Listener Feedback #105

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-275.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-275-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode 275, recorded November 17, 2010: Your questions, Steve's answers, #105.

It's time for Security Now!, the show that covers your privacy, your security, your online safety. And the man, the myth, the legend, Steve Gibson is here from GRC.com. Steve is the guy who discovered the first spyware, coined the term "spyware," wrote the first antispyware program. He long since handed that off. He's probably thanking his lucky stars right now.

Steve Gibson: I'm laughing that I'm the myth. I'm not quite sure - the man, the myth, the legend.

Leo: The man, the myth, the legend.

Steve: Every time you say "myth," I think...

Leo: Hmm, what is the myth? No, there's myths about you, Steve.

Steve: Okay.

Leo: Many people think you wear steel underwear. That's not true.

Steve: No, it's not.

Leo: No.

Steve: Never has been true.

Leo: There's myths about Steve.

Steve: Not even reinforced in any way.

Leo: Steve is also the guy who wrote SpinRite, the world's best hard drive utility out there.

Steve: That's not a myth.

Leo: That's no myth.

Steve: That's definitely true.

Leo: That's all at GRC.com. And of course we debuted last week your free DNS Benchmark program.

Steve: Yes. It's been a big hit. I have a couple little comments in the errata section today to share, just little brief things. We've been seeing about 1,100 downloads a day, so it took off...

Leo: Wow, that's great.

Steve: ...really nicely, yeah.

Leo: Not as big as Firesheep.

Steve: Nope.

Leo: And you could have written Firesheep, but I'm glad you didn't.

Steve: I'm glad for that, too.

Leo: Yeah, this is a Q&A episode, so we have lots of questions, lots of answers. We also have security news, security updates and so forth. All right, Steve. I see some errata, some stuff to cover here.

Steve: Yeah. Well, we don't have too much in the update category, though I guess when weighed by the megabyte we actually do.

Leo: [Laughing] Whoo, yeah, this is a big patch.

Steve: Oh, my goodness. I love that it's the "Double Oh Seven" (007) patch. It's the seventh patch of the year for Apple's OS X. And in my case, 572MB. So over half a gig.

Leo: Yeah, mine was like that, too. I couldn't believe it.

Steve: Yeah. Well, 130 different security vulnerabilities. Although, in fairness, 55 of them were in Flash.

Leo: You know, it's weird, but I guess that Apple has taken it on itself to patch Flash, as well.

Steve: Yes. Now, if people had patched Flash before this manually, then this would have been redundant. But among this was a new copy of Flash Player that brings OS X current with where Adobe is with Flash. So it was a huge update. The Apple page that enumerates all these things just scrolls on and on forever and ever. It brings OS X up to 10.6.5, or 10.5.8, depending upon whether you're back at the .5 or .6. So, and that was now a week ago. So I would imagine most Mac people have already encountered this.

For me, I turned this one little mini on that I use for our podcasts only once a week. And I fired it up, and I knew this was happening because I had already encountered the news about this. And it's like, oh, here it is, baby. And so luckily I turned it on a few hours ago so that it was able to download this thing and update itself and get going. So also there were fixes to QuickTime, Time Machine, Safari, just pretty much across the board, 130 different vulnerabilities patched. So the 007, seventh patch of the year. Probably the last big one, I would think, given here we are mid-November.

And then the other important one was that Adobe has fixed their - this was the patch to v9 which we knew they were going to be coming out with. We've been following this for the last couple weeks as Adobe's been trying to do this out-of-cycle patch to get Reader updated. So this is the v9 of Reader, which was lagging behind their fixes for v10 and of Flash, all of which they had already patched. So if people were back on v9, then just yesterday from when we're recording this, so let's see, we're recording this on the 17th, so on the 16th they released their v9 catch-up. So they're now current with this zero-day vulnerability which had been seeing a lot of exploitation.

In news, the U.K. has backpedaled a little bit from what we discussed, it was in the last couple weeks we talked about the U.K. passing the Digital Economy Act, which among other things, remember that it had that three strikes, then you're disconnected from the

Internet provision. Well, now they're saying, whoa, whoa, whoa, wait a minute, you know, yes, that's in the law. But I guess it caused a lot of furor over there. So they're saying that they're not going to be disconnecting people, that technically there's a provision for that, but they really don't expect to be actively pursuing that.

Leo: Good, good, good, good.

Steve: Yes, that is good news. The still-unpatched zero-day IE6 and 7 flaw, we've talked about this weekly now because last week we mentioned it had made itself into some of the most popular hacker kits, and it's seeing more widespread use. I picked up a little note that it had appeared, that the exploit for this had appeared on Amnesty International's Hong Kong website, such that people who went to the Hong Kong website of Amnesty International using IE6 or 7 would get themselves infected because there's no patch for this yet. Still no word from Microsoft about when they're going to get this done. This came out just after the second Tuesday of November, so it caught Microsoft off-guard. And they're still saying they don't feel this is a big enough deal to do an out-of-cycle patch. So we'll see if this gets worse before the second Tuesday of December, when let's hope they get this thing fixed.

A little bit of news is that Sweden is considering legislation to require ISPs to retain all of their customers' email and cell phone text messages for six months.

Leo: What?

Steve: I know. We keep seeing these data retention legislation threats floating around. And I hope that the legislators understand the burden that this creates. I mean, it's easy for them to sit in their council chambers and say, yeah, let's have everyone - let's require everyone retain everything for six months. But, I mean, that's a huge technological burden to impose on, like, just out of the blue on a carrier, which is what an ISP is currently.

They may be doing some filtering. We know, for example, that ISPs block various ports. For example, I know that the cable modem provider here in Southern California, it blocks the traditional dangerous Windows ports 137, 138, 139, and 445 that were the file/printer-sharing ports. They also block 25, which is SMTP, to prevent, to some degree, bad things from being able to be used for spamming, botnets and so forth for spamming. But it's very simple from a technology standpoint to block traffic on a port. I mean, it takes nothing.

It's a whole different scale of obligation to ask an ISP to intercept, which is what they have to do, and record all of the email transactions, and in this case cell phone text messages, of all their customers, to hold them for six months, and then somehow age them and let them go away after six months. So this is, again, let's just hope this doesn't happen. This would be a bad precedent to have. I mean, bad not as much from a privacy standpoint, I mean, that's a problem, too. But mostly just it's a phenomenal, from a technology standpoint, a phenomenal change to impose on ISPs. So I just - I hope this just doesn't happen.

Leo: Didn't the FBI put those Carnivore, or propose to put those Carnivore boxes in?

And as I - the Carnivore did the collection and aggregation itself; right? It was kind of less of a burden on the ISP. They just had to pipe stuff through it.

Steve: Well, and it wasn't doing everything. So it was a tap where they were able to say, we want to start monitoring this particular customer. And so it would filter out traffic to a given customer and feed it off. So, I mean, and so that's being done in real time. It's being stored elsewhere. It's not saying to the ISP, we want you to record everything for six months so that we can later come back to you and say, oh, we'd like all the email of this particular person. So, yeah, it's a whole different scope of obligation, so [shuddering]. I think that the legislators get very cavalier about this idea. It's like, well, Google is indexing the whole Internet, so why can't you store all the email from all your customers? And it's just like, wow, there's no infrastructure in place for that. That's a huge, huge deal. So let's hope it doesn't happen.

One million Chinese cell phones have been infected by something that they're calling the "zombie virus."

Leo: Oh, great.

Steve: It masquerades as an antivirus application, which users have installed. It then propagates by text-messaging links to itself to everyone in the infected phone's phone book. So this thing spreads like wildfire. It is currently costing Chinese users, because it sends texts to premium text messaging, it's costing Chinese users an estimated $300,000 U.S., equivalent in yuan. $300,000 U.S. per day...

Leo: Crikey.

Steve: ...is the cost. And so it's just sort of a little cautionary note. I've seen some commentators talking about this, saying here's - it's like, yes, that's happening over in China, so it's not affecting our listeners in the U.K. and Australia and in the U.S. and so forth. But this is bound to happen. This is going - it's not a matter of if, but when. And so our listeners who are security conscious, I would just say when you're looking at your new phone, and you're thinking, wow, isn't this cool, this is a computer, and you're looking at the application stores which are available, ask yourself, do I really need this, and how long has it been around?

We know that new things tend to cause more trouble than old, proven things. So, sure, it's fun to download all these toys onto your phone. But with every one of them, they're taking up space, they're in that phone's ecosystem. And if they're brand new, there just isn't any way for everyone, for the people who are hopefully vetting these to some degree, to actually know what this can do. So just use caution. Just use your best judgment, I would say.

Leo: It's just a matter of time before we get something like this.

Steve: It's going to happen. I mean, we're seeing bits of this happening. Applications are being taken away that are found to be behaving badly. They haven't done anything like

this yet, but they certainly can. So it's, as you say, Leo, it's a matter of time. I would hope that our listeners won't get bitten badly by this. And this particular thing also sends all the SIM card data from the phone, texts it back to some server somewhere of the people who created it, basically allowing them to take over your phone.

Leo: Did this get installed, I mean, that's a lot of phones that got on it. Did it get installed by people downloading a rogue app?

Steve: Yes. It started as - it's masquerading as an antivirus.

Leo: Oh, that's why it got so many people.

Steve: Yup. They installed it, thinking, oh, this will be really good. And then it propagates by sending links to itself to people that they know.

Leo: Oh, here's an antivirus you ought to have for your phone.

Steve: Exactly.

Leo: One million people is a lot of people. I mean...

Steve: Coming from someone you know. Yeah, it's a huge infestation.

Leo: It's huge, yeah.

Steve: Meanwhile, Symantec has been very patiently continuing to reverse-engineer the Stuxnet worm. And finally we're beginning to see some really interesting data. I've resisted drawing any sweeping conclusions because there were lots of things being said early on, before there was really ever any justification for it, from a standpoint of what the Stuxnet worm does. Well, now we have enough specific information that you really can begin to say with a reasonable degree of certainty that this was targeted at Iranian nuclear enrichment. It really looks that way. Symantec has...

Leo: But it did have to affect the Siemens equipment; right?

Steve: Yes. Well, even more specifically, it turns out that - and it just takes time. It takes time to reverse-engineer the code. They don't have the source. They're looking at the object code which has been disassembled back into the assembly language. They have to, of course, it's not just Windows, but it's the specific Siemens process control technology. So arguably, or I'm sure it's the case that Symantec has experts on Windows and Mac, but probably not on Siemens process control hardware. So they had to reverse-engineer that, figure out what it was doing.

So now they know, for example, that it targets something called "frequency converter drives" which are power supplies, crystal-controlled, digital-controlled power supplies whose frequencies can be set. And the frequency of the power supply drives process control motors. It intercepts commands to vary the speed of these motors wildly, but only intermittently. And only if this thing, now that we see how the code works, only if it was in a plant's network, where it found at least 33 frequency converter drives made specifically by manufacturers Fararo Paya in Tehran or Vacon in Finland would it come to life.

So, again, it's incredibly narrowly targeted. And it only targets frequency drives from those two companies that are running at speeds between 807 Hz and 1210 Hz, which are the speeds that the uranium enrichment centrifuges run at. So it looks like it's absolutely certain, I mean, as much as anyone could be, that somebody who really knew what they were doing, I mean, this is not hackers, random script-kiddy-type people. I mean, this raises the bar and lends further credibility to the people who are stating that this really looks like state-level actors, I mean, government-type empowered people were in fact targeting Iran's nuclear enrichment facility.

Leo: Wow.

Steve: And it was also noted independently that I think they had - I remember a number like 4,300 centrifuges. And for an unspecified reason, about a third of those went offline. They were taken offline. And so as I understand it, the way this worm infecting the process control of the enrichment process, it would cause these centrifuges to appear to be malfunctioning. So, like, they would bizarrely run at very wildly different speeds, like as low as two cycles, which would basically shut it down. It would just sort of stop. And then it would spin up to a much higher speed than it's supposed to go. And so somebody looking at this would think, okay, this is broken, and turn it off.

And apparently about a third of these 4,300 total centrifuges were taken offline during this period of time. So it looks like it was effective, unfortunately. Now, of course, it's all public knowledge, and it's been unearthed. So I'm sure that they've cleaned this off their systems and are back to work. But that does look like, I mean, with something this specific, I think it's clear to be able to say now that's what Stuxnet was about, was it was targeting that specifically.

Leo: That's pretty amazing. I mean, and what great detective work, too.

Steve: Yeah, yeah. And a lot of work to pull that kind of thing off. Oh, I had three little notes. When I pulled today's mailbag to get caught up on all the submissions that people have had, it wasn't till afterwards that I realized, wow, I didn't see any feedback about the DNS Benchmark. And then I looked again, and I realized that something had interrupted the hundreds of pieces of email that I was downloading, and stopped like a week ago, just before last week's podcast, that is, stopped in terms of the dates of the email submissions. And so I grabbed a bunch more, like, just half an hour ago, and there was another couple hundred. And there were lots of people who were talking about DNS stuff. But I had already found three little pieces of feedback. So I just wanted to share them as errata, to basically encourage people to take a look at the free, completely free Benchmark that we discussed last week, if you hadn't.

One guy wrote, the subject was "DNS Benchmark - U.K." And he said, "In a word,

marvelous. I'm going to inform everyone I know about this. Configured my home router and all computers to use the same Primary and Secondary resolvers after running several custom benchmarks. Amazing difference. Without this tool, I thought I had a marvelous broadband connection. It has been an eye-opening experience, and page-refreshes and loads are visually better without any need for precise measurement. It made such a difference. Thank you, Steve."

Someone else wrote, "163K worth of digital voodoo magic. Loving DNS Benchmark, enabled me to optimize my network further. Found a lot faster DNS servers than OpenDNS." And finally, this third guy said - his subject was "DNS Benchmark success." And he said, "Hi, Steve. I often listen to Security Now! and always enjoy it when I do. Today I watched the live podcast and heard DNS Benchmark mentioned, so I gave it a whirl on my Linux machine at home. Home is Dublin, Ireland. The tool found 40 servers faster than my..." current default, which was 8.8.8.8, so that he was using Google's DNS.

Leo: But of course Google in the states, probably; right?

Steve: Yeah, although I would expect Google to be...

Leo: They have a big Irish facility.

Steve: Yeah, exactly. Anyway, he said, so 40 servers faster than that, "...and the fastest was, surprisingly, my ISP's. I say 'surprisingly' because I moved away from using their DNS server last year because it was taking one to two seconds to resolve many IP addresses. Obviously they have fixed their problems. Thanks for an interesting and useful tool. I work in an electronics engineering company. Pretty much everyone in the department I work for has configured their machines to use some other public DNS server, not the company one. It'll be interesting to see who has chosen well. Cheers."

Leo: Very good.

Steve: And then I wanted to make a note about a beta of Google Chrome which caught my eye, just also in errata. This is still beta, so it's not in the normal production Chrome stream. But Chrome is adding, that is, Google's Chrome browser, now in the beta stream, has their own PDF document display, which I thought was really interesting. In their posting they said, "With every Google Chrome release, we hope to bring new features and improvements that will make your life on the web speedier, simpler, and more secure. Today, we're excited to introduce the integrated PDF viewer to the beta channel. PDF is a popular file format that's used for delivering documents on the web (such as the IRS W-4 tax form)" - that's sort of an odd example, but that's what...

Leo: Well, it's something probably people use more than anything else.

Steve: Yeah. "To open a PDF document, you'd typically need to install additional software or a browser plug-in in order to view it in a web browser. With the integrated Chrome PDF viewer now available in Chrome's beta, you can open a PDF document in Chrome without installing additional software."

Leo: That's awesome.

Steve: Yes.

Leo: You know, Google does that in Gmail. They have a viewer built into Gmail. I'm sure it's the same code.

Steve: Yes. "The PDF document will load as quickly and seamlessly as a normal web page in the browser. Just like we do with web pages viewed in Chrome, we've built in an additional layer of security called the "sandbox" around the Chrome PDF viewer to help protect you from malware and security attacks that are targeted at PDF files. For now, the Chrome PDF viewer is available only in the beta channel, but we look forward to adding more polish and features, as well as making it widely available in the stable channel soon."

Leo: That'd be great.

Steve: So I just think that's definitely a hats-off.

Leo: Yeah.

Steve: And I know that you're now using Chrome as your main production browser.

Leo: All the time, yeah. On Mac and Windows. I just love it.

Steve: Yeah. I'm still - I'm so hooked on the Firefox ecosystem, with so many add-ons that I'm liking, like hierarchical tabs down the left-hand side and other goodies. But I also like, I have to say, I like what Google's doing with Chrome.

Leo: Chrome has Flash built in, too.

Steve: Yes.

Leo: But Adobe, I don't know how much Adobe likes what Google's doing with Chrome. I mean...

Steve: Yeah, they're basically commandeering their plug-ins and building them into their browser, and then sandboxing them to make them safe.

Leo: Smart. So of course Flash is Adobe's code. But is the PDF Reader Adobe's code, or is it Google's code? Did they say?

Steve: Good question. No, they didn't say. That's the entire content of their posting.

Leo: Interesting. I bet it's their code, since they have a - but who knows. Maybe they're licensing something, I don't know.

Steve: Apple, we know that Apple does their own rendering of PDFs; right?

Leo: Yeah, because PDF is a standard. An open standard that Adobe - I don't know if Adobe owns it, or if they're giving it away. I'm not sure exactly what the deal is with that.

Steve: Yeah. I did have an interesting bit of feedback about SpinRite. This is a little longer than usual, but there's an interesting lesson here that occurred to me after I read this. And this just came in on November 8th from Paul Oaten. He said, "Hi, Steve. I'm an IT...." Oh, and the subject was "SpinRite saves the linguini." He said, "Hi, Steve. I'm an IT guy in business for myself, listener since Episode 1 of Security Now!, and SpinRite owner. As well as doing break fix jobs" - and I don't know, he doesn't spell it like "brake" as in fixing someone's brakes, but I'm not sure what a break fix job is - "I'm also a web designer. Six months ago I took on a new website client. She's a wonderful Italian chef with a passion for cooking and food writing, but not very tech-aware, and she wanted to share all this with the world via a new blog. I duly created a custom site for her, and it now has over 400 posts, all of which are well-written, entertaining to read, and extremely practical. She's using the blog as a promotional tool for her upcoming book."

Leo: I hope he gives the address of this. I want to go now.

Steve: Well, we originally had it, but she objected to being portrayed in this light, so we have removed it.

Leo: All right.

Steve: So he says, "She's writing a blog as a promotional tool for her upcoming book. Trying to attract a publisher can be enormously difficult. When I visited her house, I discovered that she was attempting to write blog articles on a very old and tired laptop which really was not up to the job. A deal was struck, and I provided her with a brand-spanking-new HP G72 laptop with a nice, big, bright screen and large 320GB hard drive. As I was setting up the new laptop in my office, I noticed my copy of SpinRite sitting on a desk. I always keep it handy nearby. It occurred to me that I ought to run SpinRite on the laptop before delivery. Sadly, however, time did not permit this. And also, hey, what could possibly be wrong with a brand new drive; right?

"About two weeks after delivering and installing the laptop to a happy customer, I got a

phone call from a very distressed cook, who was now not able to boot the laptop, and who was staring at a message on the screen indicating imminent drive failure."

Leo: Uh-oh.

Steve: "Mamma Mia," he says, "were the words used over and over when I told her that potentially all data, i.e., chapters of her new book, recipes, unposted blog articles, et cetera, might have been lost from the drive. I collected the laptop, immediately removed the drive, and plugged it into a Linux box to see if I could access any data and hopefully recover it."

Leo: Oh, she's lucky she knows this guy, I've got to say.

Steve: Yeah, this guy knows what he's doing. He says, "A worrying Linux message told me that there was no way to mount the partition, so I plugged the drive back into the laptop and reached for my SpinRite disk. My heart beat a little faster when the laptop refused to boot from the SpinRite disk. I then loaded SpinRite onto a bootable Flash drive and eventually got it running. SpinRite recognized the partition and got to work. Almost immediately the DynaStat screen appeared. Not good. After two days, SpinRite had completed 0.983 percent of the drive."

Leo: Now, does it work on the - go from the inner to the outer? I mean, is it going in order?

Steve: Yeah, it starts at the beginning.

Leo: So it makes sense it would hit system files early on.

Steve: Yes, yes.

Leo: So I bet you this happens a lot because, when it's not booting, people then go, oh, it's the system.

Steve: Yeah.

Leo: So I bet you that's when you often get those long delays.

Steve: Right. And it means that it often gets the work done that it needs to...

Leo: Right away, yeah.

Steve: ...quicker rather than later. So he says, "Not good. I let SpinRite run a few more days, all the while fending off frantic inquiries from the Italian cook by telling her that if anything could recover the data, SpinRite could. After seven days straight, I looked at progress."

Leo: Oh, she must have been crazed by this time.

Steve: Oh, yeah. "Seven percent of the drive had been processed..."

Leo: Oh, it'll only take three months to get the...

Steve: Yeah, "...and there had been multiple unrecovered sectors. I decided to halt the process and try one last time to access the drive via my Linux box. Guess what? Despite the many unrecoverable sectors, I was able to recover all the documents now, and most of the photos, from the drive. Fantastico," he says. "The drive is now being replaced by HP under warranty. My chef has her data back. And I'll be getting a fancy Italian cake as a reward."

Leo: I'd want more than that. I want a whole meal.

Steve: Yeah. He says, "Thanks, Steve."

Leo: I've been looking at her pictures.

Steve: He says, "Thanks, Steve, for a great podcast that never fails to impress me. Ciao. Paul Oaten in Bath, U.K."

Leo: Oh, he's in the U.K., okay.

Steve: And then also, "P.S.: It's great to know, when the aliens come, the U.S. government will be able to call Mr. Gibson to deploy his big green laser as a last line of defense." Now, okay. Given - the reason I've shared this story is I'm suspicious. The drive was working probably just perfectly. And it was serious, something really bad happened to it. I'm suspicious that this thing got dropped.

Leo: Yeah.

Steve: That it was - maybe there were a little too many things on the chef's cutting board, and the laptop got dropped on the floor, or it fell over, or something happened to it. Because, I mean, for it to have gone from just fine and brand new, I mean, yes, it's the case that a drive can be flaky from day one. But this just - it's too suspicious that it took seven days to get 7 percent; SpinRite was finding all these problems after two weeks. I just think that this thing got some abuse, and SpinRite did everything it could.

Luckily, it was enough to get the critical management portions of the partition recovered so that he was able to get these files off, which probably nothing else would have been able to do. But still, you see something like this, and you think, okay, this got dropped. And in that case, you're doing the best you can to get anything from a drive that's in that kind of shape.

Leo: Right. He was very fortunate.

Steve: Yeah.

Leo: All right, Steve. I've got questions. I know you've got answers.

Steve: We've got some great feedback from our listeners. So let's do it.

Leo: Let's get to it. Starting with Question #1 from Jon H. in Excelsior, Minnesota. He wonders about mixed security. Steve, I'm wondering if you can comment on the security implications - you see this warning a lot - of mixing secure and insecure elements on a web page. Obviously, fully secure would be best. But is it reasonable to send most or part of the content securely, but then send image content in the clear? Or is there no way to do this that doesn't compromise the session cookie? Your podcast is a great resource and has clarified numerous details of encryption techniques, best practices, and vulnerabilities. Thank you both for the years of great content so far. I can look forward to hearing more. Best regards, Jon. So we see this in the browser. It'll say, just wanted to warn you this page has a mix of insecure and secure content. What does that mean?

Steve: Yeah. What that's telling us is that the page itself, the base page was secure. And some elements of the page which the browser then fetched were not. I don't think you get that warning if the base page is not secure, but some of the things that are on the page are secure.

Leo: Right.

Steve: I don't think it works the other way. I think it's only saying that this page has said that it wants its contents to be secure. But remember that the way the browser works is we first load the base page. That contains the HTML, which then has references to other assets, like images and other chunks of text, the cascading style sheet, other things which the browser then separately goes out and fetches in order to assemble the whole page. So the idea is that, if those references refer to insecure things, then what Jon is asking is, is that a big deal?

Well, we know from this whole Firesheep escapade that what browsers are often doing, what websites are doing with the browsers, is not keeping the session cookie secure. So the other way cookies operate is they're sent by domain. So say that we're getting a page from Amazon.com, as a site that is typical of having many other assets that are being loaded on the page, all those little pictures and chunks of menus and things that Amazon is loading. So the main page comes from Amazon.com and is secure. And with

your logon information, after logging onto the server, there would be a cookie which is always being sent, every time your browser notices that it is requesting something from the Amazon.com domain.

So the problem with insecure pieces, which might also be coming from Amazon.com, is that when your browser asks for those, it will send the session cookie. It'll send whatever cookies it has from Amazon.com unless they were marked as secure. So cookies can be marked as "this is a secure cookie, only return this to the domain if it's over a secure connection." So if somebody were deliberately trying to, for example, minimize their use of SSL, then they could mix the security of the page if they were careful to make sure that the credentials were only being exchanged over a secure connection. And that would mean flagging those credentials as requiring security, that is, flagging that cookie as only send this over a secure connection.

The problem, however, is that then users of the site that had carefully and deliberately made itself sort of optimized so that the page was being secure, the credentials were being secure, but other assets were not, were deliberately being insecure, every time the user brought up such a page, they'd be getting a warning from the browser saying, oh, warning, this page has mixed security content. So that would be concerning the user that maybe this is a bad thing, when it fact it was deliberate, and it was being safe.

So it would be nice, and there is no such protocol for this, but it would be nice if there were a way for a page to say to the browser, like with a response header, in the same way that the server sends a cookie to the browser, if there were a response header that said, hey, this is a deliberately mixed content page, don't bother anybody about that. Now, that's a safe thing to do because you could require that that header only be honored if it was being received over an SSL connection. So you wouldn't have to worry about it being injected or inserted by a man-in-the-middle attack over a nonsecure page. You could say, only if this page comes to us via SSL securely, and if there's this extra flag in there saying this is deliberately mixed content, don't bother the user about it, then I could see how that could be a nice addition to HTML, or the HTTP protocol, that would allow this kind of optimization, but wouldn't be sending off warning messages all the time.

Leo: All right. Question #2. You never know [laughing]. I never know if there's more. Mark Cyrulik in Oshkosh, Wisconsin wonders about network masking on networks. Whatever that is. Steve, I lived in an apartment complex that gave us free Internet while we lived there. It's all past tense. I guess he's moved. They ran the switches and the connection, and all we had to do was plug our devices into a wall jack. In trying to share music with my roommate, however, we ran into a lot of problems because the admin had done something I'd never seen before. He had set up the DHCP server to give out IP info as such:

IP Address: 10.1.3.24

Subnet mask: 255.255.255.254

DNS: 10.1.3.1

Gateway: 10.1.3.1

So the gateway and the DNS were the same. What I found very interesting was that he had set up the subnet mask in such a way that your computer thought it was the

only computer on the network. So 255.255.255.254 allows for one IP address. And I was not even able to see any other machine on the network. I know you mentioned in 272 that Starbucks could enable WPA2 as a partial interim solution to solving the Firesheep problem. I'm wondering whether a solution like the one above could also help to solve that problem. That's interesting. I never thought of that. So the subnet mask is set to allow but one IP address.

Steve: Yeah. Actually two because...

Leo: Yeah, you'd have to have two.

Steve: Yeah. What this is doing, it was an interesting configuration. So imagine an apartment complex where they're giving you free Internet access. And as we know, a 10-dot network - so it's behind its own NAT router, there's a NAT router somewhere, probably a big one, in the manager's office somewhere, which is basically creating a 10-dot network. And we know that that's 16 million IPs because the 10 is the first eight bits of the IP, and then the other 24 can be anything. The first eight of the 32-bit IP have to be 10. Then the next 24 bits can be anything. So that's 16 million IPs. So of course there aren't 16 million apartments, nor 16 million connections.

But so what they did was, if they simply set up a big LAN with a normal 10-dot network, there would be this problem that individual connections in different rooms and different apartments in this apartment complex were on the same 10-dot network. So they could see each other, they could ping each other, and there was some connectivity there. Now, probably they were using a switch rather than a hub.

Leo: A managed switch, probably; right? I mean, you'd need some intelligence here.

Steve: Yeah. Well, so the point is, if you just did a packet sniff on your connection, you probably were not seeing everybody else's traffic. But you'd be seeing their ARP requests, which are broadcast, and a switch inherently broadcasts everything. So you would see other machines on the network announcing themselves. And with a little bit of cleverness you could get other IPs that other people were using. You could play ARP games. I mean, there are things you could do.

So what this particular installation did was interesting. They, instead of - oh, and I should mention that on a normal 10-dot network, your subnet mask would be 255.0.0.0, meaning that the 255 portion of the subnet mask specifies the network, the so-called network number, which is 10. And those three zeroes, the 0.0.0, say that all the other bits are variable within this 10-dot network. Well, what this subnet mask does in this particular apartment complex is it's all ones except a zero at the very end, meaning that essentially every connection in the apartment complex sees itself on its own network. It says, only my IP - and technically there's one other IP because the last bit could be a zero or a one, but probably they were always zero - essentially, only my IP is on this network. So things like pinging other IPs would not work because they would, if you tried to - normally, if you ping another IP in your own LAN, then that packet is sent to the MAC address of that IP. And if you're pinging an IP not on your LAN, then it's sent to the MAC address of the gateway.

Well, what this apartment complex cleverly did was they set a subnet mask that said there are no other IPs on this network. So everything, if you sent anything to anywhere, it's going to go to the gateway. So what that does is create some isolation. Which I think is really very clever. It's an interesting way of taking a large private network, which a lot of untrusted people are sharing, and allowing them, dividing this private network up so that it creates interperson privacy to a much greater degree than you would normally have. All that said, he then asks, what does this do for, like, Firesheep and the Starbucks open network hotspot example? And unfortunately, not much, because wireless is always like a hub. And that's one of the problems is that...

Leo: Right, right, right.

Steve: ...when you broadcast anything, everybody can receive it. So this solution that the apartment complex used works because it's essentially created a very - there's a notion in LANs known as the broadcast domain, that is, when you broadcast, for example, ARP, an ARP request for, hey, who has this IP address, it's sent out to the broadcast IP of the network, which in this case would be - that's where the other IP is. It's like all ones in the IP. So whereas the IP, for example, was 10.1.3.24, the broadcast would be 10.1.3.25. So even ARP broadcasts would be constrained within these little individual networks. Not so in the case of using this approach on a wireless hotspot because you could still receive everything.

Now, it would be trickier to impersonate a person because you'd have to be - you're not all on the same network. So you're on individual little networks. But it does not provide you the same level of isolation that this does in the apartment complex, which actually is a very clever solution.

Leo: But it has to be hard-wired to make any sense.

Steve: It's got to be switched, and that's the deal, is a switch.

Leo: That's what a switch does.

Steve: Yes. A switch isolates so that you're only providing traffic that is intended for any of the devices connected to that physical port. And the switch itself learns which MAC addresses are connected to that physical port. But if you were to use this system in the apartment complex on a hub, then you would see everybody else's traffic, even though they're all on their own little individual itty-bitty networks. But still, very clever.

Leo: Yeah. That really is the key to the whole idea of managed switches is to isolate traffic.

Steve: Right.

Leo: But most hotels and other areas don't want to spend the money because

switches aren't cheap.

Steve: Switches are much more expensive, yes, especially in a big, like in a large complex.

Leo: Yeah. And it's a VLAN. It's a Virtual LAN. Chris in London, United Kingdom, was shocked - shocked - by the WPA key setup. Steve, I was amazed by your description of WPA's initial key exchange on the current Security Now! podcast. Diffie-Hellman key exchange has existed for the best part of 30 years and is a straightforward solution to this problem. Why don't they use Diffie-Hellman?

Steve: Okay. So what Chris was concerned about was when I explained that the reason that everybody sharing the same key still was a problem, I mean, it would encrypt connections, but if you had an attacker who was listening in, they would know everything that either party knew and be able to reconstruct a client's individual session key themselves. Now, Diffie-Hellman key exchange, which we have covered in the past on this podcast, is a very clever means for achieving this without the vulnerability of anybody listening in. So the way Diffie-Hellman works, just as a brief reminder, is you take some number and raise it to a power. And then the other end takes the same number and raises it to a different power. They exchange this intermediate result, and then they raise that to another power. So the idea being that mathematically a given number, say X raised to the power of A, raised to the power of B, is mathematically the same as X raised to the power of B, raised to the power of A. That is, the order in which you do these power functions is commutative. It doesn't matter which way it's done.

So what that means is that each end is able to come up with a random number as their own power, raise the common number to that, exchange the intermediate result, which is all done modulus some other number - and that's the key, it's within what's called a "field" - and then raise the exchanged intermediate again to their random number. And that's a way for them both to get the same result. And somebody who is watching the conversation has no help here. There's nothing they can do. They can see these intermediate results go by, but that doesn't help them in order to - it doesn't help them because there's no way from the result of this number being raised to a power modulus another number, that they're able to determine what's going on inside of either endpoint.

Well, the reason this wasn't done is, as you can tell, it's complicated. Also, these numbers have to be big. The numbers that are being used, the random numbers that are chosen as the powers, have to be on the order of a hundred digits long.

Leo: Yikes.

Steve: Yes. So you're raising something to a power that's a hundred digits long. Which is to say you're multiplying it by itself a huge number of times. And the modulus, to be effective, has to be a prime number that's about 300 digits long. The point is, this is public key technology. And the one thing that we know is public key crypto is slow. It's processor intensive.

So the reason the WPA designers, the people who were replacing the very badly broken WEP encryption, could not use public key technology is they had to have this stuff still

able to run on much lower-powered hardware. In order to make WPA practical to sort of upgrade low-end hardware, they had to use only symmetric crypto. They could not use asymmetric crypto. They could not use public key crypto, which is what Diffie-Hellman key exchange essentially is. And that's the cost of public key crypto is it uses - it has to use really long keys in order to obtain its security. And that means that it's going to - you only have to do it once. For example, once these endpoints exchanged their keys in a normal Diffie-Hellman exchange, now they've got something that they can use for all kinds of other purposes during communication. But you've got to do it once.

And the designers of WPA protocol just said, wow, we'd love to use it. It was developed in '77. The patent was issued in 1980. So that's 30 years ago. So the patent expired 17 years after that, so in 1997. So it was in the public domain, that is, Diffie-Hellman key exchange was. But even so, it's just too computationally burdensome to be able to retrofit older hardware and have it work. So they had to use something. Which really the only problem of using it is that there is this potential for it being sniffed on, which is a problem only in a scenario where you do know what the shared key is.

Remember that we're sort of creating a synthetic situation here with WPA. Most people, like all of us who are using WPA at home, we know that we need to keep our key secret. So we were suggesting that the key be made public only to get some crypto on the connections to defeat Firesheep as an interim measure until sites like Facebook are able to go 100 percent SSL.

Leo: Well, speaking of Firesheep, nice segue. Shawn Poulson in Middletown, Delaware talks about some of the challenges of going full SSL, which we have prescribed as a solution to the...

Steve: Ultimately, that is the solution.

Leo: Yeah. Steve, you made another great episode of discussion in the latest Security Now! podcast Q&A. I enjoyed all the talk about the technology required to secure websites from Firesheep attacks, or basically cross-site cookie stealing, I guess. I think you made a great point that SSL is not computationally expensive these days. Google made that point for us.

However, as a web developer and having been part of a production deployment of a commercial website using a Content Distribution Network, or CDN, I can offer to you that switching to SSL is not always as easy as just throwing a certificate on your servers. Site owners can be compounded with significant server and bandwidth costs. If a site were just a handful of web servers, it would be as easy as installing certs and going full SSL. However, some amount of additional bandwidth throughput will be utilized because browsers and intermediate caching proxies cannot cache secure content like it can with nonsecure content. Browsers will temporarily cache in memory during a session. Of course that's still allowed. But afterwards it has to be thrown away. Returning to the site under a new session requires redownloading all the images, scripts, et cetera.

Furthermore, when implementing a CDN, as Facebook has for pictures at sphotos.ak.fbcdn.net, certificates will not come cheap. CDNs work like a giant distributed caching proxy server. By the way, our files are all distributed by CDN, via CacheFly. Actually, your audio is distributed - your video is through CacheFly. Your

audio is distributed by AOL. But I'm almost certain AOL is using something, probably its own CDN. A user hits a link hosted by a CDN. The DNS resolves to a CDN edge server which is geographically close to the user. That's the benefit of a CDN.

Steve: Right.

Leo: The edge fetches the requested content from its source server at Facebook and caches it. The edge server delivers the cached content back to the user. Edge servers will synchronize caches to gain greater geographic coverage. SSL caching at the edge is still possible because SSL is only between the user and the edge server. The edge requests the content from the source server in a separate session, hopefully also using SSL. Every edge server in the CDN needs an SSL cert installed for your hostname, and that could be hundreds, if not thousands of them, depending on the CDN provider. It's true, everybody who's getting a video copy of Security Now! ostensibly from CacheFly.net is actually getting it from a different server. There are servers all over the world.

If your organization requires the $1,000 per year VeriSign certs, that can quickly become cost prohibitive. One alternative is that CDNs may offer a shared secure hosting wildcard cert with a shared domain name that may be free or cheap to use, for example, Facebook.somecdn.com. That's an example, that's not an actual URL.

Steve: Right.

Leo: My suspicion is Facebook needs to rearchitect their CDN infrastructure to avoid these excessive costs. Hope this helps shed some light on the situation. Thanks again for the excellent podcast. That's a good point. Is he right?

Steve: Well, he's absolutely right. And the point that he also made about, even short of CDNs, for example, cable modem, we know that cable modem providers often have their own caching proxies, a transparent proxy such that when you're making requests, for example, for Amazon.com, many of Amazon's own page components are the same no matter who you are, no matter what user you are. So if I use my cable modem to access Amazon.com, my own browser will cache a bunch of these pieces. But the ISP's got its own cache inline, which is also caching. So that if some other customer of the ISP pulls up Amazon.com, this transparent caching proxy at the ISP says, oh, wait a minute, this user has asked for the same image that that user asked for, so it provides it out of its own cache. The ISP does that primarily because it saves its bandwidth costs. It's not having to pay for as much transit bandwidth as it otherwise would, and the advantage for the user is that image is being served by the local caching proxy rather than remotely.

So the problem with all of that is, if the caching proxy is only able to see into non-SSL connections, that ISP's caching proxy, it's completely blind to anything SSL. So if the page and all of the page's objects are over SSL, it can't proxy them at all. It can't cache them at all. So all of the fetches out from the browser to Amazon have to be direct and cannot be intermediately cached. And so this is sort of a variation on the full-scale content delivery network, which is very much the same concept, but it's done explicitly by the website provider, like Facebook in this example, as opposed to sort of implicitly, and even in sort of a hidden fashion, helping the user's browsing experience and limiting

the ISP's bandwidth. So it is the case that there are bandwidth costs, and there is some performance hit for pages that are providing all kinds of content because you're no longer able to cache them in any intermediate location. It's always got to go back to the origin server in order to provide the content.

So he does make a good point. There is some cost, not to the SSL connection, but to the fact that it does defeat caching along the way, unless you do something like, as he was suggesting, there are wildcard certificates. I got one about a year and a half ago from GoDaddy that I've mentioned before, where it was *.grc.com. And I learned quickly that it wasn't the same as *.*.grc.com, which is actually what I needed. You can only have one level of uncertainty in that wildcard certificate, so it didn't end up serving my purpose. But that is one way of defraying some of this cost. But yes, there is going to be some performance hit, not from the public key aspect of SSL, but just from the fact that more bandwidth will end up being used.

Leo: And for somebody like Facebook, which serves a huge amount of data, especially pictures, that's not insignificant, frankly.

Steve: Yeah, it'll be interesting when, I mean, I'll be interested to see, when they do this, how they do it. And I'll probably take apart a Facebook page to see how they've solved the problem. I imagine they'll be providing somehow SSL CDN content, which I imagine, I mean, I'll be they're working on it right now.

Leo: I guess they wouldn't have to SSL everything. You could have one of those mixed pages; right?

Steve: Yeah. And that, again, that's why I really wish there was a means for a server to say, allow these non-SSL pieces, because there's no security downside to them providing images and other chunks of their, like, large chunks of their site that are going to be the same for everyone over a content provider that's not SSL. But again, doing so would bring up that warning saying, oh, no, some of this is not secure, even though it's just not a problem. It's by design.

Leo: Couple more questions. Question #6 from William McMahon in Toronto, Ontario mostly has a question about router DNS configuration. Actually, maybe the "mostly" goes with the "question." I don't know. We'll find out. Steve, I've been using your DNS Benchmark tool. Great job, by the way, he says. I'm a little curious about some of the settings. I've never used a custom DNS server before and always just used my home router as my DNS, which uses my ISP's DNS servers in turn. My network at home is running DHCP; so, as you know, it pushes the DNS servers as well, in my case the router gateway. I was wondering if there's a way for my router to push the public DNS server's IP instead of pushing the router's gateway address to the machines in my home. I can statically configure my DNS IPs on my router, but it still pushes the router's gateway address as the DNS IPs. The only other way around this would be for me to go to each computer and manually type in the custom DNS servers I want to use. But that's a pain.

Lastly, a comment. Now that you're done with DNS Benchmark, you should have all the time in the world to work on CryptoLink. Right? Any updates? I'm dying to hear

more. William.

Steve: Well, actually that mostly was that he had a SpinRite testimonial at the beginning.

Leo: Ah.

Steve: And I thought, okay, we've talked about SpinRite enough.

Leo: So here's a happy customer.

Steve: So I cut it out, and I forgot to remove the "mostly." So...

Leo: Now we know.

Steve: So that's what that was. Okay. So most routers, I can't say all routers because it would be up to the router manufacturer, but to sort of clarify what he's saying, he's saying that he's used the Benchmark, and he's found better DNS servers than the ones his ISP is providing. And that's often the case. So he knows he could go and manually configure those DNS servers in all of the computers in his network. But what if he changes his mind? He uses the Benchmark tool in six months, and he finds different ones. What he'd like to do is have his router provide those to his computer, rather than have his router always provide its own address as the IP which his computers use for their DNS. And I concur. It really is a better thing to do. It's not clear to me that there's a tremendous benefit for, like, routing your DNS through the router. And in fact, as we're going to learn next week, there are reasons not to because it turns out that some routers crash when being asked to do just regular DNS. So...

Leo: Really.

Steve: Yes. I have seen, in the routers I've looked at, there is typically a setting that allows you to turn off sort of this router proxying. It's just, it's not caching. It's not powerful enough to really add any value to DNS. All it's doing for some reason is giving you its IP as DNS. I believe that our own endpoint client machines do a better job with having two DNS servers than the router providing it only with one, meaning its own IP. So, William and other listeners, I think, if you look at your settings, you'll often see a setting that says "Use router for DNS." That you can turn off, and then you can still manually configure which DNS servers you want it to offer to all the machines in your network. And that's, I think, the optimal configuration, once you've got the IPs of the DNS servers that you want to use, which of course GRC's Benchmark provides.

Oh, and as for CryptoLink, I've got some more stuff I need to do before I start CryptoLink, which is just as well because I am seriously perturbed about the FBI and what Congress may be doing relative to requiring wiretapping technology in anything like this. I'm hoping that a standalone product which just provides endpoint encryption would

not be subject to this; whereas something like GoToMyPC or Skype, which is involved as a third party, I could see where they may have a requirement to provide that kind of wiretapping. But, yeah, I'm hoping that something that isn't a third-party involvement, and CryptoLink won't be, it'll just be point to point, I'm hoping that would not fall under this kind of legislation. It's hard for me to imagine that it would. So I've got my fingers crossed. But I do have other things to work on in the meantime. I'll be getting to it as soon as I can.

Leo: I'm just looking at the configuration of our D-Link router here. I think this is maybe where I would do this. It says at the bottom, DNS in advanced settings, use these DNS servers. And you can see I've put in the IPs for OpenDNS. I presume that that means, don't use me. But I don't know. I'll have to - would it be in the DHCP section?

Steve: Well, yeah. Now look at your own computer and see what the computer using the router, does it have the gateway IP, or does it actually have the OpenDNS IPs?

Leo: I see. I see what you're saying. All right. I'm not sure I want to waste everybody's time doing that. But I will check and report back.

Steve: And that's what our listeners can do.

Leo: Yeah. That's how you would know. All right. Moving right along. Question #7 - oops - comes from Pete [laughing], listening in Rochester, New York. He wonders about GRC's transcripts. Steve doesn't know what happened there. He's not watching. So don't tell him. I have a limp microphone. Steve, I enjoy reading the PDF of the show. Do you use a software program to transcribe from the audio recording? And if so, could you please give us the name? Great show, learned a lot about SSL from the Firesheep discussion. What is the name of that software you use to transcribe, Steve?

Steve: It's fantastic software, Leo.

Leo: Isn't it good.

Steve: It never makes a mistake.

Leo: Never.

Steve: We can fumble around and humble and mumble, and one way or another, even our misspellings get fixed for us.

Leo: Isn't it amazing.

Steve: Yes. The software's name is Elaine. On-Site Media is the name of the company that Elaine is [elaine@on-sitemedia.com]. And she's been doing all of our...

Leo: Transcriptions.

Steve: ...all of our transcriptions from day one. And so, Pete, unfortunately, if it were software, boy, it'd be really popular because it'd be fantastic, but it wouldn't work nearly as well as Elaine does.

Leo: We've all seen the lousy job Google does on YouTube transcription. We've all seen - even if you watch TV, you can always tell when the closed captioning is done by a human versus a machine. The humans don't do a great job, but the machines, really, it's ludicrous. No, you need a human to do this right. And, you know, Elaine charges. Steve pays it, by the way. Credit to Steve because he really wanted to have written transcriptions of the show, and so he pays Elaine out of his own pocket to do that. So thank you, Steve. [Elaine thanks Steve, too!]

Steve: Glad to.

Leo: Finally, our last question. Dave Solon, in Lancaster, PA, will share his podcast survey results. Steve, huge fan of your podcast for years, along with This Week in Tech (TWiT). I was wondering if you might help me out with a grad class research project. I'm a K-12 Instructional Technology Specialist in Lancaster, PA, also an avid podcaster. His show is TwentyForTech.com. I'm a huge advocate for teachers and students to start their own podcasts. As am I. I set up a podcast studio for my kid's high school. And I'd like to help guide them to create podcasts in formats that most folks like to listen to or watch. I've developed a short survey to try to find some things out to help me in my quest for podcasting proliferation. He'd like us to share the survey URL for the good of the education and podcasting community. The page is davidsolon.wikispaces.com. He says, I'll share all my data and paper after it's complete. Thanks for your consideration. And if Leo could share this study, I'd be forever indebted to you both.

Steve: So it's davidsolon.wikispaces.com. And what I found interesting about this, I mean, first of all, the survey is neat. I'd love our listeners to give him their feedback. He promised to email it to me so that I could share the results back with our listeners. So we'll close that loop. But he used some Google spreadsheet technology.

Leo: Yeah, we've used this. We use this, too, for our surveys. It's really great.

Steve: It's interesting. I hadn't seen that before. But I thought that was really neat, that you're able to use just a - somehow, like, simply format a spreadsheet and collect the data and see what everyone had done. So...

Leo: Yeah. So it looks like a real survey, but then the results end up in a Google

spreadsheet at Google Docs. We're actually using - I'll tell you what. This is his survey. And once again, there's a short link for this, which is bit.ly/podcastresearch will take you right there, bit.ly/podcastresearch. But if you want to take our survey, using exactly the same technology, we're compiling "best of" shows for the holidays, the week after Christmas. Your show we've already got solved, that's the Portable Dog Killer. But for TWiT and TWiG and some of the other shows, if you'd like to participate, tell us your best, your favorite episode. That's TWiT.tv/bestof. And it's exactly the same technology. We embedded it in an iFrame on our TWiT site. But this also gives us a Google spreadsheet. It's powered by Google Docs.

Steve: Very cool.

Leo: And kshep set that up for us. So you can choose the four shows that we're doing best-ofs are TWiT, MacBreak Weekly, This Week in Google, Windows Weekly, and Tech News Today. If you know the episode number or the air date, that'd be great; what time it occurred, that'd be great. If not, just put what you know. And we will thank you, and you'll hear the Best of TWiT then the week after Christmas. Most of the shows are going to go dark for that time. Including this one, but this one will not miss a show. We've never missed a show. We'll just provide you with a repeat of one of our favorite Christmas classics. And you know what we could do, Steve, we could record a kind of an intro and some information. In fact, even if you wanted to do the tech news and errata and stuff as an intro to it, you could update that part.

Steve: Okay, cool.

Leo: Yup, the Portable Dog Killer. It's the best. Steve, you're the best. Steve Gibson is at GRC.com. His Twitter handle: @SGgrc. You can follow him there. GRC of course is the home of SpinRite, the world's finest hard drive maintenance and recovery utility. If you have a hard drive, you really ought to have SpinRite. But lots of free stuff there, including the new DNS Benchmark and many other useful utilities. GRC.com. That's also where you'll find the 16KB versions of this show for the bandwidth-impaired; Elaine's great transcriptions; the show notes. And if you want to ask a question, that's where you'll find the feedback form, GRC.com/feedback. Do you know what we're going to do next week? Is it a surprise?

Steve: Well, I do know what we're going to do because on the heels of the DNS Benchmark, the thing that actually was the stimulus for that was the revelation of how many DNS servers were vulnerable to spoofing that Dan Kaminsky provided. And I created a complete, rather amazing, frankly, piece of technology at GRC which is a system which allows people to check the spoofability of their own DNS servers. So GRC's DNS Spoofability Testing System is our topic for next week. And lots of technology in that, so it's going to be a propeller-spinner.

Leo: Yeah, that sounds fantastic. Steve, you're the best. Thank you for being here. We'll see you next week on Security Now!.

Steve: Thanks, Leo.