Disconnect WidgetJacking

Description: After catching up with a very busy week of interesting security news and events, Steve and Leo examine the growing privacy and security problems created by the ever more pervasive social widgets - Facebook's LIKE button, Google's +1, Twitter's Tweet!, and others - and they offer an easy-to-use free solution!

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-386.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-386-lq.mp3

SHOW TEASE: It's time for Security Now!. Did you know, did you know that those Facebook Like widgets, the other social share widgets, are actually revealing your identity to anybody who's on the same network? This is a problem, but Steve explains how to fix it in a very simple explanation, coming up, along with all the security news, next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 386, recorded January 9th, 2013: Disconnect WidgetJacking.

It's time for Security Now!, the show that does its best to protect you online. It's a never-ending struggle. Fortunately the Explainer in Chief is here, Steve Gibson, the man behind Security Now!. I met so many people at the New Media Expo and at CES, Steve, who said, "Tell Steve hi. We listen religiously." And it's usually the high-end geeks. They want to know more. You can never get too much security or too much geekness or too much good information. And they love how geeky we get on this show.

Steve Gibson: Well, we've got more today. There's a whole bunch of interesting stuff that happened this week we will catch everyone up on. And something happened in November with an interesting project that I've had my eye on which achieved critical mass. And so I'm going to discuss it later, but first talk about the problem that has been growing known as WidgetJacking...

Leo: Uh-oh.

Steve: Which people have not been talking about. This is essentially, it's related to the so-called "sidejacking," which is what Firesheep was doing. Then this involves leveraging

the lack of security of social widgets. There's a privacy aspect, but there's a serious security aspect to it. And the good news is - so anyway, we're going to explore it, and explain it, and everyone will understand it. And the good news is there's a solution for it.

Leo: Excellent.

Steve: Yeah. But before we go, you need to type that first URL into your machine.

Leo: Okay. You don't want to say it out loud because obviously we're afraid it will bring the site down, as you are wont to do.

Steve: Correct, although I tweeted it last night, and I've never had so many responses I think to anything. It is beyond cool. Now watch it. And it took me a while to get what it was doing. I created a memorizable bit.ly shortcut, bit.ly/factorizer. So again, that's http://bit.ly/factorizer. It is just - it is spectacular.

Leo: So it just looks like things are doubling here.

Steve: Well, no. Look in the upper left. It's incrementing.

Leo: Oh, okay, it's incrementing. Okay.

Steve: So it's incrementing one by one.

Leo: It's dots on a dice, kind of, sort of.

Steve: Well, but keep watching because you'll begin to sense what it's doing. Primes that cannot be factored show as...

Leo: Oh, as a circle, I see.

Steve: ...as a circle.

Leo: I see.

Steve: Because there's no way to subdivide them.

Leo: So what you're doing is you're going through the numbers one by one of dots and factoring them into their factors - fours, twos, threes.

Steve: Yes. And also then threes of fours of sevens, and sevens of, I mean, so all of the - it basically does a complete factorization of each number and animates it. And I have...

Leo: I'm going to speed it up because I've got it on slow play. There's a fast-forward that's a little bit faster. Maybe we'll just keep that running for a while. How far does it go?

Steve: Somebody said it goes to 10,000.

Leo: It's very cool.

Steve: It's just mesmerizing. I thought it was, you know, of course it applies to what we're doing, too.

Leo: And I do like it that it's HTML5. And I'm sure you like that.

Steve: Yes. It's code running in the browser. And, in fact, Firefox 18, which was released yesterday and which we'll be talking about, runs at 25 percent faster.

Leo: Oh, well, I'll have to download it. We have to give that a try. How funny. This is the fastest I can do it on Safari. Let's see how fast Firefox will do it. All right. How fun.

Steve: Yeah, it's beautiful. So the big security goof is one of those that is really painful because it is incredibly widespread, incredibly old, that is to say, six years old, which is in this timescale it's infinitely old. A six-year-old flaw was just found in all versions of Ruby on Rails from version 2.0 on.

Leo: Oh, no.

Steve: So, and this is bad. A security-focused Rails contributor, Aaron Patterson, posted in a Google Groups thread: "The parameter parsing code of Ruby on Rails allows applications to automatically cast values from strings to certain data types. Unfortunately, the type casting code supported certain conversions which were not suitable for performing on user-provided data, including creating Symbols and parsing YAML" - that's Y-A-M-L, and that's one of those recursive acronyms, "YAML Ain't Markup Language." Anyway, continuing, he says, "These unsuitable conversions can be used by an attacker to compromise a Rails application."

Leo: So, as always, sanitize your inputs, baby.

Steve: Yeah. Well, and the problem is this. Because this use of user-provided data wasn't expected, you were probably not sanitizing for it because you wouldn't think there

was any problem. He said: "XML Parsing Flaw, which was first introduced in version 2.0 six years ago" - that is, the GitHub commit for that is six years ago - "allows an attacker to bypass authentication systems, inject arbitrary SQL code, inject and execute arbitrary code, or perform a DoS attack on a Rails application."

So in the RubyonRails.org weblog for 1/8/2013, which was yesterday, the updates were announced. The guy posted, "Hi, everybody. I'd like to announce that" - and then the current version numbers like 3.2.11, 3.1.10, 3.0.19, and 2.3.15. So those are the latest versions of the various sub-versions - 3.2, 3.1, 3.0, and 2.3 - have been released. "These releases contain two extremely critical security fixes, so please update IMMEDIATELY," he had in all caps. And Patterson suggested, if for whatever reason, for any reason you cannot update, disable XML parsing completely or remove support within the parser for Symbols and YAML because those are the two weaknesses that he found.

So I wanted to immediately let, I mean, presumably anybody, hopefully, who's maintaining Rails-based systems will be on a mailing list and will already know this. But this is fresh. And of course the hackers, the malicious guys are going to be on this fast because, I mean, so many sites, as we've discussed many times in the past, are now running on Ruby on Rails.

Leo: Yeah. I love Ruby on Rails. And Ruby is wonderful. So it's Ruby, though, not Ruby on Rails that has the problem.

Steve: Well, he's a Rails contributor. So I don't know where you divide this.

Leo: Oh, well, Rails is a framework. Ruby is the language.

Steve: He's calling it a Rails application, yeah.

Leo: Okay. So it's a Rails issue, then. That's actually better than if it were a Ruby issue because Ruby's a...

Steve: Oh, it's not language intrinsic.

Leo: Got it. It's in the framework.

Steve: It's packages on top, yeah.

Leo: And we've seen, actually we've seen problems with the framework before. So that's okay. Okay, good.

Steve: So also in the news...

Leo: By the way, let's just check in real quickly. We're up to 1,305. That's Safari. This is Firefox. Catching up. It is appreciably faster.

Steve: Oh, and you got Firefox 18?

Leo: Yeah, this is 18. So Safari, Firefox. Yeah. I would say, I mean, 25 percent is not as huge as it might sound.

Steve: No, it's not two times or anything. There was a bunch of news also, actually this was late, this was just after our podcast last week that Chrome detected a phony Google certificate in the wild. And that really upset everybody because here we're back to a trusted, in the root, certificate authority. This was Turktrust, a Turkish CA, which when they started doing the research - this was not malicious. This was not a break-in or a compromise of their system. But it's also a little disturbing because two years ago they inadvertently issued one of their customers a pair of intermediate certificates rather than an end SSL cert.

So what that means is, of course, an endpoint SSL cert is signed by the CA, and all you can do is assert your own identity with it. If you ever get an intermediate certificate authority, you are a certificate authority that has been signed by the root certificate authority. So for two years this entity that inadvertently received these two intermediate CAs - and of course we're having to take all this on trust, that this wasn't some secret dark entity of the Turkish government that got an intermediate CA from Turktrust and so forth. So if we take this on face, what we do know is that for two years there has been an intermediate CA that could issue trusted certificates for any domain it chose.

Leo: That's wild.

Steve: I know.

Leo: Two years.

Steve: Yeah. So this is the, as we've discussed, the Achilles heel of our entire SSL trust system. Somewhere in any system there has to be an anchor of trust. And so that's where you're going to be weakest. And our long-time listeners will probably well remember that podcast [SN-104]. It was maybe, what, five or six years ago, where I had happened to check in, to look at the block of trusted CAs in whatever browser it was, I don't remember, because I'm old school. And once upon a time there was five. There was VeriSign and Equifax and, I mean, it was like a handful. I looked at it, and it was 800. And of course that's where the famous Hong Kong Post Office came from. It was like, wait a minute, what are they doing signing certificates that I trust?

But anyway, so immediately Google updated Chrome to remove trust from that one certificate that was known to be a problem, immediately informed Turktrust that something was wrong, and please find out what, and informed the other browser vendors that there was a malicious intermediate CA in the wild. And so now - we're okay now. That intermediate CA and anything it might have ever signed that has also been trusted

but never warranted our trust, they're dead now. So none of our browsers will use them, and this happened last week. But that's the story behind that, for people who sent me notes and tweets saying, uh-oh, what does this mean? So we're okay. But again it's like, oops, a little glitch in the system.

Also, yesterday was Microsoft's Patch Tuesday. And it's another important set of patches. Unfortunately, and we'll remember that last week we wondered whether Microsoft would have time to fix the zero-day flaw which had just been, I think it was the prior weekend, found in the wild. Remember it was being used for targeted attacks on IE6. But it's also known to affect 7 and 8, not 9 and 10. So if you need to use IE first - because it's funny, whenever I tweet anything about IE, I get back the predictable, well, who's using IE? It's like, yeah, I know, I know. But I also get lots of people saying, hey, I have no choice. My company makes me, my bank makes me, my whatever it is makes me. So there are - and, I mean, I have it because - for Windows Update, and there are some things where it still has to be around. And it's around by default, of course, in a Windows environment. So it was not fixed yesterday.

So, and Brian Krebs reported that it has, as we also predicted last week, now been added to the Metasploit Framework. So it is trivial for script kiddie level malicious hackers to go starting to exploit this. So we can expect to see its use expanding in the wild because now the hackers think, well, we may have a month before the next Patch Tuesday. Of course if it goes really crazy, Microsoft may be induced to issue an out-of-cycle patch.

Now, Part 2 of this is that, almost immediately after Microsoft's Fixit tool was put out, and we talked about it last week, it was discovered that there was a way around it. So even Microsoft's Fixit tool is only partially effective against this exploit. So really the only advice I have is, more than you might have already been avoiding the use of IE 6, 7, and 8, if 6, 7, and 8 are what you're using, try to minimize your use of it. Or I would say go to really trustworthy sites.

The problem is it's not the sites that are going to be malicious. But in the attacks we've seen, remember we were describing them as "watering hole sites" because the way these attacks were working is that not secure, not sufficiently secure sites were being modified to attack their visitors. And so it was a sort of a - the idea was being that the actual targets of the attack, the companies that the attackers wanted to get into, would go to this so-called watering hole site, get themselves infected there, give the attackers the access they want, and be compromised as a result.

So we don't have any fix for this, not one that apparently works. And having been dropped into the Metasploit Framework means that it's completely understood. The hackers know how to exploit it. It's now in the open source mode where anyone can get at it, and we have no fix for it. So it's about as bad as it gets. And it's, of course, being actively exploited in the wild.

Aside from that unhappy news, Microsoft this Tuesday issued seven patches, two of which were critical and addressed, like, 12 or 13 security problems. One of those only affects Windows 7 and Server 2008 Release 2 only. So rather narrow. But the other one is another XML Core Services problem. We've seen those before. This affects everything Microsoft has ever touched, even something I've never heard of. I was looking down through the list of stuff. And I thought, what is the Groove Server 2007? Does anyone even know?

Leo: That sounds like a Microsoft code name.

Steve: No, it's the official name. It's, like, in their list.

Leo: No, it would be, yeah. But, I mean, it would be what they called it. Maybe not what...

Steve: Yeah. Oh, I see, and they decided not to change the name when they put it out.

Leo: Microsoft, doesn't it sound like Microsoft Groove? Groove products, let me see, I don't see anything. What is Microsoft Groove?

Steve: Groove Server 2007.

Leo: Yes. It's something Ray Ozzie did. It's part of Office. And a number of our people in the chatroom use it.

Steve: Well, folks, update your XML Core Services immediately.

Leo: Microsoft Office Groove 2007 creates dynamic workspaces to hold all digital information related to any task. So it's like, you know, Ray Ozzie did it. It's a collaboration space.

Steve: Just drop the data in the groove, I guess.

Leo: Yeah, yeah.

Steve: They already used Suitcase. They can't use that again.

Leo: It does sound like a Microsoft name. An old Microsoft name.

Steve: Oh, god. And the other thing was an important update to the .NET framework and a few other things. So anyway, update your Windows, and don't use IE.

Leo: They call it - I think they just call it SharePoint now.

Steve: Oh, okay.

Leo: Well, it was part of SharePoint. I went to the Groove page, and it brought me to the SharePoint page, so I don't...

Steve: Because SharePoint is strictly cloud-based; right?

Leo: Yes, yes. As would Groove be.

Steve: Oh, I see. You went to Groove, and it bounced you to SharePoint.

Leo: Yeah.

Steve: Ah, okay. But back in 2007 we still called it Groove.

Leo: We had Groove, and it was groovy.

Steve: So speaking of same old, same old, we have Adobe, who simultaneously issued security fixes for all of their things also - Acrobat Reader and the Flash Player plugin. And in my notes here I just put "blah blah blah." Which is to say, just go update yourself.

Leo: Ditto. Ditto ditto ditto.

Steve: Yeah. And big news. Yahoo! Mail finally gets HTTPS.

Leo: Oh, finally.

Steve: Yes. And they've continued to have break-in problems. I've seen some notes about people getting spam from Yahoo! Mail people, and I got some myself yesterday.

Leo: Yeah, it's notorious, yeah.

Steve: Yeah, from like a week ago, I mean, from years ago, somebody who had an old email address of mine that I kind of monitor sort of for this purpose. And it's like, oh, look at that, something coming in here. Oh. Anyway, so what that means is that if you sign into Yahoo! Mail with your ID and password, hover over the Settings icon, and from the dropdown menu which you'll get select Mail Options. Scroll down to the bottom of that page. And then, under Advanced Settings, select the checkbox opposite to Turn on SSL. Then a dialogue will be shown, and a refresh is required to change the setting. Click Okay, and then click on the Save button.

So they're a long way away from on by default, but at least they do allow you to maintain persistent security. And, boy, in this day and age, it's amazing that this is so late in coming, when we know now that, if you only are secure during logon, and then you're in Starbucks or any other open WiFi hotspot, an airport or anything, then all of your non-secure transactions, which is everything subsequent, will be in the clear. Which means the cookie which you were given in order to establish your session at logon is there, and you can be hijacked.

So it's not surprising Yahoo! is having these problems. This is how hijacking happens. Exactly like this. So lord knows why it took them so long to make it happen. And they really need, unfortunately, it being buried like that, who's going to find it? So our listeners will. And if you get email from anybody who's got Yahoo!, listeners, drop them a note and say go find out - go to Advanced Settings and turn on SSL. It's not going to break anything anywhere because we know how, we can all do SSL now.

Leo: I remember when you discovered Firesheep [SN-272], which was a hack that allowed you to do this kind of thing at an open WiFi access spot. And you celebrated it because, as you said, this will force everybody - how many years ago was this, two or three years? - this will force everybody to do HTTPS all the time. And it's taken this long.

Steve: Yeah. The responsible people did.

Leo: Facebook and Gmail.

Steve: Yes, exactly.

Leo: Well, now Yahoo!. Join the club.

Steve: So Firefox 18. No huge security or functionality changes. The IronMonkey - don't use that as your password, folks. The new IronMonkey is their JIT, their Just In Time JavaScript compiler, which delivers 25 percent increased performance over the already very fast JavaScript technology they had before. IronMonkey is able to optimize more than their prior compiler technology because it compiles the JavaScript into an intermediate language that is designed for machine-driven optimization, that then applies lots of machine-driven optimization strategies to this intermediate language. Then, and only then, it translates that into assembly language in order to execute at maximum speed on the system. So it's a two-stage process with that intermediate step in the sandwich really being responsible for very clever optimizations. And as a consequence, I mean, I'm excited because we're really seeing JavaScript, I mean, you know what a mixed blessing I feel it is because it's so prone to abuse, thus the reason I run with it disabled all the time except for sites I trust. But, boy.

Leo: It's powerful.

Steve: I mean, this is - it's really clear that the browser is the platform, the application platform of the future. Speaking of which, one of the things also newly supported at the preliminary level is something called WebRTC. This is in Firefox 18. RTC stands for Real Time Communications. The WebRTC is a forthcoming HTML5 generation W3C and IETF standard to support real-time Internet communications: phone calls, video chatting, sharing, peer-to-peer file sharing. So again, here's another example of, due to the standards moving forward and the power that we're developing in our browsers, we're talking about moving what is currently standalone apps or plugins into the HTML, into the web standard. So our browsers won't need Skype to be downloaded and installed, or

Google Talk, or anything. They'll have it natively, in the same way, for example, that they now can play video without needing a Flash plugin or anything else in order to play video because the browsers are able to do that on the fly. And speaking of which, Google has donated their VP8 video codec to the WebRTC effort. This is the one that they got from...

Leo: On2.

Steve: ...On2, that's right, and are claiming that it is unique in that it is license free. There are some people who aren't quite sure that that's the case.

Leo: Yeah, including the H.264 Consortium, who doesn't want it to be license free.

Steve: Exactly. And in fact there is pressure for H.264 support in this same standard, in this WebRTC standard. But the resistance, of course, from people like the Mozilla Foundation, is they don't want to put license-encumbered technologies in the browser. They're just fundamentally against that. So I think we'll probably always be in a position where there are a couple different video standards, in the same way that we have JPG, PNG, and GIF image files. I mean, those are established. They're not going to go away. So there isn't just one way to show a picture. There's a few ways. And there won't be one way to play a video, there'll be a couple, depending upon what platform you're on.

Also with Firefox 18 we have full retina display support on the new Macs with the retina screen, which prior Firefoxes did not support. So that's here for, like, super-crisp retina font-rendering and so forth. And over on the Android side, Firefox 18 adds support for on-the-fly search suggestions in their so-called Awesome Bar. And those search suggestions are transacted over secure channel, even if you're not, so that no one can see what's going on. And there is a malicious site warning system built in for Android, which is certainly handy to have.

And I promised that I would take a look a couple weeks ago at what "Extended Security" meant for Universal Plug & Play. Remember that I saw, I think it was a couple tweets, said, hey, well, what about the extended security? Well, it's completely useless.

Leo: Oh, dear.

Steve: First of all, it's not widely available. I could only see really that Thompson was using it in some of their routers. And all it does is lock down some ridiculously wrong things that Universal Plug & Play should have never been allowed to do in the first place. But it provides virtually no security. So no malware would be at all slowed down if you had extended security turned on. And I didn't even write down and bother enumerating it because I just thought I would tell everyone, eh, you know, don't...

Leo: Or don't worry about it.

Steve: Yeah, don't worry, it's not going to help.

Leo: Plenty of holes left.

Steve: And apparently it even does mess things up.

Leo: Oh, no.

Steve: I mean, despite doing nothing for you, I ran across a lot of advice saying, oh, if you have that turned on, turn that off because that's the problem. So, you know. Now, also in the "I'm not going to go into it in detail," I wanted to just say to Jungle Disk users, maybe there's hope, because there have been so many unhappy people with Jungle Disk. They got bought by Rackspace, which is why Rackspace is the cloud service of choice for Jungle Disk, although it still works with Amazon and so forth. But the support's been really lousy. They recently changed their philosophy where it's no longer essentially free, the way it used to be, which really upset people. People used to use it, for example, to create a private network to an unattended server, for example, so they would be able to get to that server.

Well, Jungle Disk, or rather Rackspace as parent, in the quest for more revenue, decided that they would only allow the nonpaid versions to interoperate between logged-on people. So you can use it, for example, as it is used often, to set up a private little gaming network, in which case everybody would be logged onto their machines that are connected into this little Jungle Disk subnet across the Internet. But you can no longer use it in its free version, the way you have always been able to use it, to hook to an unattended machine where you don't have somebody actively logged on and using the system.

Now, the ray of hope is that, at blog.jungledisk.com, for any holdout users of Jungle Disk, they had a "What the new year holds for Jungle Disk" entry. There's a new CTO. And apparently they sent out questionnaires to some subset of their users, and the questionnaires came back polarized. There were people who were completely happy and wanted nothing more. And on the other side there were people who probably feel about Jungle Disk the way we feel about PayPal. I mean, they're only using it because they have to, because there's absolutely nothing else that they've found that does what they want. So this was apparently a bit of a wakeup call, and they're promising that they're going to fix these problems. They're going to be better with support, they're going to communicate more, and they're going to fix things. So for what it's worth, people who are using Jungle Disk, it might get better. And anyone who's interested can check out blog.jungledisk.com for the details.

And in my backlog of stuff to get to is a rather sobering analysis that an R&D company did of the silicon of a chip used in networking products where they discovered a hardware backdoor in the design. We've talked about this. We've touched on it a few times. It is a worry. Because there's obviously tension between, for example, the U.S. and China of various sorts. And we're getting a lot of fabrication being done there. And how do you know what's in the chip? The chip's got legs, but it's got a lid on it, and it's incredibly complicated. And you just can't look at it and know what the design is. It takes a huge amount of effort to reverse engineer the design of a chip from looking at it.

Now, in the old days, microprocessors were reverse-engineered. "Popping the lid" was something that was jargon in the industry, and it was the way designs got stolen. But with this insane ramping up of complexity that we've had, it's just - that's vastly more

difficult. Yet the Los Alamos National Laboratory here in the United States just removed all of its Chinese network switches. A Congressional Report found that Huawei, H-u-a-w-e-i...

Leo: I think it's Huawei. I think you pronounce it Huawei. By the way, they had a massive booth at CES.

Steve: Were they waving American flags?

Leo: Well, they dispute this. Yeah, this is a while ago that the Commerce Department came out with this. And it's reasonable. But Huawei disputes it.

Steve: Yes. So what the Congressional Report said was that the company had ties to the Chinese military and intelligence services. They deny any connection to the military and say that their products are completely safe to use. So who knows. I will, as soon as I get to it, take us through the details of reverse-engineering the hardware, which did find a backdoor, so they are known to exist. We don't know that this is an instance of it. But it is something that we need to keep in mind.

Leo: Yeah. And if I were running Los Alamos Labs, where they make atomic weapons, I might be prudent. I might act in terms of prudence. It's the router that they're worried about, right, because who knows what code's in there. Not just the chips.

Steve: Yeah. Or they said switches. And "switches" is sort of a generic term.

Leo: They make phones, too, and I don't know, I mean, this was a pretty broad report.

Steve: Yeah.

Leo: Anyway...

Steve: So I did ask for the tweeter who tweeted me a week ago or two weeks ago that he had come up with a cool synopsis page of my Twitter stream related to Security Now! podcasts. So that when I say, oh, I just tweeted this link, for anyone who's listening, the observation was made in our last Q&A, it was like, well, Steve, that might have been a year ago, that I'm listening to Episode 206. So that's a problem. Anyway, so I created a shortcut. It's bit.ly/sggrc, all lowercase, because bit.ly is deliberately case-sensitive so that it's able to encode more links in a short string. So it's bit.ly/sggrc. And it was Simon Paarlberg in Copenhagen who...

Leo: It's amazing.

Steve: Yeah, isn't that great?

Leo: Thank god for Scandinavian winters, that's all I can say [apps.simonpaarlberg.com/x/sn_twitter.html].

Steve: He called it a "hack," so I think he must have a bot which is just pulling the stream in. I did see that it's always up to date. So it has my latest tweets from yesterday in it at the top. It is in reverse chronological order. And so what I think he does is he, like, inserts a place marker for every Security Now! podcast and nicely formats it. So anybody who is listening to a podcast, and I mention a tweet that has a link, you can go to bit.ly/sggrc, which I came up with that because that's the same as my Twitter handle, and scroll back and find it.

Leo: Boy, if this is code, this is impressive.

Steve: That's very cool, Simon. Thank you.

Leo: It can't be code. This is somebody - unless he's...

Steve: No, it's code because...

Leo: How's he getting the time codes?

Steve: I'm sure that's all in the Twitter stream.

Leo: Oh, I see. I see what he's doing. It's not the timecode into the podcast. This is the tweet.

Steve: Right, right, right.

Leo: I get it. So he's just saying - oh, yeah. I can see how he'd do this. So he'd say, well, the week of January 1st through 8th, these are the tweets.

Steve: Exactly.

Leo: Yeah, that had links on them.

Steve: And by the way, Elaine also replied, listening to the podcast as she is forced to do every week, that she has been putting - she's been expanding and putting the links in the show notes all along in her transcripts. So they're also in the transcripts, for anyone who is looking.

And you're going to like this one, Leo. Go to that next link before I mention it, although I also tweeted it, so you can find it right there in my Twitter stream. Mark Thompson turned me on to this, and I thought it was really interesting. If you dig down, you can see the chart of - okay. I should, for people who aren't seeing the feed right there - but click on some of those languages, Leo, like click on Java, No. 2.

Leo: So this is from TIOBE.com.

[www.tiobe.com/index.php/content/paperinfo/tpci/index.html]

Steve: Yes.

Leo: And it is what? It's a programming community index.

Steve: Yes.

Leo: Yes, so Java.

Steve: And this in particular, this...

Leo: Oh, it's how much it's used.

Steve: Well, it's languages. It's the community's overall use of languages over time.

Leo: So Java is declining.

Steve: And with some weird anomalies. Like, look at that big notch.

Leo: It stopped in 2004/5. People really stopped - that has to be an artifact.

Steve: I think that's a reporting anomaly.

Leo: That has to be an artifact, yeah.

Steve: Yeah. But it is on a straight-line decline. Now scroll to the chart below. And one of the really interesting things is Objective C, that very blue line at the bottom that takes off...

Leo: Ooh, look at that take off. iOS, baby.

Steve: Exactly why. Exactly why.

Leo: That's what iPhone apps are written in. And starting in 2009 it really [acceleration sound].

Steve: Yup. But No. 1, still top dog, is good old C. It's sort of the high-level assembly language.

Leo: Plain C. Not Objective C. Not C++. Not C#.

Steve: Yes, Plain C.

Leo: Kernighan and Ritchie C.

Steve: Yup, exactly.

Leo: That's neat.

Steve: "The C Programming Language," one of my favorite texts.

Leo: I love C. It is one of the best programming books ever written. It's about this thick, like a quarter of an inch. And it's beautiful. It's elegant. First high-level language I ever learned, and I'm still...

Steve: It's a long URL, for anyone interested. I did tweet it, and a lot of people fed back that they were liking poking around in those charts. So if you just check my Twitter stream, Twitter.com/SGgrc, it's not far back at this moment. Or bit.ly/sggrc you can find the index, thanks to Simon, of everything. And anyway, it's called the TIOBE Index.

Leo: They say they use search engines. Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube, and Baidu are used to calculate the ratings. They're counting lines of code. Lines of code. Is that amazing or what?

Steve: Wow.

Leo: And they make the point that you should probably, if you're a programmer, be paying attention to this because these are skills that you'd want. Ruby, by the way,

is No. 11 on this, right at the bottom on the Top 10.

Steve: Well, but go to the second chart. Way down is the ones that didn't make it onto that chart. There's the real dogs. I mean, Logo is still there.

Leo: PL/SQL, Transact, SAS, COBOL, Fortran, R; Scheme, which is a great language, and I wish it were more used; ABAP, which must be some specialty language; Logo, Prolog, Erlang, Haskell, Scala. These are just too hard for most people. Q, D, RPG...

Steve: Or they're just weird. They're, like, declarative languages or, like, really bizarre...

Leo: APL requires a special keyboard.

Steve: Can't even type APL.

Leo: Common Lisp, ActionScript, way down there. Awk, F. But these are, yeah, some of these are special. I mean, Awk and Tcl, you know...

Steve: ActionScript is still there, huh? Wow.

Leo: Yeah, yeah. No. 43 on the Top 50. And then, by the way, the next 50: FoxPro, Algol...

Steve: I didn't see that.

Leo: Oh, yeah, there's the bottom 50, 51 to 100.

Steve: Oh, my god.

Leo: PowerShell, SPARK, VBScript, WebDNA, xBase, X10. Some of these, for instance VBScript, probably you're not going to show up on a Google search because it's all internal code. And so that's probably misreported, heavily misreported. But who knows. LabVIEW's in there. Emac. Mathematica.

Steve: So Ars Technica did a nice explanation of what happened at the beginning of the year with iOS 6's Do Not Disturb bug.

[arstechnica.com/apple/2013/01/ask-ars-why-will-apples-do-not-disturb-bug-fixitself-next-week]

Leo: Yeah, wasn't that great? I want you to explain this because I don't know what the difference is between four capital Y's and four lowercase Y's.

Steve: Yeah, and I just want to...

Leo: Would you 'splain that to me, Lucy? 'Cause that's what happened.

Steve: If you really care, I could tell you next week. Because I just sort of looked at it, and I just shrugged. It's like, eh, okay.

Leo: Well, the short form is that they used an ISO numbering scheme which has some known bugs in it. There's a Gregorian numbering scheme, and they used the ISO scheme without testing for bugs, the bugs that are known, well known. And everybody else does. Most people use the ISO scheme. But people know that, oh, well, yeah, in the first week of the year you've got a problem. Week 1, you've got a problem. Or Week 0, maybe it's Week 0.

Steve: Wow.

Leo: Yeah. It's really - it's very interesting.

Steve: Yup, good old boundary conditions will get you every time.

Leo: Boundary conditions, baby. You guessed it, really. I think you picked it up last week.

Steve: Well, that's essentially what...

Leo: It always is, yeah.

Steve: Yeah, I do all my math - because I've got all kinds of date stuff on my server, and I, of course, wrote it all in assembler. I have really careful linearization of calendar dates from 1900, I think it is. And then I just base everything on a simple linear number. And it's like, okay, pretty easy. So you divide by seven or do mod seven, and that tells you what day of the week it is. So not that difficult, folks. But I guess if you're using canned packages and not doing it with taking the responsibility that you really have to when you use a canned package or a standard with bugs, then that's what you're going to get.

A little quick update on the TrebleShooter dog training project. We've got lots of people building them, Leo. I'm getting where they're posting pictures. We have a topic in the Google group, the Portable Sound Blaster Google group, where people are posting the pictures of their construction projects, building sort of flashlight form factor devices. We

have two designs. One is the one I will end up making. It's a fixed-frequency, incredibly low parts count, no microcode or firmware or anything required, that costs about $10 and just turned out beautifully. I never expected we were going to get something like this when I began. But I also want to explore a microprocessor-based version, just to sort of - just to have it before I switch my attention from this to something else. So I'm still working on that.

Many people are saying, hey, where are the designs? Where is it? I want to build one. And also I'm getting tweets saying that the GRC pages are blank except for the introduction page. And it's like, yes. This is still very much in flux. So I would recommend that it's probably easiest if people just wait. The Portable Sound Blaster group in Google is very active. But those groups are so awful that it's just impossible to find anything there. Do you know that Google can't even search them, Leo? It's unbelievable. They're hosted by Google, and you can only search the subject line. You can't search the content of the posts. And it's like, okay, crazy.

So if anyone is interested, there's a ZIP file, a ZIP archive that I do maintain at GRC, only because you can't maintain such things easily in the Google groups. And that's GRC.com/tqc, as in The Quiet Canine. So GRC.com/tqc/latestdesigns.zip. And I'm updating that as I go. And that's got a collection of resources for the project, the schematics and PDFs and even a big PNG file of the source code for people who just kind of are curious and want to scroll down through that to see what it looks like.

Leo: You're using Google Communities now. You're not still on...

Steve: No. Turns out - no, no, Communities do even less, Leo.

Leo: I don't know why they wouldn't search into the Google groups.

Steve: Isn't that amazing? Yeah. Many people were posting - the reason I did this, the static file at GRC, is that people were saying, hey, I can't find the version 2.2.2 design. And it's like, I'm referring to it all over the place. But sure enough, when you try to search, it doesn't show it to you.

Leo: They do say that - you're using the new Google Groups; right?

Steve: Oh, I am using the new Google Groups. And I did go look, I did go check out the whatever it is, the other thing, the new thing. But it's even less.

Leo: Communities, yeah, this new G+ Communities.

Steve: Communities. It's even less than this is. So anyway, what - well, anyway, so I won't go into any more details. But we're having a lot of fun. It's coming along. And I will have some results pretty soon.

Leo: I'm going to mention something before we go on that I want everybody to know, is if you watch this show on YouTube - I know not many do, but some do. Most people want to download, and they what a transcript and so forth. But we do make all of our shows available on YouTube. And as of this week we've switched everything over on YouTube. So we have a whole new - I'm very proud of what happened. Our engineering staff and Russell Tammany and Spiro and everybody got together, and we've written code to automate the encoding of audio and video. So editors finish, they drag a mezzanine file, and all this stuff happens automatically. It gets posted automatically. Which is why there've been glitches from time to time.

But now that that code is - and by the way, that means 720p versions are also available for all shows for download now. But also it means that we've moved the YouTube channels. Previously we had a single YouTube channel, YouTube.com/twit, for all shows. That really is kind of a nutty way to do it because, if you subscribed, you got every one of the 25 shows we produce. Not really useful. So now what we've done is, if you go to YouTube.com/twit, you'll see on the right there are new TWiT channels for all the shows. Unfortunately you can't get them all on there. But, well, you can. If you push "More," you'll see all the shows. And I don't see Security Now! on here yet. Maybe because the first one will be today. So I think the channel is, and let's just check, twitsecuritynow. We were able to secure channel names for all the shows. Yeah, I think this is going to be where it'll be once the - this episode will be the first on there. So twitsecuritynow will be the channel, youtube.com/twitsecuritynow. We couldn't get plain old securitynow. Somebody else has that. And so, if you are among the few who care about this, just be aware, and we'll put a posting on our YouTube channel and let people know you should now go to a new channel. But that way you can say, I want to subscribe to the YouTube Security Now! feed and just get Security Now!. We needed to do this. You go show by show.

[Note: www.youtube.com/securitynow also works]

Steve: Nice.

Leo: Yeah. To me, my favorite way for you to get the show is to subscribe to the downloads. But I want to make it easy for everybody. All right.

Steve: So this is completely random. And so just, you know, completely random. But for years, people whose opinion I respect have been saying, hey, Steve, are you watching "The Good Wife"? Now...

Leo: That's not the first show you would jump to in TV Guide.

Steve: Precisely. That's the problem. Are you watching "The Bedwetters"? No, I'm not. And I realized, if it were named "Cyclotron Crash Hammer," I would have never missed an episode.

Leo: But is it about a cyclotron clash hammer? Or is it about a good wife?

Steve: Well, I kind of thought, how good can it possibly be if it's called "The Good Wife"? Really, I mean, it's just not going to get my attention. "Justified" just started last night was the next season, which is on FX, is, whoo, for those who watch "Justified," you know what I mean. It's wonderful. But "The Good Wife." And so it was a little slow over the holidays. And Sue, who's been with me for, I don't know, 24 years, every so often she'll say, "Hey, have you ever tried, have you ever watched that?" Anyway, it's fantastic. That's all I wanted to say. Just to share something that I find when I do, like the way I found "Homeland."

Leo: Love "Homeland." And I finished it. But, by the way, it's not universal, the love for "Homeland."

Steve: I know.

Leo: And some people are a little tired of Claire Danes's cry face. And I admit, by the end of Season 2, I was a little tired of it, too. But it's still a great show.

Steve: So "The Good Wife" is now in its fourth year. It's about halfway through its fourth season. Up to the end of the third season it had been nominated for 21 Primetime Emmy awards. Julianna Margulies, who's one of the stars, she of course was one of the - really came to fame as the nurse, one of the head nurses in "E.R.," which Michael Creighton of course produced. The executive producers of "The Good Wife" are the Scotts, Ridley Scott and Tony Scott.

Leo: Okay, that's a good start.

Steve: I know.

Leo: That's encouraging.

Steve: I know.

Leo: Creators of "Alien" and "Aliens."

Steve: Exactly. She got a Primetime Emmy, a Golden Globe, two Screen Actors Guild awards, and a TCA award for individual achievement in this series. One of the other actors got another Primetime Emmy. And a third actor or actress got a Primetime Emmy. So it's won a Peabody Award, twice nominated...

Leo: But, Steve, is it science fiction?

Steve: No. It's a courtroom.

Leo: Oh, it's a courtroom. Okay.

Steve: You wouldn't think so. You have "The Good Wife," like what - I'm not going to get myself into any more trouble.

Leo: Good thinking.

Steve: So why do I like it? First of all, I do enjoy the courtroom interaction. I've always found, obviously, that's a genre that appeals to a slice of the viewership in the country because courtroom dramas have had long-running success over the years. But it's very well written. It's very well acted. All the characters are distinct and interesting. And so I'm not saying it's going to curl your toes. It's not the best thing that I've ever found. But I'm really glad I finally - I got the first three seasons on disk, and I'm going through them. And it took maybe eight episodes. I mean, I liked it right off the bat. But it's like, okay, where is this going? And it's got the classic long cross series story arc running in the background with episodic things happening that develop the characters and move them around like chess pieces on the chessboard. So anyway, if other people have told you about it, and you haven't made the move, now you've heard it yet again from one other person. And I like it. Totally random.

Leo: I have to point out something. And it's one of those things that these things happen so gradually, nobody notices. But we have really changed our TV viewing. You could not have said this five years ago. I mean, maybe you could. You could say get the DVDs, I guess. But I would say, at least everybody in our set, very few people are watching live television anymore. Tune in Thursday night for "The Good Wife" on ABC or something or whatever it is. In fact, I don't even know or care what network it's on. Increasingly, we're watching shows on download, on demand, or on DVD. And, much more importantly, not one episode at a time, but like a box of candy that you just snarf up.

Steve: Exactly the way you absorbed the first two seasons of "Homeland."

Leo: Yeah. And I have to say it's a much better, much better experience. There's no advertising. But also there's not this long...

Steve: Continuity.

Leo: Yeah, continuity is much better. Now, I wonder, I'm very interested if people who create these shows are going to kind of think about that and change, because I think until now you really assumed, well, there's going to be six days between these episodes. And they really spend a long time still on "Previously, on 'The Good Wife.'" I don't know if they do it on...

Steve: Actually, none of that.

Leo: Not on the DVD, maybe.

Steve: And I could not watch "24" because every single hour ended with, like, [gasp], you know, and it's like...

Leo: So much better to watch it in one swell foop.

Steve: Yes. Yes.

Leo: And I suspect that that - now, maybe it's just us because we're nerds, and we have the means, and we have the technology. But I suspect there is a sea change in the way people consumer episodic television.

Steve: Well, I do know that I asked my sister, who's got two - now my niece and nephew, her kids, are in college. But they were in high school. And I said, "What TV do Evan and Jenna watch?" And Nancy said, "Oh, they don't watch TV. They watch everything on their laptop."

Leo: They watch YouTube. Yeah, they watch YouTube. My kids watch YouTube. That's it. Very interesting. It's great. I love it. I love being - because you and I like change. Well, I don't know about you. You're kind of a fuddy-duddy.

Steve: Yeah, I am.

Leo: But most people love technology.

Steve: I'm not denying it.

Leo: You don't embrace the new.

Steve: I've had the same thing for breakfast every single day for the last six months. I'm quite happy with it.

Leo: There were several years where Steve had a ham sandwich every day for lunch, and that was it. But I think those who like change are in hog heaven because the world is changing fast.

Steve: Well, yeah, although there is a problem with distraction because that's one of the things that all these possibilities bring to us is, oh, my goodness, distraction.

Leo: I did some research because I was speaking at the New Media Expo, the podcast show, this week. And we contacted, on Monday I contacted Brian Ellis over at iTunes to say how many podcasts are there? 250,000.

Steve: Ooh.

Leo: Not episodes.

Steve: Series.

Leo: Different series. A quarter of a million, in 140 different countries and 50 languages.

Steve: Do you remember the days when we used to be - we would look on iTunes to see, not where your podcasts were, but how many were simultaneously in the Top 10.

Leo: Those days are gone.

Steve: I think it was Top 15.

Leo: Yeah. Those days is gone. Which is fine. I'm happy. I'm thrilled about it. But you also remember people were kind of going crazy about the idea of a 500-channel cable universe. Quarter of a million channels. Try that. Wow. All right. We shall move on and talk more about security.

Steve: So I have a nice note from Todd Sankey in Vancouver, Canada, who did something. This is less a testimonial than a tip to serious diehard SpinRite fans. He said, "Hi, Steve. With all due respect" - and I'm not quite sure what this reference is, so maybe you can help me out here, Leo. "With all due respect to Jack Black and 'Kung Fu Panda'..."

Leo: Okay. You don't have children.

Steve: "...this combination really is fantastic."

Leo: So "Kung Fu Panda" is a series of movies, animated movies about a panda that is good at kung fu. I think there have been two, maybe three. And Jack Black plays the voice. Okay. There you go.

Steve: Ah. Got it.

Leo: I should [indiscernible] on the air.

Steve: Oh. So he's saying, "With all due respect to Jack Black and 'Kung Fu Panda,' this combination really is fantastic." So apparently he is a fan of Jack Black voiceovers of the panda?

Leo: He loves it, yes.

Steve: Ah. Okay. And so is what he has figured out how to do with SpinRite.

Leo: Okay.

Steve: "I'm a fairly new SpinRite owner, learning about it after listening to your Security Now! podcast, great product, great podcast, thanks. I don't use my cherished copy of SpinRite very often, but I have a small workspace, and my wife 'resists'," he has in quotes, "'resists' letting me clutter it up with spare machines. I just have my one main work machine. So when I have needed to run SpinRite, it's fairly intrusive because I lose access to my main machine." Since of course SpinRite runs, you boot SpinRite, and it is the OS while it's running.

"Did you know it is possible to configure a virtual box, VM, to have raw access to a physical disk drive? Once I learned this, I had to give it a try with SpinRite. The golden goal? SpinRite running in a VM with full access to all its magic and simultaneous full access to all of my main machine, and it just works. It takes a little bit of work because you need to work directly with command line utilities as the functions needed are not exposed in the VirtualBox GUI. But VirtualBox is well documented, and the few steps are really very straightforward. So I write this to you from my main machine as my SpinRite VM grinds away on a reluctant eSATA drive that I need to refresh to get my PVR working again. Please share as you see fit, but I find this combo really useful. Thanks. Todd." And thanks for the tip, Todd. Very cool.

Leo: Yeah, good idea.

Steve: I know that we'll have some listeners who'll go, oh, my god, that's perfect.

So, WidgetJacking. Everyone who's been listening for a while will remember, and you already refreshed everyone's memory, Leo, about Firesheep. Firesheep was an add-on for Firefox which made it embarrassingly simple to hijack people's social networking logons in any open public WiFi unencrypted hotspot. This was only a few years ago, when Facebook was not encrypted, Twitter was not encrypted, LinkedIn was not encrypted, Google was not encrypted, and so forth. And the mechanism of logging on is that you establish your authentication over a secure connection, but then all websites would then drop you, would give your browser a cookie and then switch you back to non-HTTPS, back to a regular HTTP, unencrypted connection.

The problem with that is that, because of the session granularity that browsers have, that is, when a browser makes a request, it's just - it's a request coming into the server

in the sky, like any other request anywhere. And but we want to create a persistent relationship with the server and ourselves as we move around the website, as we do things, whatever it is, after logging on and authenticating ourselves. So that's done by having every single request send back the cookie that we were given when we were granted access and prove that we were who we say we are. So that token is our unique identity.

The problem is, and what Firesheep really exposed and turned the pressure up on these companies and essentially forced them to go HTTPS everywhere, or all the time, was that all of those roamings around Facebook were sending that cookie that was the person, it was their identity for the moment, for the session. It was sending it through the air, over the WiFi connection, in the clear. So Firesheep grabbed that, parsed all of the - it operated in so-called "promiscuous mode" with its WiFi radio, which is where, rather than the WiFi radio only receiving traffic meant for it, it would receive all traffic.

So it was seeing everything everyone was transmitting from their laptops to the access point. And it was looking at it, parsing it, finding out whether it was Facebook or LinkedIn or any of the growing number of sites, social networking sites and others, that it understood; and, if so, it would grab the cookie in the request header. And then it would go to the page, get the person's picture, and then stick it in a thumbnail in a little toolbar down the side. So you just turn this thing on at Starbucks, and it starts going bup bup bup bup bup bup bup, and people's faces are appearing, and you're looking around you, saying, oh, there he is over there. And then double-clicking on it allows you to impersonate him.

So this was a huge problem. And as you reminded us, I celebrated it because this was going on all the time anyway, and there was no way that it was going to be foreclosed on unless it was, like, really made to be a problem, so that the companies would go through the burdensome process of changing all their systems over so that they can be HTTPS, that is to say, SSL connections all the time. So we're more or less there.

Now, a security researcher named Brian Kennish gave a talk, a presentation at DEFCON about a year and a half ago, DEFCON 2011, on the prevalence and consequences of social widgets. He was noticing that widgets were just exploding all over the Internet. I mean, like Facebook's Like button, for example, which is, you go to sites now, and they're just lined up in a row. I sometimes am a little bit bemused that we're all supposed to recognize all of these little icons because it doesn't say Twitter and Facebook and Google. It's just they're little mini icon buttons. And look at the level of identity that has been established among these things. It's sort of breathtaking.

So back then, so this is a year and a half ago, widgets from Facebook.com, that domain, were then, and it's certainly more so now, found on 33 percent, so one third of the top 1,000 sites. Google.com had its presence, its widgets on 25 percent of the top 1,000, and Twitter on 20. And certainly those numbers are bigger today. And there's a neat site, BuiltWith, that I think we've talked about before. These guys monitor the technologies behind the web pages that we see, what are the technologies that websites are built with. And you can query their data, which is massive, in all kinds of ways.

Brian did this, to discover that Facebook Like buttons - and again, a year and a half ago - are up 63 percent in popularity year over year across the top 10,000 sites. Google +1 buttons were up, a year and a half ago, year over year 33 percent. So that's growth. And Twitter Tweet buttons, 35 percent growth. Now, here's the problem. Facebook deliberately used Facebook.com to host their widgets. Why? Because their members' browsers carry Facebook session cookies which identify them. And even if the cookie is no longer fresh, so that if you went to Facebook it would say you must log in, the cookie

may have expired, but it's present. So it knows who you were, even if it's not sure who you are.

Well, as we know, browser cookies are stored by domain and sent by domain. Which is to say that, if your browser asks Facebook.com for anything - a page, a picture, the image of the button, the social networking button, your browser identifies itself. It sees, oh, I've got cookies that were issued to me once by Facebook.com. I'm going to give them back. That's what browsers do. That's the essence of our ability to maintain a stateful relationship with a browser is that every single request sends back the cookies that the browser has.

So what that means is that the social networking sites are sort of the next generation of privacy concern. It used to be, and we've discussed this often, that the advertisers, the so-called "advertising networks" like DoubleClick, they were a concern because their prevalence across the Internet meant that your browser established an unwitting relationship with them because they would give you a cookie. If you didn't give one to them, they would give one to you, and then henceforth you would give it back to them every time you pulled an ad on any website you went to on the Internet. And so thus that gave way to this whole tracking industry that has upset people to varying degrees.

Well, this has now changed sort of - okay. That's all still in place. This is why Google bought DoubleClick is because it generates revenue, apparently from this information aggregation. Facebook, and other social networking sites, is even in a stronger position because the argument used to be that, well, DoubleClick doesn't know who you are. They just know you're browser 39264783, and you've gone to these various places. We can argue that Facebook, on the other hand, knows a lot about you. I mean, it knows all kinds of things about you. In fact, it's becoming controversial how much of themselves people are putting into Facebook. So Facebook has all of this volunteered information and networked information. And then the 'Net has become littered with their Like buttons. So every time any Facebook user goes to any website that has a Like button, Facebook knows. So I'm not judging it, I'm just saying this is what's happening.

But that's the privacy side. There's a security side. And it's bad. The reason is these things are still not HTTPS. The widgets, the social networking widgets, Like buttons and +1 buttons and Tweet buttons and so forth, even though when you're using Facebook with HTTPS everywhere, all the time, turn it on, that's your use, your first-person use of Facebook, which Facebook is now enforcing as being over SSL. But other websites that host the Facebook Like button have no such constraint. Even on their secure pages, they may be HTTPS all the time, always secure. But their assets, the page assets, the images for these buttons are more often than not simply HTTP because that works better. They're guaranteed of it working. They don't have to wonder if it will or will not work. HTTP, the lower common denominator, always works.

So if you are back at Starbucks, and you are a Facebook user, and you are now not using Facebook because now you're protected by Facebook, if you've got SSL on and always secure, Facebook is protecting you. But if you then go anywhere else and encounter a Like button, your session cookie is in the clear once again.

Leo: So what does that mean? That somebody can impersonate - well, I guess that's what you're going to talk about next.

Steve: Yes. Well, yes. The next step. We are right back to where - almost worse than we were with Firesheep because Firesheep was only intercepting your first-party interactions

with the target site, and seeing those. Now, thanks to the fact that these social networking sites have their assets spread, just like ads are, they are essentially little identity ads, they're spread all over the Internet, and they're not secure. So you can go to one page, for example, TechCrunch, and it's got all those little icons lined up. Visiting that one page, every cookie that you have identifying you with any and all of those sites that you participate in goes out, in a series of queries, to retrieve those images for the buttons on that page when you visit that page.

So this is known as WidgetJacking. And it exists. So it's arguably more of a problem even than Firesheep showed us with first-party hijacking of sessions, so-called "sidejacking," because here, one page that's not related to any of the pages that are being compromised through this identity leakage, those cookies are going out with those widgets.

So what do we do about it? Well, it turns out there's a wonderful solution, much as there was with Firesheep. That, of course, required people to turn on security. There is an open source offering called Disconnect, and the site is Disconnect.me, as in Disconnect Me, obviously. It is open source, free, and cross-platform - Firefox, Chrome, and Safari. Brian Kennish, who I referred to earlier, he was the guy who did the 2011 talk at DEFCON about this, get a load of where he used to work. He was an engineer for DoubleClick.

Leo: So he knows all about it.

Steve: He knows all about it. Brian is the cofounder and original developer of the first version of Disconnect a couple years ago, which is why I've had my eye on this for a while. In his own biography he wrote: "Brian is the original developer of Disconnect. He spent many years tracking users as an engineer for DoubleClick, then Google, but now, like George in Episode 86 of 'Seinfeld,' is doing the complete opposite."

And in a blog posting of his on October 24, 2011, he posted: "Exactly one year ago I noticed a virus infecting the web. Facebook widgets, mostly Like buttons, were popping up everywhere - alongside the articles I read, the music I listened to, the videos I watched. Worse, Facebook was, and is, serving these widgets from the same domain, Facebook.com, as their login cookies. Being a tracking aficionado," and he says, "(I developed DoubleClick's mobile ad server and the kludgey precursor to Google's AdWords API)" - so this guy knows his stuff.

He said: "I recognized Facebook's strategy, collecting user browsing habits to sell to advertisers. That night I spent two hours writing 53 lines of JSON and JavaScript, and then two more hours making a Ghostbusters-inspired logo for it, to inoculate my browser. I called the Chrome extension, which works by stopping the flow of personal data from third-party sites to Facebook, Facebook Disconnect. I had done side projects before and included another extension that had 37 users. But I was thinking big this time. I imagined Facebook Disconnect could have 50 users.

"I was off by three orders of magnitude and change. Today Facebook Disconnect" - and remember, this is October 24, 2011. He's saying: "Today, Facebook Disconnect has over 150,000 weekly users, and the extension has been Chrome-only, until now. To celebrate Facebook Disconnect's birthday, we've created versions for Firefox and Safari and open-sourced the code as usual." And there's links to it. It's over on GitHub.

Then, in a much more recent blog posting, end of April 2012, they are celebrating having 511,900 active users across a seven-day period. What is significant, the reason I'm

telling everyone this, is it is a cool add-on: lightweight, behaves itself, installs easily into Firefox, Safari, or Chrome. And it's very quiet, a little button on the toolbar. And if you're curious, you can see how much blocking it has done. What Disconnect does is disconnect you from the tracking that the major sites that it understands are doing. And I'm looking at mine right now. Supports currently Facebook, Google, LinkedIn, Twitter, and Yahoo!.

And but the reason that I finally decided I needed to tell everyone about this is a couple months ago, November 2012, they added WiFi security to Disconnect, meaning that their technology is now able to prevent widgetjacking, which I just described. They're able to see that the subsidiary assets being pulled by a browser, being requested by a browser over a nonsecure channel, can be secured. And so on the fly they intercept, before they leave the user's computer, they convert them from HTTP to HTTPS. So where you would be surfing with any of these browsers at Starbucks and be exposing your Facebook cookies, if you went to any page that had a Facebook Like button on it, all you have to do is install Disconnect in that browser, and it will make the browser's same query for the button. Everything works identically. It'll just coerce it into HTTPS, and then your Facebook cookies are as secure as the cookies for the main site you're visiting that was inadvertently leaking this information into the air.

So I've been running it for a while. I am very impressed. It does a very nice job of protecting. And so if you're someone whose machine never leaves the house, then it's like, okay, not such a big deal. I mean, it is the case that unsecured data is going through the Internet as opposed to being wrapped by security. And our listeners who are generally more security conscious would probably just as soon have that. And why not break this tracking by the major social sites also, so that they're no longer aggregating, as he explains in his first posting. So it's free. It's open source. It's Disconnect.Me. And click one button, and it puts the shields up for you.

Leo: We should say that it's kind of a manual process; right? I mean, you have to go through one by one your stuff; right?

Steve: No. No. No.

Leo: Oh.

Steve: I installed it.

Leo: Oh, he has updated it. Aha. I see now. It's a plugin.

Steve: Oh, yeah. Yeah. He has it available in three flavors. I'm not sure why. I think it's mostly historical. It's because he did the Facebook Disconnect by itself, and then he, like, supported the different platforms, then did one for Google and did one for Twitter. But it's all in one now. I installed this one thing. It installed seamlessly in Firefox, and the secure WiFi, it's like a little checkbox in the menu, was turned on for me by default because why wouldn't you want it? And they have some very cool adaptive technology. If you try to use HTTP Everywhere, that plugin that attempts to force everything to be SSL, unfortunately that can break pages. And these guys understand that you can't just do that. You can't use brute force, force it on. So if it encounters things that fail, it falls back. It can't lock it down, so it just says, okay, well, we're going to let the page

continue functioning. We tried to make this secure, but were unable to. And the number of sites that it knows about is growing over time.

Leo: Interesting.

Steve: So, yeah, it's just - it's an install-and-forget little goody. Unlike NoScript, which is in your face and you're needing to mess around with, this thing just works.

Leo: How about Ghostery and things like that? This doesn't replace them because Ghostery's more about information.

Steve: Correct.

Leo: But it might be a nice adjunct.

Steve: Yeah. Ghostery is great because it just gives you - I ran Ghostery for a while. Then I got tired of the window blocking the page that I was on.

Leo: Right. It's more informational than anything else.

Steve: Yeah. It's a really nice wakeup call to show you just exactly how much of this is going on.

Leo: And in my defense, by the way, we use the Facebook Like button because that's a great way for people to share what they're, I mean, we have social sharing on all of our stuff.

Steve: Right. And this isn't breaking that, Leo.

Leo: Right, it's making it better.

Steve: It's making sure that your use of it is secure for the people who visit your page and encounter that button.

Leo: Right. Cool. Does it turn on SSL? I mean, what is it doing?

Steve: Yeah, it turns on SSL.

Leo: So if I don't have SSL on my site, is that when it steps back and says, okay,

well...

Steve: No, because the SSL on your site is incoming...

Leo: Oh, it turns on Facebook's SSL.

Steve: Correct.

Leo: Got it.

Steve: Yes. All it simply does is it coerces the non-secure outgoing queries to be secure.

Leo: I got it. You said that. I just didn't understand. I suddenly started thinking about myself, you see, as a purveyor instead of a user. All right. Cool. Very nice stuff, as always. A great way to keep track of this stuff is to listen to this show every week, of course. We do it 11:00 a.m. Pacific, 2:00 p.m. Eastern time, 1900 UTC at TWiT.tv on Wednesdays. But if you can't watch live, or listen live, by all means subscribe. In fact, you should anyway, so that you have every episode. There's 386 so far. This is our 386 episode.

Steve: I know. I saw that, too. I saw it, just a little smile to myself. I remember when I was coding to that chip.

Leo: [Codger voice] Yeah, I remember the 386. Remember the DX?

Steve: Yup.

Leo: Remember the SX? They just disabled the numeric processor.

Steve: And the DX2 that followed. I mean, god, remember, we were, like, living on megahertz.

Leo: Oh, I thought - oh, man.

Steve: I can get 9MHz.

Leo: The 386 was how fast? It wasn't till we got Pentium that we got, like, into the double digits, was it? I guess we were at double digits. I remember Pentium 90.

Steve: I remember 33MHz and then 66.

Leo: 33, that's it. And the DX2 was 66. Boy, you have a good memory. That's exactly right, yeah. Anyway, what was I saying? Oh, yes. Sidetracked.

Steve: We were trying to wind this down.

Leo: No, no, I just, you know, start talking about 386's, and, well, you know what happens.

Steve: Yeah.

Leo: You can get copies of this from Steve's site. He has 16Kb audio, the smallest we make available, and text transcriptions at GRC.com. While you're there, you might want to pick up SpinRite, the world's finest hard drive maintenance and recovery utility.

Now, the people in the chatroom were asking - because we had that SpinRite letter from a user who was using it in a virtual machine.

Steve: Yeah.

Leo: Okay? Wasn't there an issue maybe that maybe you'd get some rights to that disk while you're SpinRiting it or something?

Steve: No, no, no. What happens is, in order for you to make direct physical access available, it takes it away from the rest of the system. So that drive disappears from outside the virtual machine and is only available inside the virtual machine.

Leo: Good. So nothing to worry about there.

Steve: Right.

Leo: I knew you wouldn't recommend such a use unless it were safe.

Steve: No. And I've heard about this for a while. People have been doing this on various of the virtual platforms. But I just ran across this nicely assembled explanation. I thought, oh, that's a perfect thing to share.

Leo: Good. I think you - there were a number of people saying, wait a minute, I thought you couldn't do that. Yes, you can.

Steve: Yeah.

Leo: So that's the kind of thing you find out when you go to GRC.com. You get SpinRite, you get the freebies, you get the stuff. You can also follow the Steve on Twitter at SGgrc, so you keep up with all his links. But then we also have that bit.ly/sggrc. All lowercase?

Steve: Yup.

Leo: So people can just - this was - thank you, by the way, to our Danish listener who made that available - follow your [indiscernible] tweets. And I guess that's going to be up to date automatically. And then of course we make higher quality audio and video available at TWiT.tv. But best way to do it is subscribe at iTunes. Do you make a podcast feed of the 16Kb version?

Steve: No.

Leo: So you just can download that.

Steve: People just go there and get it.

Leo: That's for you primitive types, still using 386's. But for the rest of you, subscribe in iTunes or Zune or whatever, DoggCatcher, Instacast, whatever you use to get podcasts, and you'll get every week. You won't miss a single episode. We will see you back here next [Wednesday], Steve.

Steve: Thanks, Leo.

Leo: Oh, it's a Q&A segment. GRC.com/feedback, if anything you've heard on this show has stimulated your thought process.

Steve: Yes, by all means, send a note, and I will go through the mailbag the day before, and probably also the morning before, and pull a bunch of goodies. So thanks, everybody.

Leo: Thank you, Steve. Somebody in the chatroom is saying, does TWiT benefit more if we subscribe to any particular source? No. As long as you are downloading - the only version that we don't count is Steve's 16Kb version. But if you're downloading a version of the show...

Steve: Those are being counted, Leo.

Leo: Oh, yeah, that's right, we added those to the counts, didn't we.

Steve: Yup. I'm bouncing through Podtrac so you get credit for those.

Leo: Thank you. Bless you. Bless you, bless you. So, yeah, as long as - and you can tell because, yeah, you'll see a Podtrac redirect. But if you subscribe to any of the XML feeds, that will also be counted. Any download or subscription counts. And it's a great way to let us know that you like this show. This show, by the way, one of the few that is growing all the time.

Steve: That's neat.

Leo: More and more nerds. I think the show is highly shared among security experts.

Steve: We are anabolic rather than catabolic.

Leo: That's what I say.

Steve: [Laughing]

Leo: Steve, thanks so much. We'll see you next [Wednesday].

Steve: Thanks, Leo.

Copyright (c) 2012 by Steve Gibson and Leo Laporte. SOME RIGHTS RESERVED

This work is licensed for the good of the Internet Community under the Creative Commons License v2.5. See the following Web page for details: http://creativecommons.org/licenses/by-nc-sa/2.5/