: Matt in London shares a useful tip of Microsoft email users: Hey, Steve. I use Outlook Express always set to open email in plaintext. However, if I see something that looks odd, I point to the unread email, click and drag onto the desktop, where it becomes a little .eml envelope. I then open Notepad and drag the envelope onto the Notepad window. There I can see the headers and read in plaintext, just in case someone has a clever zero-day Outlook hack.

Steve: That's a great tip, and so I wanted to share it with our listeners. And it's simple to

do. I often do the same thing myself. I've got a little hex editor that I ran across decades ago that I've carried with me from one Windows update to another. It's called Hex Edit. And at any time something just looks a little suspicious, it's an attachment that I really want to open but I'm just scared to open it, I just say open in my little hex editor. And that allows me to browse around and kind of see what's in there without committing to opening it in one of the typical proper format handlers. And so this is something very similar, but using tools that everyone has. So I think Matt's got a great little tip.

TOM: That's great.

Steve: Just drag the file to the desktop and then drop it on Notepad, and then you're viewing it in something that is just text, and Outlook has no chance to touch it.

TOM: Yeah, everybody's sharing their editors now in the chatroom. This is great. Tamahome says Emax. Web7157, Notepad+. Strength says MadEdit. I'd probably say Gedit. There's lots of different approaches to that, but it's a useable tip no matter what kind of software you've got. That's pretty cool. I just usually delete the email. I'm like, eh, don't want anything to do with it.

Student 17 in New Zealand shares an OAuth observation: Something I noticed when logging into a site using OAuth to authenticate with Twitter was that, when my browser jumped to Twitter, LastPass recognized the site and auto-filled my login details. Since LastPass would not auto-fill on a domain that looks like Twitter but is not Twitter, wouldn't this be a way to protect yourself from spoofing?

Steve: That's a great thought. And the answer is yes. One of the things that I have noticed is, since I'm an avid LastPass user, sometimes I'll go to a page, and I'm expecting LastPass to see fields that it ought to populate. And it's like, okay, wait a minute. Why is LastPass not engaging and working? So I've seen myself that, once you get used to having LastPass around to fill in the fields for you, if something happens and it doesn't, you say, okay, wait a minute, what's wrong? Now, what I've noticed is sometimes I just have to kind of click in a field to wake up LastPass's observing stuff. I have to give focus to one of the form's fields, and then LastPass will say, oh, you want me to fill this in.

But we've been talking about the problem of - and here we are, predicting the future again. There is going to be an exploit where people get used to using OAuth, the so-called login using your Twitter account, login using your Facebook account, which we're seeing more and more because it's a low-friction way of acquiring credentials from someone on a site where you aren't known, using a site where you are known. And it's very convenient. But the problem is spoofing. We are going to see instances of OAuth spoofing as this becomes increasingly popular. There's just - there's no other way about it.

TOM: And it's so dead simple to make a page that looks like it's OAuth.

Steve: Oh, yes. And once people get used to doing it, they'll just say, oh, yeah, of course, and won't notice that the URL is Faceb00k.com instead. But again, as Student 17 in New Zealand observes, LastPass would not engage. And so it's sort of nice to have it there and to have you say, wait a minute, why didn't LastPass log me in?

TOM: My favorite is when they use the Cyrillic O's instead, so you can't even tell by looking.

Steve: Yup.

TOM: [Ricardo] J., Toronto, Ontario, Canada wonders about hardware made in China: I have been listening to your podcast for about two years now. And while I don't necessarily understand the propeller hat episodes, he calls them, I do learn a lot from the show. I own SpinRite; and, though I have not had to use it for a catastrophic HD failure, it did fix a mysterious random slowness issue with my mother's laptop.

With the recent congressional warnings about Chinese telecoms being allowed to enter the U.S. and possibly spy on the communications they are helping to deliver, has anyone given thought to the hardware made in China? Even if Chinese companies are kept out, I would have to guess that most of the hardware that is used is either entirely or some part is made in China. Do companies that sell networking equipment tear down random samples of retail products to see if the components have been modified? For that matter, does a consumer electronics company like Apple perform tests like this? I would think that putting compromised equipment into your network would be just as bad as allowing a person to walk in off the street and sit at a terminal.

Steve: Well, I'm uncomfortable picking on China or any one nation. And we do know that apparently a lot of hacking is coming from Chinese IPs. The government officially denies any state sponsorship of that, and I want to take them at their word. But it is the case that there exists some international tensions between major global competitors.

TOM: There's a lack of trust.

Steve: Yes. And Richard's point, or Ricardo's point, sorry, is dead-on right. I mean, we've seen instances where microcode on products has been mistakenly, you know, already contained malware. Flash drives were shipped with malware on them. I mean, little mistakes like that have happened. And if a state sponsor like China really wanted to infiltrate this country, or any company that was importing their exports, they could certainly produce a modified network chip, that it would be, I mean, you'd have to really go to some lengths to see that there was something unusual about it.

I mean, state-level sponsorship can produce a microprocessor with a whole chunk of private microcode that you only access if you put specific data into certain registers and execute a certain instruction, and that takes you off into a different area of microcode. I mean, this sounds like science fiction, but it's absolutely possible.

TOM: And the U.S. does this to other countries. I mean, the U.S. was behind the Stuxnet attack on Iran. So it's not just China or Russia or - all the governments are in this race to spy on each other, for lack of a better word.

Steve: Well, yes, and...

TOM: And I was going to ask you about that detection because the U.K. I know has a partnership with Huawei, Chinese company, where they will tear down Huawei equipment at random, so Huawei doesn't know what equipment's going to get checked, and look for vulnerabilities. But what we're saying is there might be vulnerabilities that they just can't catch.

Steve: Oh, I mean, yes. For example, it would be necessary to, in the case of a processor, to completely reverse engineer from the silicon the die itself and all of the microcode and understand what every bit of it does. I mean, it's exactly like taking a huge piece of software and trying to find something that was deliberately done, a

backdoor put in a large software product. The only way to do that would be to reverse engineer, and you wouldn't have the source. You would have only the binary. You'd have to disassemble it, decompile it, turn it back into source, and then understand what every single line does and be looking at it from the standpoint of how can this be abused.

The problem is, this has all become so complicated that it is absolutely possible for non-intentional, non-designed-in features, if you're going to call them that, to hide, almost in plain sight. I mean, even simple network switches now are managed and have - you're able to log into them, give them passwords. I just bought a little five-port gigabit switch, and I couldn't - I was amazed at the technology that is in this thing. And it's like, well, do we know that it doesn't respond to some secret sequence of packets that come along and allows it to bypass the passwords that the user has installed? We don't know.

TOM: So what do we do? Because you could say, well, I'm just going to only buy if the chips are assembled and made in the U.S. But it could be that somebody in the U.S. is spying. It could be anybody that puts the stuff in there.

Steve: Well, yes. And even if the chips, even, for example, if China was building iPhone 5s using Apple's A6 chips, which Apple supplied, again, if you've got state-level sponsorship, nothing prevents a truck from making a left turn and a replacement truck takes its place and has chips that look exactly like the Apple A6 processor. I mean, I don't know how - you ask, what do we do about it? It's like, well, you just don't worry about it.

TOM: [Laughing] You just live; right? And I guess you look for the effects of the sort of thing; right? You try to keep secure. You keep encrypted. And you try to make sure that you're not opening yourself up to as many vulnerabilities as possible.

Steve: Well, yes. One thing somehow we've never really discussed on this podcast in all of the podcasts we've done is the conclusion that the best security people have come to, which is that monitoring is the only way to know. That is, you can have firewalls, you can have antivirus, you can have all of these sort of static defense systems. But actively watching the traffic, monitoring what comes in and out of your network and asking questions about, whoa, what's that connection over to there? I mean, it requires the right balance of a chill pill and knowing when to worry because, if I ever do a netstat/an, I think that's what it is, or maybe it's /abn, that shows you all the connections, I've got stuff connecting out all over the place. It's like, wait, you know. And so if you sit there and try to explain every single one of those, it's like, okay, well, I've got so much software now, we all do, running on our machines, it's busy talking out to the Internet. It's very difficult to really audit that.

TOM: You've just got to do spot checks.

Steve: But it's really the only solution is to really, really closely monitor what's going on.

TOM: Or get a backscatter scanner for all your - no, never mind. That's it. John Lockman in Ottawa says that UPEK's security cannot be fixed. Is it UPEK? Is that how you pronounce that?

Steve: Yeah.

TOM: Okay. The entire problem with the UPEK model of using the fingerprint as a sole authentication factor is that Windows requires the password for other purposes, encrypted files on NTFS, for example, and thus they must store the password in some

way. Even if they used the most secure algorithm in the most secure way, the password must, at some point, be decrypted into plaintext before it can be used to tell Windows to log in as a specific user. AuthenTec can increase the security all they want, but anyone who reverse-engineers the encryption on the password will be able to decrypt the password simply by definition. The only reasonable approach is to only use fingerprints as a second factor for authentication, the "something you have," obviously, because you've got it with you.

Steve: Right, yeah, your finger. So I agree with John in principle. First of all, I love his conclusion: The only reasonable approach is to only use fingerprint as a second factor for authentication. I absolutely agree that that makes sense. That's not as convenient as the whole, like James Bond, oh, cool, this thing just scanned my finger, and now I can log into my computer using nothing else.

That said, there are two approaches which do solve this problem. One is the use of the so-called "trusted platform module," the TPM, which exists in many laptops. And it's sort of having a hard time achieving critical mass. But the idea being, and we have done a podcast on it in the past, is that it absolutely will not export the things that it contains, and that it is potentially possible to design a system where features of a fingerprint are used in order to unlock it, and without those features it will not produce the password. So you keep it from Windows. Now, it is the case that, if you've got malware watching what goes on, well, then, you've already got malware in your system upfront.

The second possibility is to actually use features of the user's finger as the password. The problem is, those are rather soft. The fingerprint changes angle; they swipe it differently; there's a different level of grease on the person's finger today as opposed to yesterday. And so it's easier to match a fingerprint against the learned template than it is to actually use features of the fingerprint as the password. In which case, if you did that, you would actually need that fingerprint in order to decrypt the stored password; and, once again, it would be safe.

I would argue it's probably possible to make that second approach work, though certainly UPEK hasn't done that. They've said, oh, yeah, look, this matches the template, and then they perform a rather weak decryption of their weak encryption in order to provide the password. So in principle I agree with John. I think he makes a good point, that there's a fundamental problem with using a fingerprint as the sole authentication, and using it in a multifactor setting really does make the most sense.

TOM: Yeah. That makes sense to me. I mean, anything is going to be turned into digital information at some point, and that digital information then is no longer the thing itself, so that kind of is what John's pointing out, I guess.

Steve: Well, yeah. I guess the distinction, the reason I chose the question and wanted to discuss it a little bit, is that there's a difference between deciding that there's a match and then unlocking the encrypted password because then all you have to do is you find the where-is-the-match decision, and you tell the software, oh, there was a match. And then it unlocks the password. So that's the less secure way to do it. The alternative is to actually use the information in the fingerprint as the decryption key. And the problem with doing that is if you don't make it a weak enough set of features that they're not going to vary from one swipe to the next, you'll never get the same key twice, and you'll never be able to properly decrypt it. So there's a little bit of a Catch-22 there, as well.

TOM: Yeah, you need to limit the number of times it has to be turned into something else, and then it becomes more secure. But I guess we need technology to catch up with being able to have that be valid every time. Do you use - have you ever used CLEAR, the

transportation security line jumper thing? A lot of people don't like the idea of CLEAR because they're like, oh, somebody's got a database of all of my information. But I kind of figure somebody already has all that information in a database.

So what CLEAR does is they do a fingerprint scan, and then you get to skip the line at TSA. Now, you still have to go through whatever security screening. You don't get to skip the security screening. But you do get to skip the line. And they have a very forgiving fingerprint scanner. But it makes me wonder, now that we're having this conversation, just how secure it actually is.

Steve: Yeah, well, we could all skip the security line, and we'd just be fine, too, so...

TOM: Yeah, I know. Well, that's kind of what we were just talking about your network security applying to people, too, doesn't it.

Steve: [Laughing] Yes.

TOM: Jeroen van den Berg (or to Leo, John of the Mountain) in Waddinxveen, which I'm sure I didn't pronounce right, near Gouda, The Netherlands, says that, Steve and Leo, you guys are criminals. He says: I was just listening to Episode 375 at which you talk about having removed DRM from a Kindle book that you owned, and that there are tools available to do so. That sir, as crazy as it sounds, is illegal. I was shocked, shocked to hear that you did not know this, as legitimately as you may feel about it.

According to Section 103 of the DMCA, quote, "No person shall circumvent a technological measure that effectively controls access to a work protected under this title," end quote. The Act also prohibits the distribution of tools that enable a user to circumvent access controls or controls that protect a right of copyright holder. And you cannot claim that it's "fair use" because the DMCA does not contain any explicit exception. That's what got the DeCSS guys in trouble. He also points to a Wikipedia article about anti-circumvention and says: Just wanted to point that out. Personally I'm against any form of DRM on any media because it prevents making a backup, lending, or selling something you legally own. So if you want a DRM-free eBook, pirate it. Better not to pay and be a criminal than pay and still be a criminal. According to John of the Mountain, that is.

Steve: So, okay. I didn't intend to imply that Leo and I did not know that that was illegal. I understood. Just to fill you in, Tom, I had a Kindle book with a low number of installs on its counter, and I was using it on my stair climber, which is a PC-based Kindle software. And I reformatted that drive and set it up and forgot, and didn't remove it and increase my count. And so when I wanted to put it back on the rebuilt machine, it said, oh, you've exceeded your install limit. It's like, what? So...

TOM: Happens with print books all the time. No, wait.

Steve: Yes.

TOM: No, it doesn't.

Steve: [Laughing] So anyway, so I thought, okay, well, I've been wanting to play around with Calibre, which I was pronouncing Calibre [ca-lee-bray], but Elaine informed me that she did the research, and it really was Calibre [cal-i-ber].

TOM: Calibre, yeah. I have Calibre, too. It's a good program.

Steve: And I just wanted to see if it worked. And so, sure enough, I was able to remove the DRM and read the book that I already purchased and I legally owned. And I know that Leo and I both know that that's a breach of the DMCA. Now, it's not a breach of my own ethics because I didn't give it to anybody else. I'm not intending to. I would argue about some aspects of the DMCA. I will take responsibility for having done that. At the same time, I don't think that I'm the target of the DMCA. It is more the DeCSS guys and people who are doing mass decryptions of Blu-ray DVDs and producing them at low cost and so forth.

TOM: Also the Library of Congress issues exemptions to the DMCA for certain things, like unlocking a phone that you own. Even though that technically would violate the DMCA because you're breaking copyright encryption, the Library of Congress has said, no, we're going to allow that. We're going to give exemptions. And those exemptions just changed recently. So it's worth checking, if you're concerned about this sort of thing, to find out if the behavior you want to do is in fact an exemption. I know that removing DRM from eBooks to make them accessible to text readers for blind people is an exemption that's allowed under the DMCA. Now, that's not what you're doing, I know, but...

Steve: Yeah, and the conclusion of better to pirate it and not pay and be a criminal than to pay and still be a criminal, I'm on the other side of that.

TOM: Yeah.

Steve: I mean, I'm proud to have purchased the books that I spend many hours enjoying. And if the DRM gets in my way for my personal use, then I'm going to remove it and not feel bad about it.

TOM: DRM-free eBooks are starting to gain some traction. Baen Books has been doing it for a long time. Tor, the sci-fi publisher, just announced that all its eBooks are now DRM-free. So you can find really good stuff, without even having to come close to breaking the law, that doesn't have DRM on it. I was just going to say I have been in the situation where someone very close to me had illegally acquired a copy of a movie that I watched with her. And what I did to make myself feel better about it is I bought a ticket. I didn't go to the movie. I just went online, Fandango, bought a ticket.

Steve: Nice.

TOM: So I'm on the other side of that, too, Steve, which is I want people to do the right thing, and I want the system to work so that it encourages you to do the right thing, so that John here doesn't feel like pirating is the best option.

Steve: Yeah, well, and I wanted to mention, you mentioned Baen, and they're the publisher of all the Honor Harrington novels, and all very available at very low cost and DRM free.

TOM: Dave Kodama in Cerritos, California also comments on Amazon books and DRM. He says: Steve, Amazon's DRM has always bothered me. So when I found out that O'Reilly books come DRM-free and include multiple formats such as PDF, I have made an effort to always purchase the books directly from O'Reilly. I have noticed that they often a little more than the same book on Amazon, but I try and support O'Reilly's business decision by voting with my wallet. Perhaps some of your listeners may be unaware of that option. For the most part, I buy only "throwaway" books from Amazon. If Amazon disappeared, I would not be happy, as I like doing business with them, but I won't be losing anything I really want to keep around. SpinRite user since version 3, or maybe 2, he says.

Steve: So I thought this was a very good point. I follow exactly the same policy with O'Reilly's books. I mean, I've got them all over the place. And now that they're offering them both in soft cover and eBook format, what I'll do is normally I'll buy both because they offer a discount. And on the electronic side, they make them available in every format, so EPUB, Mobi, PDF and a couple others. And, for example, when I was learning how to use the HTML5 canvas features on web browsers to do the magnetic recording waveform animation that I did a while ago, I had the PDF version of the book up in a PDF viewer where the table of contents column was there. I was able to do a search through the whole thing. I was able to jump around. It was just fantastic to have the book in PDF format.

So it's really convenient. And I'm the same way. Sometimes I'll be over on Amazon and almost click that I want to purchase it. And I go, oh, wait a minute, this is an O'Reilly book. And I go over to O'Reilly and buy it there. And then I download, like, all the versions that I could think I might want, EPUB and Mobi and PDF, and then sometimes have the softbound book sent to me physically so that I have it to flip through the pages, as well. So, yeah, I'm a 100 percent supporter of O'Reilly stuff for the books, the typically techie books that they offer.

TOM: I just published my "Chronology of Tech History" on Amazon Kindle Direct. I swear there was an option where I could say no, don't put DRM on it. Maybe I'm confusing that. But I wouldn't see why O'Reilly wouldn't have that option, too, when they're selling their books. So you might want to check and see. They may not have DRM on the Kindle books. I think Amazon says you can choose as the publisher.

Steve: For me, it's the fact that I can have it as a PDF.

TOM: Yeah, you get all the formats, I know. That's the best thing ever.

Steve: Yup.

TOM: Christopher Ursich in Lyndhurst, Ohio comments on OAuth with Facebook and Google, suggesting that we just look for HTTPS: In the past two shows you've discussed the problem with OAuth where the user is tricked into entering their credentials into a phony Facebook or Google authentication page. I think it's worth mentioning or reminding people that this situation is no different than entering credentials on any web page. You simply need to look for HTTPS, and that the domain listed is what you expect. This is always my careful practice. What burns my butt is how some sites will try to make the authentication window pretty by eliminating the address bar. In those cases, I just don't proceed. I find another way to log in, or I just decide that I don't need whatever the site is offering. Displaying the HTTPS and domain should be part of the OAuth spec. Fortunately, sites that use Mozilla BrowserID don't seem to misbehave this way.

Thanks so much to you and Leo for Security Now. Your coverage always prompts me to dig even deeper into the topics on my own. Listener since The Onion Router Episode 70, SpinRite owner, and aspiring ketosis-izer.

Steve: [Laughing] Well, I don't mean to belabor the OAuth issue endlessly, although we're going to do it on our final question, the next one coming up, as well. I do think that it's important. And I want to focus our listeners on it because, like I said, we're predicting the future. I know this is going to bite us. And Christopher's approach, while it sounds good, I think probably our listeners are far less likely to be caught out by it than the world at large. It's not our relatively small, compared to the global, listener base that I'm worried about because we understand these things, and we're cautious.

But there is no way my mom is ever going to understand HTTPS and whether it's an extended validation site and what it means that the address bar turned green, I mean, or even that she would look to say that it didn't say Facebonk instead of Facebook. She just would assume. I mean, she just - she wouldn't even look there. So this is a case where convenience and security are colliding. And I've got a bad feeling about that.

TOM: Well, and even educated listeners and viewers of this show will occasionally forget. You're human, you know? You make mistakes.

Steve: You're in a hurry. You're like, you know, your spouse is waiting for you, and it's like, okay, well, just one second.

TOM: Yup, yup. And you just click that Okay button without thinking one time, and all of a sudden you've lost all your data.

Steve: All it takes.

TOM: And you're Mat Honaned. I mean, that's a horrible cautionary tale, what happened to Mat Honan at Wired.

Steve: Yes.

TOM: A very sophisticated user. Not a dummy, by any stretch. And Blake Waud in Troy, Michigan does suggest an OAuth spoofing solution using security pictures and phrases: Hey, Steve. Started listening just before Christmas of last year and have fallen in love with this podcast, bought SpinRite, et cetera, et cetera, blah blah blah, as Leo would say.

I was listening to Listener Feedback #153 and the discussion about OAuth. The problem you were having was how do we ensure the users can authenticate Facebook's page so they know they are entering their credentials into the right site and not a spoofed OAuth Facebook portal. Well, I am an IT Auditor in the financial institution sector and instantly knew of an answer that is a natural progression from the idea that we use pictures of our Facebook friends: a predetermined security picture and security phrase.

Almost every financial institution I audit now has this simple security feature installed as part of the initial online banking setup phase, where the user selects their image and writes up their own unique phrase to go along with it. In the case of OAuth, the real Facebook page would be able to display the image you selected when you set up your security settings in Facebook, since you would have navigated to Facebook yourself to set up the images before Facebook would allow OAuth to work. But the important part is that the bad guys wouldn't know what the security image or phrase you chose was and wouldn't be able to display it to you.

The only problem I can see is that it might be possible for the bad guys to scrape the image or phrase off of Facebook's actual login page, if they know the email you used for Facebook, and then be able to display it to you as part of a spoofed site. But I would imagine that attack might be more complicated and only be part of a spear phishing attack. Can you think of a solution to that point, since such a vulnerability would exist for financial institutions as well, and the only protection being a somewhat hard to guess username.

Steve: So, okay. I promise this is the last time we're going to talk about this. But it still confuses people. So I thought it was worth saying it once again. I'm sure we could sit down and logically demonstrate a proof for the fact that this is a Catch-22. If you go to

an OAuth login site like Facebook, and you do something to identify yourself, and then they show you something to attempt to do mutual authentication - that's what we're talking about is mutual authentication. It's not just we want to authenticate with them, we need them to authenticate with us. And that's why, as Blake has suggested, we ask them to provide us something that only they know.

The problem is, and this has already been done, so this is not theoretical, a theoretical attack, it's exactly the same as if we are bounced to a spoofed site that looks exactly like Facebook's, who are you claiming to be? We provide that information on the web page and submit it. We think we're submitting it to Facebook. We're submitting it to the spoofed server. The spoofed server immediately goes to Facebook, provides that information, gets the reply page, and that's what it sends us in reply to our submission. So it has inserted itself as a man in the middle, in a classic man-in-the-middle attack. And we see the information we're expecting from Facebook.

So what's actually happened is this is worse than if we weren't expecting mutual authentication because now we have affirmative proof, we think, that this really is Facebook. But in fact it's still Facebonk, and Facebonk just queried Facebook behind the scenes to provide us with what Facebook would have provided us directly. So now we're really sure that we're in the right place, and we provide the balance of our authentication information, and it's stolen by Facebonk as part of this OAuth spoofed login.

TOM: And there's no way around that because...

Steve: There is not.

TOM: ...whatever you would give Facebook to get that image, Facebonk can take, too. You can come up with a million different scenarios, but they all have that particular weakness; right?

Steve: Yes. And that's what I meant when I said I'm sure we could rigorously demonstrate that there is no solution to this because of the possibility of a man in the middle, or in this case a website in the middle, that is grabbing our submission, then on the fly getting it from the real Facebook and then providing us what Facebook sent to it. And so then it receives the second phase of our authentication. There just isn't a way around this.

TOM: Well, so in other words, we're still stuck with the usual problem of security, which is you have to teach people to be better at security. Which works for some, but not for everyone.

Steve: Well, and we keep coming back to this. It is the tension that exists between convenience and security. Is it convenient to use a third-party site to authenticate? Yes. Oh, my goodness. I mean, I see that option increasingly. It's like, oh, here, rather than filling out these forms and telling us all about who you are, no one wants to do that, and especially for, like, you just want to make a posting on a blog somewhere.

TOM: No, that's why LastPass and 1Password and all those companies have a business; right?

Steve: Yeah. Yeah.

TOM: By the way, Facebonk.com is available for sale.

Steve: [Laughing]

TOM: That's what I was doing just then when you saw me, when you caught me looking at my laptop. Well, that's it. That's the last of our questions. Steve, this has been really fun, as usual. And I'm so glad I get to do this a couple more times, selfishly, before Leo comes back from Australia.

Steve: Yup. Looking forward to it next week. I'm not sure what we'll talk about, but I'm sure I'll have an interesting main topic for us, and whatever interesting news has happened in the meantime.

TOM: Well, I cannot wait to find out. Of course, folks, don't forget, if you haven't figured it out already, GRC.com for SpinRite, for ShieldsUP!, for all kinds of cool projects. Is there anything new going on over there? What's the most recent thing that you've put up, because you've always got something cool cooking.

Steve: Yeah, I've got a few things in the works. So I think I'll just keep them close to the vest for the moment.

TOM: All right. Well, I'm looking forward to finding out what those are. Show notes are always at TWiT.tv/sn. That's it for Security Now!, folks. Stay secure. We'll see you next time.