SECURITY NOW!

Transcript of Episode #218

Listener Feedback #77

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

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-218.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-218-lq.mp3

INTRO: Netcasts you love, from people you trust. This is TWiT.

Leo Laporte: Bandwidth for Security Now! is provided by AOL Music and Spinner.com, where you can get free MP3s, exclusive interviews, and more.

This is Security Now! with Steve Gibson, Episode 218 for October 15, 2009: Q&A #77. This show is brought to you by listeners like you and your contributions. We couldn't do it without you. Thanks so much.

It's time for Security Now!, the show that covers all things security oriented, like your privacy, browsers, hackers, bad guys. Steve Gibson's the man in the know, the head at the Gibson Research Corporation, GRC.com, creator of SpinRite, discoverer of spyware, and our esteemed host for the last four years plus. Hi, Steve.

Steve Gibson: Hey, Leo. Great to be with you again, as always.

Leo: Sorry I wasn't here last week. I hear it was a scary show.

Steve: Well, we did. We frightened a lot of people. Not, you know, not, I think, unduly. We had twice the normal amount of feedback from people who wanted basically clarification of different types about specifics of what we discussed last week. So today's Q&A, we have a few random things. But largely I did, because we got, like, literally twice as many submissions of questions, I wanted to spend some more time to clarify some of the finer points of this. And, you know, unfortunately you missed last week's episode. So this will give us a chance to catch you up on this because it's pretty significant. And I think we're going to be touching on this aspect of what I called last week "The Broken

Browser Model" somewhat here and there in the future because it is, it's a fundamental aspect of the way we're using the Internet today, which as we'll see is not very secure.

So we've got a bunch of security news. Get this. The largest Microsoft second Tuesday of the month update ever. There's never been more things fixed at once. And the good news is we've talked about several of these things that have been fixed previously, waiting for Microsoft to catch up. And this is their catch-up Tuesday. So lots of things caught up. Some other random security news. A really sort of interesting fun SpinRite story. And then some great questions that'll help to clarify some of what we discussed last week.

Leo: It's kind of ironic that we all think that things are getting better and better, and you should need fewer and fewer patches, and yet there's more and more patches.

Steve: Well, remember...

Leo: Seems like it's the wrong direction.

Steve: Well, things are getting more complicated. And as we've often said, complexity is the enemy of security because the more complex something is - I think, you know, it seems natural that things are getting more complicated. And techies enjoy complexity, I think, for its own sake. So you just, as things get more complicated, there's more opportunity for mistakes. And the security always is the weakest link in the chain. Well, the more links you have in the chain, the longer the chain is, the more opportunity there is for someone to miss a weak link. And then of course the bad guys spend all their time looking for the weak links. We just hope the chain holds together. So it really, it is, in many ways security is a much tougher battle than the way people use software, which is, oh, it's working fine. That's all I need is it to be working. Well, no. With security it needs to be working perfectly. And I mean with a capital "P."

Leo: Yeah, yeah. Do you think Windows 7, which is due out in a week, will change everything?

Steve: No. It's new. It's bad. New is bad.

Leo: [Laughing] I love - you're so - you're so conservative. It's new. I don't like it.

Steve: Look at the evidence. XP was - remember Ballmer was jumping around, saying it was going to be the most secure Windows that had ever been made. And it turned out it was the biggest disaster they'd ever had.

Leo: True, true.

Steve: It's new.

Leo: True, true.

Steve: It's new, it's bad.

Leo: Well, we've got questions. You've got answers. So what's the latest? Do you want to do security news?

Steve: Yeah. Well, we've got some. As I said at the top of the show, the big news of the week is the biggest ever mega monthly update from Microsoft. We talked about the SMB v2 vulnerability a couple weeks ago, which was a concern for the newer Windows clients which support v2 of SMB. There was the possibility of remote code execution. That was fixed. They fixed multiple vulnerabilities, critical vulnerabilities in the Windows Media Runtime, also in Windows Media Player. IE got four critical vulnerabilities fixed.

Leo: Geez.

Steve: Oh, we're just warming up here. The ActiveX Killbits was updated. Remember that that's the thing which prevents ActiveX controls from being instantiated or invoked by Internet Explorer. So those were updated so that fewer things on the system could be misused. They finally fixed this ATL, the Active Template Library problem. That was a bug in their library which meant that all programmers around who were using the ATL system to create ActiveX controls were inadvertently creating vulnerable ActiveX controls. So that's been fixed so that anyone who now compiles using the Active Template Library, which is one of the tools Microsoft provides for creating ActiveX controls, will no longer be creating inherently exploitable and commonly exploitable, that is, with an exploit that everyone knows about, ActiveX control. So that was finally fixed.

Multiple vulnerabilities were fixed in the .NET system which is becoming an increasingly popular programming model for using Windows. Multiple GDI+ vulnerabilities, which is the enhanced Graphics Device Interface library used by pretty much everything. We talked a few weeks ago about the problem with Microsoft's web browser, IIS, and its FTP server, warning any of our listeners who did have a publicly exposed FTP service running on IIS to think twice about that. That got fixed. There was a vulnerability in their indexing service that they fixed, Windows kernel elevation service that they fixed.

And finally the biggie of the month, that we've been waiting for, which we've talked about several times now, is they had the problem with null bytes embedded in certificates which allowed spoofing of SSL certificates. It turns out that another little bit of news is that this was shown by our friend Moxie Marlinspike, who brought to the world's attention the fact that you could create a certificate with the name www.paypal.com and a null character, that is, a zero, and other web domain like mymachine.insecuresite.com. You could, because the certificate was really being issued to insecuresite.com, you could get a certificate authority to run through the automated process to give you such a certificate.

The problem was that, if you used that certificate, the web browsers would only parse the name up to the null character because the so-called null-terminated strings is the modern way of storing strings of characters. So we talked about this months ago where the original Pascal means for storing a string was for the first byte to be the number of characters in the string, that is, the length of the string, followed by those characters. The problem was that allocating a byte for that meant that strings could never be longer than 255 characters, which is the maximum value you could store in a byte. So that approach was abandoned in favor of so-called null-terminated strings.

But the problem with that is it's one of the sources of major vulnerabilities in buffer overruns and so forth that are assuming, they're, like, scanning a string, waiting for the null. But if a bad guy can put in their own code that doesn't contain any nulls and then, like, store that where a string is expected, that's the source of many of these problems. So Moxie publicized a similar vulnerability.

Well, Microsoft finally fixed it with the update two days ago, on the second Tuesday of the month, which was two days prior to this podcast on Thursday. And until then IE and the web-based version of Safari and Chrome, Google's Chrome browser, all which used and were dependent upon this Crypto API, which finally got fixed - and this has taken a long time. This vulnerability's been out there and known for a long time. Firefox fixed it promptly themselves. So even Firefox on Windows had not been vulnerable. Finally now nothing is.

But in an interesting related little tidbit, PayPal suspended Moxie Marlinspike's donation account, apparently out of some fit of pique with him over having created this certificate. What happened was that somebody else created one which was then made freely available on the Internet, posted on various sites and in various security blogs, as a further demonstration. Moxie in his presentation showed this as a proof of concept but, being responsible, did not use it to create a certificate that was then made public. Somebody else did. And PayPal suspended Moxie's account because he had on his page a donation to support the use of his SSL sniff utility. And PayPal said, well, we don't encourage or, you know, it's a breach of our terms of service to collect money to support a program being used to deliberately promote insecurity. So that was a little disappointing. I don't know how long that's going to stay the case. But that's something I just picked up on a couple days ago.

It is time for people to check the currency of their Adobe Acrobat Reader and Acrobat program. There are targeted attacks now against - that are occurring for people who open a PDF file using Acrobat v9.1.3. Apparently 9.2 was supposed to be made available. And I did check - I'm still using 8, so I never was under the problem with this vulnerability. But anyone using 9, supposedly Adobe did their monthly update, also on the second Tuesday of the month - I'm sorry, their quarterly update. In Adobe's case it's a quarterly update, not monthly. Although, as we've already seen, they've not been holding to that at all because they've had so many problems that they've had to be addressing.

Leo: They had 29 fixes yesterday [laughing].

Steve: Yes.

Leo: Well, that just shows you, if you hold onto them long enough, you can really build up quite a backlog.

Steve: Which is why it's just nuts, this idea of doing this every three months. It's like, what are you guys thinking? And they said in their original policy that, well, we want to let people know when we're going to be doing updates. It's like, okay, well, as we know, several have been so bad they haven't been able to wait. And then they have like this huge batch that they fix at one time.

And the problem is, when we say "targeted attacks," what that means is that there's a known problem. And rather than just in the wild problems, specific people are being fed malicious PDFs. Like executives in corporations or in banking firms are being targeted with known email addresses and letters written specifically to induce them, sort of based on knowing who they are, to open this PDF. And when that happens, malware gets installed. So, I mean, it's interesting that, you know, here's Adobe being reluctant and slow to fix these problems, yet they're a vector of really significant security threat. So...

Leo: And everybody has it.

Steve: Yes, exactly.

Leo: On Windows, anyway. You don't need it on the Mac. But on Windows everyone does.

Steve: Exactly. It's a very, very common application, in order to be able to read PDFs. Comcast has started doing something interesting. They've opened a pilot test in Denver. And I'm of two minds about this. They call it Comcast Constant Guard. And what they're doing is doing 24/7 traffic analysis monitoring of...

Leo: To protect you. It's to protect you.

Steve: ...of their subscribers. And...

Leo: For your own good.

Steve: And then doing a browser intercept, which is I think the controversial thing.

Leo: Oh, dear.

Steve: Now, they say that they've been doing traffic monitoring and notifying their customers, their ISP subscribers, by phone for the last two years. Well, I think that's preferable because - and apparently they report that their subscribers to Comcast love being notified that they've got malware on their computer. So you get a call, a phone call from a Comcast person saying hi, this is your cable provider. We wanted to let you know that your computer is evidencing a traffic pattern on the Internet that gives us strong suspicions, strong reason to believe that it's infected with something. So go to the following URL, and you can get instructions for free how to remove what it is that we believe you've been infected with.

What they're now doing is, they say because of the extreme popularity of this - of course it's expensive for them to have their people phoning their subscribers - they're going to automate this with a browser intercept. Which, eh, is a mixed blessing. It means that when you are surfing somewhere, some server at Adobe that is - I'm sorry, not at Adobe, at Comcast. Some server at Comcast, because they're your ISP, they are in your traffic stream, your traffic is transiting across their equipment, they will give you a - presumably return a redirect or just give your browser content different than what it's expecting, which is an intercept page notifying you that you've got malware on your computer, they believe, that the traffic analysis that their systems have generated lead them to believe there's some bad stuff on your computer. Click the following link to go and take care of it. And apparently you can push past that if you choose not to, that is, it's not a you can't do anything on the Internet until you fix it. But it is intercepting your use of the Internet.

Leo: That's what worries me. Because, you know, Comcast was before using that software from Canada to watch for BitTorrent. And I worry that this is a backdoor way to kind of, we're protecting you, but also we'd like to see what else you're doing online.

Steve: Well, yes. And it's funny because this takes us - it's a perfect segue into the next thing I wanted to mention, which is that an Australian ISP, iiNet, has been taken to court by a consortium of movie companies who have sued this ISP for not disconnecting subscribers based on the movie companies' say-so. There is a so-called "safe harbor provision" in Australia that protects ISPs as long as they take "reasonable" actions to prevent copyright infringement. Well, I mean, this is one of the problems with laws that are written broadly, you know, what is "reasonable"? Now we have to go to court to have an interpretation of what that means.

The Australian ISP, iiNet, is saying, look, an allegation of infringement is different from a proof of infringement. They're saying to the movie companies, you take these infringing subscribers of ours to court, prove that they are infringing your copyrights, and we'd be delighted to disconnect them. We'll jump up and down to disconnect them. But we're not going to disconnect anybody based on your allegation that they're infringing, just because you say so. So it's an interesting question which - and it'll be interesting to see how the court decides this because, again, this is - we've already covered in the past this general move by, in the U.S. case, the MPAA that has said as a policy they are no longer going to go directly after the end-user infringers. Instead they're going to work with ISPs to somehow discipline these guys. And so here we have a case where this is not working out so well, where the ISP is saying we need a reason, we need good cause to disconnect users. And in our opinion the movie industry's statement that the following users have copy-protected content and are distributing it is insufficient grounds. If you prove it, that's fine. Not if you just claim it.

Leo: Interesting. I mean, we have the same kind of a takedown rule in the states, of course. But generally what happens is they write a letter to the ISP saying, you know, this guy is stealing from us. And the ISP then warns the person - I've seen this happen time and time again - saying stop stealing.

Steve: Stop doing it.

Leo: And usually they give them a few strikes.

Steve: Yes.

Leo: So that's interesting. I'll watch this with interest.

Steve: Yeah. Be interesting to see how it goes. I have two little bits of errata. One is that a couple users wrote in to say they cringe every time they hear me say Mac OS X. And I have to say...

Leo: Notice I don't correct you on that.

Steve: I noticed that. And you always say OS 10. And I thought, well, okay. Finally I'm going to get a clue, here. OS 10, period. So I've seen the light.

Leo: You know why I don't correct you? This is actually something newspapers and other journalistic endeavors always have to deal with, is do you go with the commercial typography? Remember CNET was C, a bar, Net capitalized, lowercase and stuff.

Steve: And how, yeah, how do you pronounce that?

Leo: Well, and it's not even - in a newspaper it's not an issue of pronunciation. It's an issue of - here's a really good one. Do you put the exclamation mark in Yahoo!?

Steve: Right.

Leo: Many journalists believe that's commercial speech, and you just put Yahoo without an exclamation mark. But Yahoo! would say, no, no, that's our trademark is Yahoo exclamation mark. We want the full thing. And so I think - I don't - look, it reads Mac OS X. It's doing Apple's - it's carrying water for Apple to say, no, no, that's 10. You read it any way you want. It's not, you know, we're not here to advertise for Apple. So it's an "X" on the page. You can say "X." It's not incorrect.

Steve: And so someone said, well, clearly when it was OS 9, it was OS 9. Now, if they went to XI, I'm not going to go OS XI every time. I would probably say, I would get a clue, say okay, OS 11. But anyway, so...

Leo: Yeah, you know, a lot of people say OS X. I don't have any problem with you doing that or I would have said something.

Steve: Well, thank you, Leo. I'm going to be going - I'm going to try to correct myself and say OS 10 from now on. Even though somehow, I mean, even now knowing it, I just look at it, and I want to say OS X. So...

Leo: It's an X.

Steve: I've got to shake the habit. I discovered something interesting the other day. Some stuff wasn't working that I expected to have working. And I discovered that installing Microsoft Security Essentials replaced my hosts.ini file.

Leo: [Gasping] Interesting.

Steve: Yes. And not surprising, but I thought I would just bring that to the attention of our listeners. We've talked often about the hosts, how the hosts file can be used to redirect DNS lookup. And there are people that maintain hosts files that you can download. It's a very handy way of just ever keeping your computer from asking for a specific domain name. You're able to assign it to 127.0.0.1 or 0000, whatever you want, an IP that doesn't go anywhere, essentially. And your computer by convention, this has been the case ever since UNIX first got put on the Internet, the hosts file is checked for domain names prior to any DNS lookups being made.

So I had a bunch of things. In this case some of them were privacy and security related, but also I was just sort of using it as a poor man's DNS server to redirect a bunch of strange domains that I use internally. And that broke. And I was scratching my head for a long time until I remembered, wait a minute. Didn't I put this in the hosts file? And I looked there, and it had been - mine had been renamed to .bak. And the date of the new one was the date I installed Microsoft Security Essentials. And Microsoft Security Essentials had a note I thought was very nice, they put a note in there to say that that's where this hosts file came from. So it wasn't at all a mystery to me. It told me. But I thought, okay, well, it would have been a little nicer if they'd made a popup or something that said, hey, we noticed you've customized your hosts file.

Part of installing Security Essentials, because this could be maliciously changed rather than deliberately changed by you - and this of course is why they do it. Malware has been known to make changes to hosts files. And so Microsoft is...

Leo: That's true, that's true.

Steve: ...putting it back to something - basically they nulled it out. They removed - there were no fancy changes they made. They just put it back to the original one, which does nothing. It's there, but it's got no entries in it, essentially, just a bunch of commented lines. So...

Leo: Do they make it read-only?

Steve: Good question.

Leo: I guess that'd be easy for malware to change, though.

Steve: Mine was. So, yeah, I mean, that wouldn't slow down malware very much. So, yeah. So anyway, I got a kick out of that. I thought I'd just bring it up to our listeners' attention, for anybody else who was using a hosts file and installed Microsoft Security Essentials. And I got a fun SpinRite note. The subject line, this was sent through our sales email from an Andy Kinsey in Haddington, Scotland, UK. And he said, "SpinRite saves SALON." And he had "salon" all in capitals. And I'm thinking, well, I don't think he means the magazine or the website or anything. But it caused me to read it.

He said, "Hi, Steve. I want to thank you so much for SpinRite and what it did over last night. Friday was my day off. I have a job, and I freelance technical and web design. Anyway, I was called at 9:00 a.m." - apparently on his day off - "by a client whose computer wouldn't boot. The machine was the till," as he put it, "was the till and had three years of accounts on it. For some reason my web design client hadn't externalized backups nor looked after the machine at all." So I think what he's saying is that his responsibility was not this salon's front counter cash register till, but they called him because he was a technical guy that did their web design.

So he said, "When I got there, the hard drive had no signs of life. I took it out and attempted to manually power it via my cables and USB devices. No win. I was at a loss and began the long, laborious process of indexing the backups and attempting some kind of recovery as there was no access to the hard drive." So I guess they had non-recent backups or something. He said, "Through USB in my other machine, it was visible, but no access was given, nor any indication of its size. It was only some hours later I had the brain wave," as he put it, "SpinRite could help. So I went to your site, bought the software, put it on my USB, and stuck it in the machine with its dead hard drive. I ran SpinRite. Some 18 hours later SpinRite was complete. I removed my USB and waited in anticipation. Two minutes later I booted the machine with bated breath. It worked. I was in. I immediately took a copy of all the data and backups to my external USB drive. I rebooted the machine again, and it was bricked. I knew it would be. I've never seen a hard drive in such state."

So basically SpinRite brought it back, literally, for its last gasp. And he did the right thing because this drive was in such bad shape, as he said, he immediately pulled the data off of it and got everything. He said, "Anyway, we managed to find another system and reinstall everything, including all the data. It took another day to get it all sorted, so two days of work. Whilst this was going on, the manager was calling around, trying to get another system. The cheapest was 5,000 pounds per year, money this salon can't afford. They border on extinction every day, never mind having to find 5K. Thanks solely to spending $89 for SpinRite, the salon was able to use what it had and is still in business. Thank you, Steve. You saved the salon."

Leo: Wow. Now maybe they'll do better backups.

Steve: That was cool. And I think they probably got a lesson. They probably learned a lesson in the process.

Leo: SpinRite is probably that, you know, it's that little window of opportunity between success and failure. And it probably has taught a lot of people a lot of lessons.

Steve: Yeah, yeah. I mean, I think you're right. It's like just if I could only have my drive back one last time, I promise I will be good from now on.

Leo: It's the answer to that prayer, please, just one more time.

Steve: It's a little bit of a time machine. Just, you know, it's just turn the clock back one day. I just need yesterday again.

Leo: Leo said I should back up. All right. Let's - we've got questions for you.

Steve: Yes, we do.

Leo: And we can get right to them, starting with Andrew Branagan, in Carteret, New Jersey. This has to do with what we were talking about earlier, Adobe's Patch Tuesday. He says it's a Patch Tuesday headache: Good Wednesday morning, Steve. We record the show on Wednesday, so that's apt. Just wanted to point out that, even though Adobe did release their quarterly update, they're not making it easy to distribute. They're dragging their feet on releasing an .msi - you know, those are the Windows installer packages for version 9.2. When you go to their site and enter your information to get a distributable copy of the software, they send you a link which contains version 9.0. Just thought you might want to incorporate this into the security news today. Looking forward to the show. Nice. Well, maybe they'll fix that by the time we get this on the air.

Steve: We can hope so. But I did want to give a heads-up to our users that of course there's the sort of the slipstream automatic update approach that you get when you use your Reader or Acrobat and tell it to look for updates. That's just the update that Adobe normally provides. There is, however, for people who want a so-called redistributable update, where you get the actual file itself in, as you said, an .msi format, which you can then yourself individually run on different machines. It's probably just an oversight. Maybe - I mentioned that it was Wednesday morning because, as of our recording of this, 24 hours from the time Adobe released this, it's still not fixed. So I just wanted to point out to our listeners to make sure, if you're using that, that you're getting the 9.2 fix from Adobe and not this older, retro version that you definitely don't want.

Leo: Question 2 and Question 3 in one big ball. Starting with Patrick McAuley in Guelph - near Toronto - Canada. He has a man-in-the-middle question for you, Steve: I'm not really a techie, but I've been listening to Security Now! for a couple of years now. I've learned a lot about keeping myself safe online. Last week's show with Alex subbing for Leo was a great one, but a bit scary as you revealed how someone can get between me and an apparently secure login screen to capture IDs, passwords, et cetera. Were you talking about click fraud?

Steve: Nope.

Leo: That's another one.

Steve: We'll get through this in a second, yeah.

Leo: One thing that was not clear to me was whether this loophole only occurs if the man in the middle has somehow gotten access to my LAN, or if it's a danger on any Internet connection. Right now I'm on your site from my home computer, connected to my router by cable. I don't think there's any way someone can get access to this LAN. So am I safe? And further, if I use my notebook to connect to my router wirelessly, using WPA encryption, am I safe there?

And Ted Lind in Woodstock, Illinois had a similar broken SSL question: I want to make sure I understood you correctly, Steve. The man-in-the-middle attack you described requires the bad guy to be on your local area network. If I'm using SSL to do a bank transaction, I'm connected to my private network using WPA2 and one of your really long passwords. It's my understanding that this is still secure because the man in the middle cannot get through the router. Both my wired and wireless computers should not be vulnerable to this attack on my home network. Am I right? Also if I'm on a public network, but the first thing I do is set up a connection with Hotspot VPN, is this also a secure way to do an SSL transaction? Love TWiT; love Security Now!. My car radio is constantly tuned to one of Leo's podcasts. Also a SpinRite owner. Thank you, Ted. So this I want to hear because I didn't hear everything you talked about on the episode. Obviously I'm going to have to go back and listen to it.

Steve: It was a good one.

Leo: Yeah.

Steve: So here's the - I've been thinking about it in the intervening week. And I think I have a simpler way of describing the problem. The example I gave was a specific instance using ARP spoofing, which we've talked about in the past, in any local area network scenario, to allow someone to intercept traffic to users of the network, thus creating the man in the middle. The focus of the podcast, though, which we titled "The Badly Broken Browser Model," was it noticed that, if you were logging on from a nonsecure page, that you could not trust the form that you were using to accept username and password because, if the page was nonsecured, then a man in the middle could have intercepted the page and, for example, taken the "s" off the https on the form's Submit button so that the submission would not be secure, which would allow that man in the middle to intercept and acquire the login data or whatever it was you were submitting to the site that you were connecting to. So the idea being the broken thing about the browser model is that there's nothing that protects the content of the pages we're receiving from a remote server from being edited on the fly unless it's secure, unless the content that's delivered is secure.

But we've talked about how pages you submit, pages that you use to provide information don't really have to be secure. It's your clicking of the button, it's the submission of the information that needs to be secure. However, as we looked at last week, that's not the case if you have somebody clever in the middle. And what Moxie brought up and made very clear during his Black Hat presentation was that somebody in the middle could deliberately edit pages which you were receiving from a secure site to quietly drop the security.

And so the example I gave was in a public, for example, in a public WiFi scenario, in a hotspot, where you were inherently using an open LAN, I mean, it's a LAN, an Ethernet LAN, which is very sniffable and where ARP spoofing can be used to insert a man in the middle. And last week we described the statistics of the number of secure logons, PayPal logons, credit card numbers, very common logons that he acquired doing this during a 24-hour sniffing period. So it's extremely effective. So now to answer both of these listeners' questions and many similar questions that people submitted, any man in the middle, that is, a person at any point between you and the website you are connected to, has the ability to do this.

So the one example I gave was a LAN scenario in WiFi. These guys were asking about what about their personal local area networks. Well, the point to remember is that anyone anywhere between you and the remote server. So certainly the location of greatest vulnerability is probably the network closest to you. But in theory somebody who had some malicious intent anywhere in the traffic pattern, upstream of the ISP, downstream of the ISP, at any of the routers along the way, anywhere in the stream, someone could insert themselves and perform this kind of filtering.

So the thing that SSL connections are designed to achieve is end-to-end, that is, endpoint-to-endpoint privacy and authentication. And so the beauty of SSL is it does protect you from man-in-the-middle attacks anywhere, anywhere between those two endpoints. You need to make sure that you actually have connected to the remote server. We talked earlier in this podcast about that null, the null character in the middle of a deliberately malicious certificate that could spoof you so you thought you were at PayPal, but you were actually at another site, but your browser and the system would only see www.paypal.com. So there was no way, looking at that, for you to tell that that wasn't where you were. Microsoft fixed that with day before yesterday's mega security update. So that's a good thing.

That's been a glaring hole. But the problem is that any time you receive a page from a remote server which is not over SSL, you don't know that it wasn't modified. That's the focus. Anytime you receive a page that is not over SSL, a man in the middle occurring anywhere could have changed it so that you cannot rely on it. Now, several people...

Leo: But that almost seems trivial to point out.

Steve: Well, it is except that...

Leo: I mean, of course my Internet service provider and every server along the way can modify that page.

Steve: Okay. And so of course the point is that, if someone did so, you would have no way of knowing that, when you submit your username and password or your credit card information, that the button you're pressing is not SSL, or isn't SSL to some malicious party. And so that's really, I mean, that's what, at its core, that's what we made very clear last week is that you have to, in order to trust form-based submissions, the form itself has to be delivered to you over an SSL connection. It's really not sufficient to trust that the button will be SSL because, if the form itself is not secured, then anyone could have changed it. And so what we were making very clear is that those changes completely bust the security model. Basically people are using browsers. We've adopted a model which sort of works, but which is really not secure.

Leo: Okay. It doesn't, I have to say, that doesn't terrify me. But if it terrifies you, okay. It's good to be aware. Question 3, Jean-Matthieu Bourgeot in Tarare, France had an interesting idea for securing public WiFi hotspots. He says: Hi, Steve and Leo. Listener from day one, love the show, been learning so much with you guys, blah bah blah. Here's an idea for securing public WiFi hotspots that came to mind. Not sure if it'd work.

On public WiFi hotspots, users obviously do not need to have their computers be able to directly talk to each other just as if they were on an office LAN. Usually you're talking to the outside world, not to the guy sitting next to you. The fact is that all the computers are on the same LAN, and therefore, as you just said, are prone to ARP spoofing or OS exploit attacks, et cetera. If the WiFi hotspot's DHCP server would assign IP addresses belonging to different subnets - oh, this is interesting - to every new computer, so 192.168.0.10, then .1.10, then .2.10 and so on, would this - I don't know if that's a different subnet, though; is it? Would this prevent many of the possible via-the-LAN attacks? Also, this solution would be very cheap to implement by just changing the DHCP server's behavior. What do you think?

Steve: Well, it was an interesting idea. We're familiar with the idea that, for example, if you have a net mask - which the net mask is used to create subnets. So a netmask of 255.255.255.0 would say that the network number is contained in the first three bytes, or first three groupings, and that the machine within the network is in the fourth one, the last one. And so then the idea would be, if you were to assign each machine on the WiFi system its own subnet, so that they weren't - so that no two machines were on the same subnet, would that give you more security? And the answer is, well, it would give you some little bit more security. But it would not give you any protection from, like, from strong hacking.

The way any Ethernet works, Ethernet is addressed based on MAC addresses. And the ARP table, as we did discuss last week because we did a little bit of review of how ARP spoofing works, the ARP table associates IPs to MAC addresses. So that the packets which come into the gateway, for example, inbound to the hotspot, the ARP table in the gateway looks at the packet based on its IP addressing and sees which MAC address owns that IP on the LAN, and then the packet is routed based on the MAC address.

So the problem is that subnetting is sort of a - it's a logical addressing layer on top of the physical addressing layer, which is MAC-based. But if you were doing any, for example, promiscuous sniffing, where you had a WiFi adapter, and they're readily purchasable, which allows you to sniff all the packets on the hotspot, it would see all the packets in all the subnets that were using that WiFi. So it does not provide you any useful security.

Leo: Damn. Seemed like such a good idea.

Steve: It's a neat - it's a neat idea.

Leo: Jason Learmouth in Sydney, Australia writes. He's got some thoughts about the broken - I don't want to call it the broken browser model because I think that's confusing. It's the broken browser paradigm. Let's use that.

Steve: Okay.

Leo: Because browser model means something else to programmers. Steve and Leo, I listened with great interest to your discussions on the state of play with secure browser sessions and the session hijack trojan out there, stealing people's money. Steve, you mentioned in one of your listener feedbacks that the authentication needs to be moved closer to the transaction. While I agree this would fix the problem for now, I expect it would only be a matter of time before attackers moved closer to the transaction, as well. Discrete applications were suggested as a way to offer a secure connection-based solution. Steve correctly pointed out we have enough stuff installed on our computers already. The browser is very convenient.

So maybe - I think this is actually a good idea - the browser could run an application based on Java or some similar technology to provide the best of both worlds. I've seen some SSL VPN providers - I think GoToMyPC does this, and GoToMeeting - download a Java app to create a tunnel to the network. That's exactly how Citrix works. I believe Google uses this type of technology in its Docs product, which offers very near real-time document collaboration. There must be - I don't know if Google's doing that with Wave. I don't think so. There must be some two-way traffic there beyond just http. They're using - actually they're using the Jabber protocol.

But anyway: What about Jungle Disk? It encrypts before sending data to the cloud through an SSL tunnel. How does that avoid being vulnerable to attacks? Or does it? Could a site offer a local application to the user that would handle all the security, authentication, and encryption through its own persistent connection without requiring a local install? Love the show, happy to hear my name on the show if you feel like reading this. Thanks, Jason.

Steve: Well, that is, I think, as you say, Leo, it's a fundamentally good idea.

Leo: Of course you have to trust Java. But barring an exploit in Java.

Steve: Right. Now, it is also essentially what we've been talking about with some level of disparagement about all this ActiveX stuff. You know, ActiveX is an application which is transparently run by the browser. Microsoft, recognizing the fundamental security problem with that, has in recent versions of IE, and Firefox does, too, warning users that this page wants to execute an ActiveX control, you know, do you want to proceed?

So I do think that the notion of using the browser to encapsulate an application which is provided to the user in a transparent way, which exists in sort of transient form on their machine, which they don't have to separately download and install and manage, which won't clutter up their Add/Remove Programs list with an infinite number of individual applications for everyone you want to have a secure transaction with, I think that makes a lot of sense. And if the remote system then refused not to - if it refused to operate without its own dialogue with its own application, then it would in fact be able to create the kind of containment that we're looking for which is fundamentally more safe and secure than the transaction-based, sort of fundamentally dangerous model that we've so far been using with our browsers. So, yeah, I think it's a good idea.

Leo: Great. Question 5, Dale Willer in Kansas City asks about ARP spoofing on a home network. In Episode 217, the last episode, "The Broken Browser Model," it wasn't clear that the ARP spoofing attack, if the ARP spoofing attack and the scenario presented in that episode is a threat on a home LAN behind a router. My first impression was it's only a threat at public hotspots such as airports, Starbucks, et cetera. Later on I wasn't so sure. Please clarify. Also one way to protect against this at a public hotspot, always use your VPN if you have one; right?

Steve: Yeah. I put this in here because I wanted to make sure as I was going through these that I didn't forget to mention that...

Leo: Because we already answered the first part.

Steve: Exactly. I wanted to make sure I didn't forget to mention, because many people asked this. They were seeing the example I gave last week in a public setting and wondered about what's happening in a home setting. So, I mean, ARP spoofing is much less likely in a LAN. It's somewhat more possible in a wireless environment. But it is definitely the case that, if you're using a VPN, or you somehow have a persistent SSL connection, which is what a VPN would provide, or which is what a custom app running in the browser would create, then you really have nothing to worry about because a good VPN and/or SSL technology provides authentication of the endpoint and privacy so that no one in the middle has any opportunity to do anything bad to you. So the worst that ARP spoofing could do would be to keep you from getting a connection. But it would not be able to allow someone to intercept what you're doing.

Leo: Perfect. John Clayton in Billings, Montana reports that Astaro has upgraded their free home use licenses. We love Astaro. We talk about them all the time. Hi, Steve and Leo. Know that Astaro is one of our longest and most loyal advertisers on the show and thought your listeners might be interested in this news. For the longest time I had used Astaro on an old PC as my home firewall, using their free home user license. Unfortunately, with so many connected devices in the house I outgrew the 10 IP limit of the license - wow, he's got a lot of computers - and had to switch. I've never been nearly as satisfied with any other firewall solution as I was with Astaro.

Fast forward to yesterday, when Astaro announced it was raising the limit for the non-commercial home user license to 50 IP addresses. I guess this guy's not alone. This is more than enough to protect my home network and is likely sufficient even for a larger family with even more devices. This is truly generous of Astaro. The restricted license was partly to deter businesses from using it for free, and most of the community was only expecting 20 or 25. Really, it's true, you know, it's really honor system now because, you know, my business is less than 50 IP addresses, even with all the computers we have. I'm happy to say I'm back on Astaro. There's simply nothing else that can touch it as far as power, features, and ease of use. And now it's even more accessible for your listeners to run in their own homes. Always love the show. Keep up the good work. Well, that's nice. Thank you, Astaro, for doing that.

Steve: You know, Leo, I think I'm going to have to poke around at it. I haven't yet. I've just got so much going on and all that. But I've got just a regular cheesy consumer home router over on my cable connection, which I don't normally use for things. But I think I'm going to take a look at it.

Leo: A couple, you know, the easiest way to do it is VMware has an appliance, or an Astaro pre-installed appliance you just put on your system. Because it uses Linux. But you can put it on any beige box. It's easy enough to do. It's not a difficult thing to do.

Steve: Yeah. I've got a cute little - Soekris is the name of the company. They make beautiful little embedded PC appliances that are, like, multiple NICs. And they run UNIX and FreeBSD and so forth. So I think I set it up with FreeBSD which, as we know, is my UNIX of choice. But, you know, OpenBSD and NetBSD and all the other ones work, as well, so.

Leo: That's true. Any UNIX, any UNIX, yeah.

Steve: Yeah.

Leo: Finally, our last question, from Alan Goldstein in Franklin, Massachusetts, commenting once again on the broken browsers. Steve, I'm a SpinRite owner and a fan. It has saved me many times, including helping me get more than an extra year out of my Pentium 4 desktop. Oh, wow. That's a great return on my $90 investment. Great episode last week. It made me think that both Internet Explorer and Firefox should do more to clearly indicate if the connection is secure with https. In the short term my approach is to otherwise change all my more critical bookmarks to include https for those pages that support it, just so I won't forget, and I'll get a secure connection even without thinking about it. Perhaps we should suggest that someone in the know write a Firefox add-on that would highlight both the address bar and the status bar in green whenever you're securely connected. It's too easy to neglect looking for the https on every page. Top and bottom green bars would stand out and clearly show when you're not on a secure page, when there's no green bar. Unfortunately the padlock indicator just doesn't stand out sufficiently. Keep up the great work and the great podcasts. Alan.

Steve: Well, it's interesting. This has been an issue that the browser vendors have been aware of for a while. IE has a configurable setting in their security settings which says submit nonencrypted form data. And you can disable that, you can enable that, or you can tell it to prompt you. So the idea is that you could set it to prompt and so you would just be advised, if you clicked a button that was not secure, that you were about to submit nonsecure form data.

Leo: Yeah, I've seen that little box.

Steve: Right. Now, Firefox has a bunch of things under their sort of extra security settings. And they've got - it's five checkboxes. They've got one that says show a warning dialogue when, one, I am about to view an encrypted page. I'm not sure why you'd want that. But these are all turned off by default, by the way. So when I've about to view an encrypted page. Or, number two, I'm about to view a page that uses low grade encryption. Okay.

Number three, I leave an encrypted page for one that isn't encrypted. Now, that's useful because that would be an encrypted page, for example, that had maybe a button, a form submission that was going to take - that was not going to be secure. Except that the problem is, that would be popping up all the time because anytime you went to a nonencrypted page you'd get a warning. And so that's hard to have that one turned on. Number four is I submit information that's not encrypted. So that's certainly a useful one.

Or, five, I'm about to view an encrypted page that contains some unencrypted information. Now, that one's annoying because that's - you get that all the time from, like, an encrypted page which has other components on it. IE calls that "mixed security," where an encrypted page will have maybe just images or thumbnails or other things which are nonencrypted. Well, I guess that can be a problem, but I don't really see how that's a huge security problem.

So of those five on Firefox, really the one of I submit information that's not encrypted, I would say that's useful to turn on because it will just give you a warning if you're using a form, and you're about to - and this form data would be going over a nonencrypted connection. So it's important to know, though, that none of those prevent exploitation from the problem that an unencrypted page could be modified. Because someone in the middle could change the unencrypted page to send the form information securely to them, rather than to where you think it's going. So...

Leo: So you have to have a padlock. You have to have an encrypted page. And the form information has to be sent in encrypted form.

Steve: Correct.

Leo: Okay.

Steve: Correct.

Leo: The two. You need both.

Steve: Right. And so Leo, you know, you were unimpressed by this because probably I tried to give it to you without going through everything that we discussed last week. The problem is that, if you were completely vigilant, if you were never distracted, you were never in a hurry, you absolutely never logged on anywhere without making sure that the logon page was secure, then I agree, nonissue. But no one here can say that that's the way they use their computer. So...

Leo: I think I'm less - I don't think there's a lot of evidence that people are doing this. And it's not a trivial thing to do, this ARP spoofing. It's possible.

Steve: Correct.

Leo: But it's a theoretical possibility. But somebody would have to really, I mean, first of all they'd have to compromise a server somewhere.

Steve: Well, no.

Leo: Assuming that they've not compromised your LAN.

Steve: All they would have to do, and this was the example I gave, and it's what Moxie did, is simply go to any open WiFi hotspot. And that's, I mean, that's all it takes.

Leo: There are plenty of other dangers in an open WiFi hotspot; right?

Steve: Except that you're assuming that your logins are secure. You're assuming that when you're providing...

Leo: Well, that's foolish.

Steve: Yes.

Leo: That's just foolish.

Steve: When you're providing credit card information and, like, you're assuming that it's going to be secure. And so his point was...

Leo: Are there a lot of sites that are not secure in this regard?

Steve: It turns out that many financial sites are not. I used a bad example. And many of our listeners pointed out that PayPal, which I just used because it's so - it's common...

Leo: Because we hate them, yes, okay.

Steve: Because there's so many other dumb things they do. Well, they do not do this dumb thing. The form you use for logging into PayPal is secure. So a number of - a bunch of our listeners wrote in and said, well, Steve, PayPal was a bad example, but here's a good one. And so there's, like, lots of other examples of financial institutions where you log on on an insecure form.

Leo: Well, and that should be fixed. I mean, that's the place to go to fix that. Those people are morons if they have, I mean, what - that's nuts. Now, let me ask you a couple of questions. I wasn't here. I apologize. But I'm just looking at my Macintosh here, for instance. This is Safari. This is the box, it says ask before sending a nonsecure form from a secure website. So that box should be checked.

Steve: Correct.

Leo: Now, as long as I go, say, let's see, to my Amazon account here, and I'm looking at my Amazon account, and I see that it's an https, I'm safe; right? Because not only am I on an https, but this form, even if it's poorly coded, I'm going to get a warning if it says, hey, that's a nonsecure form.

Steve: Correct. The key is, I mean, we're assuming that Amazon knows how to get their - we're assuming that Amazon knows how to protect you, except that many companies are still not protecting the form where they ask for your data. And that needs to get fixed.

Leo: Right, but I would get that warning on that page; right? That's why I checked that box in Safari and in the other browsers, to say warn me if I'm sending a nonsecure form from a secure website.

Steve: Well, and remember we're not so much talking about catching Leo as catching my mom.

Leo: Yeah, yeah. But by default Firefox has that box checked. So the real issue is - would be an insecure page with like a bank page that's not a secure page.

Steve: Right. And the point I made last week, which Moxie made in his presentation, which I didn't say this week, is that most users don't put in https://www.

Leo: No, nobody does; right.

Steve: Exactly. So really the point was that we're relying on our browsers, that is to say, on the remote server, to switch us into and out of SSL as necessary. We often start on, you know, non-logged-in on regular pages. When we go to the login page, we're assuming that the remote site is going to take us to a secure page where we're going to do the things that need to be secure. And then it's going to take us back out of it. Because historically bringing up SSL connections, which required public key crypto, was an expensive, computationally expensive thing to do. And it's one thing for individual users to do it. But if all of those users concentrate on a single server, the server can quickly be brought down by just needing to negotiate SSL connections.

So the idea was that we're relying on the remote server to put our browser into and out of secure mode. But if that's the case, and somebody did insert themselves into our traffic stream, they could always filter out the s's" on the https's and then get the data back from us and create secure connections themselves to the remote server, but keep the link that we have apparently to the remote server not secure. And it is - it's been done as a proof of concept. It's as easy as somebody...

Leo: Let me ask you this to clarify this. I'm sorry you're going back through this again.

Steve: Sure, no. No, it's okay.

Leo: You're saying that they could spoof the padlock?

Steve: Yes. In fact...

Leo: So if I see a padlock on a page, and it says https, if there's a man-in-the-middle attack, that could be a lie.

Steve: Well, it was the case that, until Tuesday, that you could spoof a secure connection, so you were actually connected to somewhere else with all the browser's security up. One of the things that Moxie...

Leo: That was because of the white space issue?

Steve: Correct. One of the things that Moxie did...

Leo: That's why, by the way, Brian Krebs said don't use Windows to bank.

Steve: Yes. Exactly.

Leo: Because that's a Windows flaw.

Steve: Exactly.

Leo: Yeah. He suggests using a Live CD of Linux to bank. Or at the very bottom he says if you're a Mac user you're okay, too.

Steve: So one of the things that Moxie did that was kind of clever was that he changed the favicon on the fly.

Leo: Oh, that's clever.

Steve: To a padlock. And so you sort of saw the padlock. And again, no hardcore security guy is going to get caught out by this. But my mother wouldn't know the difference. She sees a padlock up there by her URL and goes, oh, that means secure. It's like, okay. But in this case it doesn't. So...

Leo: So what I want to tell my listeners on the radio show, who are basically your mom, is, well, I just went through all my financial institutions one by one through the bookmarks. They all show up as https.

Steve: Great.

Leo: So that's the first thing to do is make sure that they show up as https and you see the padlock, not in the browser icon, but in the corner of the browser window. And it should be locked. And then turn on the setting that says warn me if I am sending an insecure form on a secure site.

Steve: Yup.

Leo: That's on by default in most browsers. But if it's not, check, make sure. And then, when you get that popup, understand what's happening here, that there's a risk now that somebody could be capturing that data because it's an insecure form.

Steve: Yeah. I would say the simplest thing is make sure of the security of the form you're filling out. That is, if the form you're filling out, if where you're being asked for username and password, if that is secure, if that's got the proper padlock-y icons...

Leo: Ah, then you're okay.

Steve: Then really everything is okay.

Leo: So but you can't have an insecure form on a secure page, though; right?

Steve: You can, except that if the form is secure, then you got it from a remote website. That is, it wasn't edited. The form wasn't changed.

Leo: Ah, okay. And that's the key, okay.

Steve: And so, exactly, you don't - it's the form wasn't changed. If they don't want to - if they don't care about what you submit being secure, it's like, okay, well, I'm not sure that you have to care about it. But the danger is, if the form is not secure, the form that you received is not secure...

Leo: It could be [indiscernible].

Steve: It could have been changed.

Leo: Got it.

Steve: To remove the security of your submission. And again, someone hypervigilant would probably catch that, hopefully. But so many of us, I mean, logging into the sites we log into every day, it gets to be kind of routine. So it's easy just to miss it one time.

Leo: What would be prudent to do, what I just did, which is go through all my financial, the bookmarks that I use to go to my financial sites, and make sure that that landing page in each and every case is in https with a closed padlock, not as a favicon.

Steve: Yes.

Leo: And maybe do that once in a while.

Steve: Yes. And in fact that was one of tips that one of our listeners in this episode made. He said, hey, I've got all my shortcuts. I just went through them, and I just made sure I put s's on all of the URLs and then checked to make sure you could still use it. Because some sites will allow you to get to them either secure or not.

Leo: These bookmarks are just the default bookmarks from the site. I don't know if they have the "s" in there or not. But anyway, that's probably a prudent thing to do.

Steve: Yeah.

Leo: Yeah. Cool. All right. Well, that's good advice. I'm looking here, yeah, yeah, these all have s's. That's a good thing.

Steve: That's a good thing.

Leo: See, I wouldn't - I wouldn't say this should scare you away from using your browser. I thought it was - I was actually shocked that Brian Krebs went so far as to say don't use Windows to bank. That was a little bit of a shock to me. I mean, whoo. But, you know, I guess in a way you could justify that.

Steve: Well, I mean, and was he speaking only of this issue, that is, of the null...

Leo: He used as an example a couple of things we've talked about on the show before, the guy who had the trojan on there that he authenticated, but only authenticated once, and of course then all the other transactions were sniffed. They also used click fraud as an example and talked about the very widespread prevalence of click fraud, which is another kind of man-in-the-middle-like attack that is possible. So I don't think he mentioned ARP spoofing. But it's clear, he gave enough examples of what could go wrong.

Steve: Well, a man-in-the-middle attack, I mean, ARP spoofing is just one means for achieving a man-in-the-middle attack. So and of course click fraud is prevented by secure things that you click on. So as long as the pages are secure, I mean, really, Leo, what we ought to do is just drop http, that is, the nonsecure. We ought to just say, okay, it's time for us to just switch over to SSL.

Leo: That's a good point.

Steve: Because who, you know, when do you not want, I mean, when is having a secure connection a problem? Well, it's never a problem unless the server can't handle the SSL negotiations. But several things that have happened since it was invented, since SSL 1 happened, namely, the notion of persistent connections. So the browser maintains a connection to the server, means you're not constantly renegotiating SSL connections. That, and remember when we talked about the SSL protocol, the notion of caching your previously negotiated data also prevented you, as long as both ends agreed to use the recently used data, then you avoided all the overhead of doing it again. So many things have happened to lessen this burden. It would be really interesting to know from someone big, like, Adobe or Yahoo! or Hotmail or Google, Google with Gmail, where you've got the option of always using and forcing SSL, that's what we want from everyone. We just want our browsers to say, hey, keep me secure unless you absolutely can't. Or better yet, I only want SSL connections, period.

Leo: I'd like that. I would like to - I think we can campaign for that. All SSL, all the time. It's a very simple thing to do. Everybody's got enough horsepower now to do this. That's not the issue. Let's just all SSL all the time. The only people who wouldn't want to do that are people who don't want to pay for the certificate. And, look, if you come into my blog and it's not SSL, so what? Right?

Steve: That's a very good...

Leo: But anytime there's a login, SSL, all the time. Let's make that our campaign.

Steve: Yup.

Leo: Yeah. And then you could bank with Windows again. I think Microsoft should make that their campaign. Seems like they have a dog in this hunt. Boy, I mean, I was just - it was like, wow, that's a - he says use a Live Linux CD. See, now, there's something that nobody's going to do.

Steve: No, exactly.

Leo: Mom's not going to do that.

Steve: She's not going to shut down her machine and reboot with a CD. She'd go, huh? What, honey? What do you want me to do?

Leo: And even what we described is too hard. That's why ultimately comes down to the websites themselves that have to do it right.

Steve: Yup. I think someday we'll look back on these quaint days where ASCII text moved across the Internet...

Leo: Unencrypted.

Steve: ...unencrypted in little individual characters that anybody could sniff and capture. It's like, what were we thinking? How did we even survive those?

Leo: Somebody pointed out in the chatroom, let's get DNSSEC working first. Even that's not out there universally. And we know that needs to be done. It's incredible, incredible.

Steve, as always, a pleasure. Thank you so much. You can find the notes to this show and transcriptions and 16KB versions at Steve's website, GRC.com. That's of course the home of SpinRite, the world's finest hard-drive maintenance and recovery utility, a must-have at GRC.com. If you've got a question for future feedback episodes, GRC.com/feedback. And of course all those free programs like ShieldsUP! and Wizmo, it's all there. Gibson Research Corporation, that's the name, GRC.com. You can watch us do this show live every Wednesday. We do it at 2:00 p.m. Eastern, that's 11:00 a.m. Pacific, at live.twit.tv. 1800 UTC for those you living outside the U.S. And you know, Steve, I was in Dubai, and I met so many people who listen to this show and all the TWiT shows. But this show's very popular in the Middle East.

Steve: No kidding. How cool.

Leo: Yeah. A lot of listeners from Bahrain and Kuwait, Lebanon, Saudi Arabia, I mean, just all over the Middle East, who came to Dubai, a lot of them just to say hi, I listen. So that's really nice.

Steve: That's neat.

Leo: Yeah, we've got some great fans out there. You can of course subscribe to the podcast, if you're not already. You don't want to miss an episode. And really you should keep an archive. Steve does keep an archive of all the shows. We also have one at TWiT.tv. But you should have your own because it's frequently we will go, oh, well, you should listen to our ARP spoofing episode, you know, back a hundred episodes. So just subscribe at iTunes or your favorite podcatcher to Security Now!. And then you'll get every episode automatically.

Oh, one more thing. Steve, you won the Podcast Awards Best Tech Podcast I think last year.

Steve: Yeah.

Leo: And the podcast awards have come around again. Nominations are being accepted through the 19th. So a lot of the other hosts want to win, too. So I've decided not to take sides, but just to tell you, if you listen to this show, and you love this show, go to the Podcast Awards page, PodcastAwards.com, and nominate it. There are a number of categories. Technology, obviously, People's Choice. I don't think comedy. But, you know, whatever...

Steve: [Laughing] I hope not comedy.

Leo: Put the Giz Wiz in comedy. But pick a section and nominate your favorite TWiT shows. We would appreciate it. We just love to get on that, you know, have all the nominations filled with TWiT programs.

Steve: Yeah, we want TWiT. We want Leo's podcasts.

Leo: Security Now! would work for me. I'm very happy that you won last time. So that's all you have to do. You've got to the 19th. And then after the 19th, after all the nominations are in, it'll be a little easier. Then you go and you vote for your favorite. Thank you, Steve.

Steve: Always a pleasure, Leo. And next week we've got John Graham-Cumming is going to join us. He did a great presentation on, well, I know this is where you and I go back and forth. I would title the next week's show "JavaScript: Just Say No."

Leo: [Laughing] Good luck on that one, Steve.

Steve: I know, I know. But he's going to explain exactly why.

Leo: If you were scared last week, wait'll next week. All right, Steve. We'll see you then. That'll be a lot of fun.

Steve: Yes, it will.