SECURITY NOW!

Transcript of Episode #184

Listener Feedback Q&A #60

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-184.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-184-lq.mp3

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

Leo Laporte: Bandwidth for Security Now! is provided by AOL Radio at AOL.com/podcasting.

This is Security Now! with Steve Gibson, Episode 184 for February 19, 2009: Your Questions, Steve's Answers #60. This show is brought to you by listeners like you and your contributions. We couldn't do it without you. Thanks so much.

Time for Security Now!. We're going to talk about all those things that happen on the 'Net that can scare the pants off of you - your privacy, your security. Steve Gibson is here. He knows more about it than practically anybody because he's on the front lines. GRC.com is his site. Creator of ShieldsUP!; the discoverer of spyware. He's like Christopher Columbus. He discovered spyware. Hey, Steve Gibson. How are you today?

Steve Gibson: Hey, Leo, it's great to be with you again, as always.

Leo: Great. Today it's a Q&A segment.

Steve: Yes, Episode 184, and this is our 60th, six zero. That's a decimal six zero. I've been deep into octal notation recently because I've been learning about the PDP-8, like relearning. It's funny because the book I was reading says "Introduction to Programming." And it was sitting on the coffee table at Starbucks. And the baristas there pretty much know me and sort of have a sense for where my area of expertise is. I've been there for years. And one of them looked at it and says, "'Introduction to Programming.' You're reading an introductory book?" And, you know, DEC printed it on non-acid-free paper, so it's very yellowed and aged looking.

Leo: Oh, how funny.

Steve: And I said, "Well," I said, "let me put it this way. I first read this book when I was 16 years old."

Leo: It's an oldie but goodie.

Steve: So I am rereading it and enjoying it. And in fact that is a perfect opportunity for me to mention that I'm going to be joining Ray Maxwell over on his podcast that you and he do. I guess you record it live on Thursdays.

Leo: Yeah, we do. It's called Maxwell's House.

Steve: Maxwell's House.

Leo: Yeah, it's really fun. It's 2:00 p.m. Pacific, 5:00 p.m. Eastern on Thursdays. And it's, you know, Ray is like you. He's an autodidact, and he's got wide-ranging interests. So it covers...

Steve: And I had no idea he was, like, a computer developer.

Leo: Oh, yeah.

Steve: I was listening to him talking about the old days. And it's like, my goodness, I mean, he and I, there's a tremendous overlap between what he's done and what I've done. I just thought he was Mr. Photography.

Leo: Oh, gosh, no. You know, that's the hobby. So it's fun. And so, yeah, so you two are going to do an old-timers' show.

Steve: On February - is it the 28th? It's not this coming...

Leo: It's the 26th.

Steve: It's not the Thursday that this show airs.

Leo: That's the 26th, a week from today.

Steve: The 26th. So I wanted to advise our listeners, if they want to listen live, they can certainly do so. It'll be Thursday afternoon at 2:00 Pacific, 5:00 Eastern on Thursday the 26th?

Leo: Yeah.

Steve: And then of course they could also explicitly grab that podcast. What we're going to be doing is talking about old times, old-time computer technology stuff.

Leo: But we don't do a podcast of it, Steve. It's just live. But I think that one...

Steve: Oh, no kidding.

Leo: Yeah.

Steve: So you're not recording it.

Leo: Well, we record them, and we play them back as reruns. But we have, you know, this is a video show, and we haven't yet really figured out how to do those. But this might be the exception. And if we do it, what we'll probably do is put it out on YouTube or somewhere. But I'll make sure people know where they can see it again.

Steve: Well, and I...

[Talking simultaneously]

Leo: And of course you have ODTV. And ODTV at ODTV.me records many of these shows and makes them available.

Steve: Okay. Well, I would definitely like to get the audio, and I'll host it on my server because I'm going to...

Leo: Oh, good.

Steve: ...spend the time to - anyway, so he and I are going to hang out and talk about, sort of do a nostalgia computer episode that he's eminently qualified for. And I've been living in the past a lot recently myself. So I think it'll be - we're really going to have some fun.

Leo: You know what we could do is release it as a auxiliary - like an extra show on this feed.

Steve: Yeah, that's a good idea. I mean, why not?

Leo: Yeah. And so if you're already a subscriber to the Security Now! feed, whether it's on iTunes or your podcatcher of any kind, we'll just have, you know, it'll be Episode 185-A or something like that.

Steve: Yeah. You'll just get it. And if you don't want it, you can delete it.

Leo: Throw it out, yeah.

Steve: Yeah.

Leo: Good. That's what we'll do. That'll solve that. So an extra Security Now! next week.

Steve: Yeah. So anyway, I'm excited about it. And as I said, I've been studying these older machines. And I've - I realize and appreciate some things I never appreciated when I was 16, you know, from the vantage point now of looking back, how many years is that, 33 or something?

Leo: Long time.

Steve: Yeah. So it's going to - I've got some interesting observations I think people will - certain Ray and I will have fun talking about it. And I think our listeners will find it interesting, too.

Leo: Why do they work in octal? I mean, isn't - well, we know that binary is the natural state of a computer.

Steve: And that's the heart of what - and that's, of course, I was looking at six zero, Q&A 60, and that's the hardest problem I'm having is, I mean, I live in hex. Unlike, you know, almost all programmers now...

Leo: Hex is base 16.

Steve: ...who live in decimal. Yes. I mean, my world is hexadecimal because I'm programming in Assembler. But back then the machines used three-bit groupings, so you had zero through seven, instead of four-bit groupings. And DEC produced minicomputers with bizarre word lengths. There were some that were 18 bits long. Some were 36 bits long. Some were 12 bits long, and 16. And so they had multiples of nine and multiples - and so it made sense to have, like if you had an 18-bit word, you'd do multiples of three bits in order to represent that.

Anyway, so it's difficult for me because I'm so used to a computer representation being hex that, I mean, it's been interesting because I haven't appreciated the degree to which I translate hex now automatically into binary and into decimal. It's just - it's automatic. But if it's octal, that automatic reflex is wrong. And so I'm having to catch myself constantly, say whoa, wait a minute, this is not, you know, 200 when I'm seeing it in octal is a very different number than it is in hex, so.

Leo: That's really, really geeky, but very cool.

Steve: Yeah. Oh, but I've got more for you in two weeks.

Leo: I can't wait.

Steve: I've got some serious geekdom.

Leo: So, Steve, dare I ask, are there any news and errata from...

Steve: Well, yes. We have lots of errata and a little bit of news. It's been a relatively quiet security week since last week when we had the big Apple second Tuesday of the month update.

Leo: Hallelujah.

Steve: In this case, the only real big security news is Apple. There is a new Mac OS X update and Java update.

Leo: Yeah, I got those, yeah.

Steve: The Mac was about 44MB. They fixed a whole bunch of problems, more than two dozen security flaws. There was an arbitrary code execution flaw in Safari's RSS handling. There was an information disclosure flaw in Apple's Remote Events, a denial of service flaw in the AFP server, arbitrary code execution in Core Text, and a bunch of other things. So definitely something that Mac users will want to make sure they get. And they didn't really talk about what they fixed in Java. They just said "addresses security and compatibility issues." So...

Leo: They never say much. They're very tight-lipped about that kind of thing. What do you think of that? I think their thinking is we don't want to say too much because we don't want to tell people what we're doing. On the other hand, I would really like to know what they're fixing.

Steve: Yes.

Leo: And what they haven't fixed, more to the point.

Steve: It's my feeling is that we're used to Microsoft giving a relatively full disclosure of what's going on. And of course this has come back to bite Microsoft. So I think Apple is saying, uh, we're going to fix it, and why do they really need to know? I mean, you might argue that the typical Mac user, being less screws and knobs and widgets, just wants to know that they're secure...

Leo: I don't think that's true anymore. I think a lot of programmers and techies use Macs.

Steve: Yeah, well, as I look around Starbucks, I am seeing a clear shift towards Macintoshes; although I'm right next to UCI so we're in student land there, too.

Leo: Well, I mean, because the Macs come with all the programming languages and so forth, I think, and it's a UNIX [indiscernible] terminal, I think a lot of geeks just kind of shifted over to the Mac platform.

Steve: Yeah.

Leo: And they want to know. I mean, especially because, it's actually related, a lot of the Mac software is UNIX software. So some of these patches and some of these exploits are in the UNIX software. And if you know of a problem, for instance, in the version of Bind that OS X is using, you want to know if they've patched that or not. I guess you could figure it out, but it's nice to have a list.

Steve: Right. The only real interesting security news I have, aside from Apple, was just a note...

Leo: A yabba-dabba-do.

Steve: There's a yabba-dabba-do. I'm going to talk about that a little bit when I talk about SpinRite later. But there was a - the trial has started in Stockholm of the four founders of Pirate Bay.

Leo: Yeah. It's closely watched by a lot of us.

Steve: Yeah. I'm interested to see what happens. They're facing charges of accessory and conspiracy to break copyright law. Their defense is that the software is not stored on their servers. So they're not breaking the copyright law. Thus they're not being sued for copyright violation, but rather accessory to and conspiracy to break copyright law. The lawsuit is seeking about $14 million U.S. in damages and interest. It's estimated that Pirate Bay has about 25 million users, if you can believe that. I mean, that's just amazing to me. And if, I guess, based on the nature of the charges, if they were convicted, these four guys could face sentences of up to two years in prison and fined as much as $180,000.

So the only thing that I have found annoying about them, I mean, I recognize the reality of software piracy and the way the Internet works, is just - and I'm sure you've seen this, too, Leo - is the brazenness of these guys. I mean, all kinds of companies large and small have sent them letters asking that their software be removed. And they post them and laugh at them, saying nani nani nani nani, you can't get us. So it'll be interesting to see how that turns out.

Leo: The first day of the trial was a couple of days ago, and they already have thrown out one of the charges because they couldn't prove they had a - I think the prosecution is fairly inept. They were unable to prove that the actual - that the torrents they were - the torrent files they were showing actually were on Pirate Bay, and there was no evidence that they were. So the judge said, uh, hmm, guess that one's out. So now the only charge remaining is making available of copyrighted works. You know, to be honest I think it's going to be a very tough one to prove because they don't make available copyrighted works. All they publish is this little tiny torrent metafile.

Steve: Right.

Leo: I don't know how you prosecute that.

Steve: Yeah. And, I mean, I don't wish them any ill. I just - I don't - certainly obviously I don't condone software piracy. I'm a person, you know, when you hear that yabba-dabba-do it's because an honest person purchased a copy of SpinRite.

Leo: Yeah, and I guarantee you that SpinRite is one of the things, one of the torrents on Pirate Bay. I haven't looked, but I wouldn't be surprised. Everything else is. So you're, you know, these guys are stealing from you, too.

Steve: Now, I have my own horrific PayPal story.

Leo: Oh, no.

Steve: Not just a caller-in. This is something I stumbled on that is a complete dongle bypass, believe it or not.

Leo: Okay. Because that dongle we use seems to secure PayPal better because, you know, enter that number in, only I have that dongle, and...

Steve: Yup, multifactor authentication. And in this case the second factor is a one-time password that gives you proof against keystroke loggers or somebody looking at what you're typing in and then going and trying to use it because it will have been used, and it expires. I've been buying a lot of things off eBay. You know, I'm in the antique or vintage computer mode at the moment, thanks to you having started me off by holding up that plane of core memory, and I thought, I gotta get some of that.

Leo: Get me that. I gotta get me that.

Steve: Before it goes away. So now, that, I stumbled back into, you know, DEC, the world of DEC and my first minicomputer, the PDP-8, and then the 11.

Leo: Boy, two in a row.

Steve: Well, okay. I'm going to hold off talking about that till I get there.

Leo: Okay.

Steve: So I've been buying a lot of things from eBay. And eBay of course is closely affiliated - that's putting it mildly - with PayPal. And so all the eBay sellers want you to use PayPal in order to complete the purchases. So I went to pay for something, and went to the sort of the summary screen where this is what it's going to be, and here's the shipping costs and so forth. And you're still sort of in eBay land, and you click to confirm that to go over to PayPal. So I did that, went to the PayPal screen, which asked me for my username and my regular password, the first factors of authentication, which I entered. Then I clicked Login, and it took me to the - it recognized in its database that I had the security token associated with my account.

And so I looked at that screen, I thought, wait a minute. I had meant to add a comment to the purchase I was making. There's a, you know, two screens back when you're agreeing that this is the price and shipping and so forth, there's an option to send a comment to the purchase - to the seller. And I don't remember what it was. But so I hit back screen and went to the login page, and back again and got back to the eBay page. Opened up the little dialogue area where I could put a comment in and typed something to the guy. Then I clicked on Confirm. This time, when I went to the eBay login page, the fields were grayed out, you know, because I had done that already. So I clicked Login, bang, and I was at eBay, logged in, never having to use my football.

Leo: Whoa.

Steve: So, and I did it...

Leo: And not asking security questions, either.

Steve: No. It happened once before. And then yesterday I remembered that it had happened and so I deliberately duplicated it. So I've done it now twice. I don't know for sure how far back it's necessary to go. I went back two screens. It may only be necessary to go back to one, where the login fields were grayed out. And then clicking Login might jump you over. My guess is that would. Next time I buy something I'm going to try that. So I'll...

Leo: Do you think it's something that has been cached?

Steve: Well, it just means there's a bad implementation. I mean, we keep seeing example after example of people at PayPal just not having their act together from a programming standpoint. And, you know, and also from a security philosophy standpoint. But here, I mean, this is something where you just - it bypasses the whole requirement of entering any sort of a second authentication credential. Not good.

Leo: It's a handy tip, though, because if I forget my key at least I can get in. [Laughter]

Steve: There's a handy tip for you.

Leo: How to bypass all the security at PayPal. Just in case you forget your password.

Steve: Oh, yeah.

Leo: Wow.

Steve: Many listeners have thanked me for my mentioning and recommendation of KatMouse, the little Windows scrolling utility. I mean, people are just as nuts over it as I am. So I just wanted to acknowledge all the people who wrote and said thank you, thank you, thank you, I would have never found out about this if it weren't for you mentioning it. It's changed my life. I mean, there are people who have, like, they've changed their habits, where they're deliberately now scrolling Windows that are behind other windows, like they can see part of it, but they don't have to, like, make the windows come forward. So they just put the mouse over it, scroll that one, then go back to the window where they were actually working. So, I mean, it's been a huge, life-changing event for a lot of our listeners. I just wanted to thank them all for the feedback that they appreciated that.

Last week was the first episode of "Dollhouse" that I mentioned, Joss Whedon's new series. I just wanted to say that I was kind of underimpressed, underwhelmed.

Leo: Yeah, you were not alone. I didn't see it, but a number of people said, eh.

Steve: Yeah. It just doesn't grab me. But I saying that, I wanted to say that my absolute number one favorite show on television is now, and has been for some time, but I haven't mentioned it again, "Fringe."

Leo: Is it back?

Steve: Oh, it never left. We're in the first season. And it is just - I look forward to it every week, more than any other show on television. "Terminator" sort of seems to have lost its way. They're running around in circles every week, and I don't really know where it's going. But "Fringe," for what it's worth, I mean, I know there'll be listeners who are like, yeah yeah yeah yeah yeah, I agree completely. Maybe there'll be people who disagree. But I just love it. I think it is really well written. And it's another J.J. Abrams production. I think J.J. Abrams. I'm pretty sure. And it's just spectacular. I love the writing. I love the casting. And last week this episode with the light box, the episode was titled "Ability," a whole 'nother layer of story arc was unveiled. It's like, oh, goodness, I hope this show is doing well enough to be continued because I'm really enjoying that hour of television. Or 40 minutes after TiVo gets through with it, so.

Leo: 40 minutes. It's true, it's true.

Steve: And then, on the topic of collectibles, I have an offer for some lucky listener or listeners. A listener of ours, who's also jumping in with the SpareTimeGizmos, building the PDP-8 kit, Lance Reichert, said - he wrote and said, "I have an Osborne 1 - late-model, light-blue case - and a Kaypro 4, both in storage, with bundled software and manuals, plus several programming languages, as well as symbolic math package. The Ozzy even has the original box and packaging. Could you please connect me with someone who might be interested in these free-for-the-cost-of-shipping machines?" He says, "I already checked DigiBarn, and neither is on their list of machines they're looking for." And so I wrote back, and I said, well, "Why don't you create a throwaway email account that you can use, and I will give everybody who's listening to Security Now! that throwaway email account so that they can contact you if they're interested." And so his email account, it's Gmail, it's kaypro.iv - as in Kaypro IV - at gmail.com. And anybody who is willing to pay the shipping cost for either this late-model Osborne 1, which was a little CPM machine...

Leo: Oh, I'd love to have that, wow.

Steve: Huh?

Leo: I'd love to have that.

Steve: Well, you can have it, Leo. I'm sure Lance would just send it...

Leo: Wonder what the shipping cost would be, though? That was a - that, quote, "portable" was 30 pounds.

Steve: Well, yes. Remember that the term "luggable" was coined for that machine. Or maybe it was the Compaq.

Leo: I wanted that Osborne so badly. Does he have a Kaypro, too? Is that what...

Steve: Yeah. An Osborne 1 and a Kaypro IV is the - both of them in storage. You know, if you'd - can you display them somehow?

Leo: Well, that's the thing. I think that - yeah. My problem is, unlike you, while I would love to have a computer museum in here, I don't really want - I could be bulging at the seams here with all the stuff that I would love to have. And that's a bulk- both of those are bulky units. That Osborne had a little tiny screen. That was a tough - but the Kaypro was bigger. It was nice.

Steve: Right. And the Osborne was what, it was like eight-inch diagonal?

Leo: Yeah, it was teensy.

Steve: Little micro, and it was, you know...

Leo: But it was 80x24, I think, wasn't it? Or no?

Steve: I'm sure that - I'm sure it was 80x24 characters, probably text. I don't know, I don't remember if it did graphics.

Leo: No. No no no no no. No graphics.

Steve: It was just a CPM machine, so probably just text.

Leo: Yeah, yeah. And they came with, as did the Kaypro, WordStar.

Steve: Yup. Hey, now, careful, Leo, I'm still using WordStar keystrokes for all of my editing.

Leo: Control - wait a minute, don't tell me, Control KS to save? Is that right? Control KV?

Steve: Yes. Yes yes yes.

Leo: Some things never leave the mind.

Steve: And, I mean, what I love about it was that your pinky sat on the Control key, which is now where everybody has moved the Shift Lock, which is just...

Leo: Which is really frustrating, yeah.

Steve: ...the dumbest thing. When was the last time you ever hit Shift Lock on purpose? I mean, it's just...

Leo: I always map it to the Control key.

Steve: Yes, actually there is a simple little registry hack you can use that'll map it in Windows machines so it is the - exactly. And so both my Shift Lock and my Control key are Control keys. But, yeah, I just, I mean, it's the definition of a perfect UI because it disappears. I just will the cursor to go or the screen to scroll, and my left hand does it for me using the Control key and the little alphabetic keys over in that area.

Leo: But you don't have a program that does it automatically. You have to map the keystrokes to whatever program you're using.

Steve: No, well, in my case I'm still using Brief, which is an old 16-bit DOS box editor. And I've completely rewritten it - Brief has a list-like macro language which I customized about 20 years ago. And it's still running strong.

Leo: And if you tried to change it one line today, you'd probably break the whole thing.

Steve: Oh, yeah.

Leo: I'm sure there's no way you could look at that and say I know what it's doing now.

Steve: I'm not touching it.

[Talking simultaneously]

Steve: A person posted in the GRC SpinRite newsgroup really kind of a fun and, again, different sort of testimonial. The subject was "Thank You, SpinRite," and he posted on January 6th. And he said, "Okay. I'm sure you're all sick of thank yous." And I guess he's speaking now to everybody in the newsgroup. "But this disk just saved me a lot of time, even though it didn't touch the drive." He said, "I bought SpinRite a while ago to support Steve and Security Now!" - oh, so he's a Security Now! listener also - "but never had the need to use it until last week. With the Christmas shutdown, I was looking to get out and away on the last workday, but I got stuck with a service call to one of the offices with a computer hard drive failure. The staff there are actually pretty good. So when they say it's a hard drive failure, I tend to believe them.

"Anyway, I go there, boot the computer, and it comes up with nonsystem disk message. So I can see some time being wasted recovering data that should have been stored on network drives that are backed up, and not on the actual PC. Oh, well. I popped in SpinRite, and it comes in with a cabling error message. A quick Google didn't really come up with much, but I had a few days to turn this unit around, so I posted the question on GRC's newsgroups. Within a day I got a reply to check the cable. So I popped the box, then reseated the SATA cable." And he says, parens, "(They should design these so you have the option of connectors locking in), and I rebooted. The computer came right up, Windows loaded, and it all works. Nothing was lost."

Leo: Wow.

Steve: "Needless to say, they walk in on the first day of work and are impressed that I sorted it all out over the Christmas break, and even more so when I mention the mystery disk that diagnosed the problem." I guess he means SpinRite. "So just one more for the records. Thank you, Steve. And even though SpinRite never really touched the drive, it was still worth every cent. Signed, Tim in Perth, Australia."

And what SpinRite does is - this is sort of a consequence of the product's maturity over time. I remember clearly adding this in SpinRite 3.1 because we were seeing instances where people were running SpinRite when they didn't really need SpinRite. A classic was that their BIOS had lost its drive settings. And so they would run SpinRite, even though there was nothing wrong with their drive. The reason their OS wouldn't boot was the BIOS had lost its drive settings. And so I added code to check the BIOS drive settings and tell them, Hi there. You probably don't need SpinRite. Thank you very much, though. But here's what the problem is. And I would, you know, take them through fixing that.

Well, one of the other things that was happening was that cables were becoming unseated. And there is a protocol from the IDE drives forward which does a CRC of the data transfer across the cable. And so I added some code in SpinRite - of course, and if the cable's loose, there's really nothing wrong with their drive. It's an electrical problem. So I added some code in SpinRite to test the cable separately from testing the drive. And that's been in there ever since. So, and of course it's a problem running SpinRite if your cable's flaky because we're testing, we're trying to correct electrical errors rather than magnetic data errors.

So, you know, that's been a real boon to SpinRite users. And of course this just fixed Tim's problem, which wasn't with the hard drive, but rather was with the interconnection of the hard drive to the motherboard.

Leo: Very cool. So if you test it, will you warn?

Steve: Yes. SpinRite pops up and says, wait. Before we go any further, you've got a problem with your cabling.

Leo: That's so cool.

Steve: Fix that, and then try again.

Leo: I wish more software would do stuff like that. So you could tell because it's kind of distinctive?

Steve: Well, yeah. I'm able to transfer data back and forth across the cable to the hard drive's buffer without actually reading or writing the hard drive. And I can pick up CRC errors in the transfer.

Leo: I see. So you know it's not the drive. You know it's the connection.

Steve: Exactly. Exactly.

Leo: Oh, that's very interesting. See, it's little things like that that make - that really make you an exceptional programmer. I mean, I think that's really cool. You're using your head every time. You're thinking about this.

Steve: Well, and with SpinRite 3.1 it was a major rewrite. It took a long time. But I was committed to doing everything I knew I could possibly do in that program. And, you know, I have the advantage also of being the boss. So the boss is never annoyed with me that it's not done.

Leo: It's not going to ship because I haven't put in the cable check software.

Steve: Yeah, yeah. And in fact I got into trouble with my wife at the time because SpinRite 3.1 was really late. And she came home from work one day, and I was jumping up and down, all excited because I had invented this brand new way of doing surface testing. And she was an attorney, and not really into computers at all. Which was fine with me. I was happy to have that be my little world. But I just wanted to - I was sharing my excitement over this really phenomenal surface analysis system that I had come up with that afternoon. And so when she finally understood what I was talking about, she said, wait wait wait wait wait. Stop. You're telling me that you're putting something new in this? And I thought, uh-oh. And I said, uh, well, yeah. But I always knew I was going to get around to doing this. I just hadn't gotten to it yet. And she said, Steve, you already have their money. It doesn't matter what you send them. And I...

Leo: Ooh, that's an attorney talking.

Steve: I just thought, boy, have I made a mistake.

Leo: You went, oh.

Steve: And I explained to her that the reason I had everyone's money who were waiting for me to ship this upgrade was that they knew when I got it done it would be everything I could possibly make it. And so I wasn't doing this for the money, I was doing it to create the best product I knew how to create. And anyway, I'm no longer married.

Leo: [Laughing] I'd break up with her, too, over that.

Steve: Yeah, well...

Leo: Steve, you already have their money. Why make it better?

Steve: I know. I just - I thought, you know, the reason I have it is that everyone trusts me to do the best job I can.

Leo: Yeah. This is why I don't talk about TWiT with my wife.

Steve: Yeah. Well, we've had a couple yabba-dabba-dos that we've heard in the background.

Leo: Yeah, what's that all about?

Steve: Well, as we know, that's when someone purchases a copy of SpinRite. Because I've got systems here that are monitoring our network health and what's going on with the server over at Level 3 where our datacenter is, the facility we use. Last week I muted that, as I normally always do, and as I historically have. Oop, there's another one.

Leo: Wow. And I have to say, that's more than I've ever heard before in one...

Steve: Well, and this is the point that I was going to make, is that we had a - I didn't mute it the week before. I muted it last week. But I noticed that there seemed to be an unusual number of sales...

Leo: People are trying to get on the show.

Steve: Yes. That's what I think. And so I felt...

Leo: So if you're going to buy the product, buy it during - between 11:00 and 1:00

Pacific Time on Wednesdays because you'll get a yabba-dabba-do.

Steve: And I could imagine somebody on TWiT Live, listening live, clicking the Purchase It button and then hearing yabba-dabba-do come out. And that's like, hey, that's my yabba-dabba. So...

Leo: You've come up with a remarkable new way to sell products.

Steve: Well, it wasn't deliberate. And, I mean, I'm sort of self-conscious and embarrassed a little bit. But...

Leo: I think you should turn it up a little bit. We just barely hear it.

Steve: So I muted it last week. And there, like, seemed to be, I mean, again, it's anecdotal. Maybe it's just my imagination. But there seem to be an unusual number, a concentration of purchases during the time we were recording Security Now!.

Leo: So you feel guilty because of the people last week didn't get their yabba-dabba-do.

Steve: That's exactly where I was headed was then I felt badly that people bought SpinRite hoping for the yabba-dabba-do treat, and they didn't get one. So we're not muted this week. And we won't mute it next week because if there's a week delay in people hearing the podcast and then saying, oh, good, I'm going to get my own yabba-dabba.

Leo: Steve, I've always said, not only should you not mute it, you should turn it up so we can really hear it. And if it goes on, you know, if you get a hundred during the show, well, what's wrong with that? And in fact, if I can only - if I can figure out - you've got to show me the code. Because if I could figure out a way to have a yabba-dabba-do go off every time somebody donates to TWiT, I think it would be a very smart thing to do.

Steve: Seems a little exploitive to me.

Leo: No, it's - if somebody's going to buy it anyway, and they want to hear their purchase go through...

Steve: Their own purchase, yeah.

Leo: I think it's a great idea. I should have - I might put - I might figure out a way.

I guess I could do that. Wait a minute. Let me think about that. Because I get an email from PayPal. It's a little bit of a lag, though. You really want to have it when you press that button.

Steve: Yeah. For me, that's what it is. They push the button, and within a second I get a yabba-dabba-do at this end.

Leo: It's like, you know, they ring the bell when somebody gives them a tip. I think that works. It's just psychology. People want a little extra. Little shout-out.

Steve: Well, they get it here. Whether they want it or not.

Leo: Yeah, because I've never heard - we just - we had three in half an hour. I've never heard that many before.

Steve: Yeah.

Leo: And I think now we're going to have a few more. Well, before we do our questions, because we have 12 good questions for you all, I am going to...

Steve: Neat, neat feedback, as we always get from our listeners. I just - there were 374 pieces of mail since I last pulled them down. And, yeah.

Leo: They're saying they can't hear it on TWiT Live. They want you to turn up the yabba-dabbas.

Steve: Oh.

Leo: Steve, I've got questions, 12 of them, sitting in front of me. Are you ready? Do you feel ready to answer? Do you want to take a break? Do you want to go yabba-dabba-do or anything like that?

Steve: Someone posted that they want a copy of the yabba-dabba-do WAV file. I will - I'll put it somewhere on GRC and...

Leo: I think you can find that online, too. I mean...

Steve: I did. It was just really just a...

Leo: There's another one.

Steve: Just an enthusiastic, happy Fred Flintstone. And it just, you know, makes me smile, so.

Leo: I think it's just - if I had a little light behind me that lit up every time, or maybe we could have a little angel fly across the screen every time somebody donated - I think that's a great idea. I like it.

Steve: Oh, here's another one coming.

Leo: You're kidding.

Steve: No.

Leo: Wow.

Steve: As they're filling out the form, there's a cash register sound, kaching-ching, kaching-ching. And then the yabba-dabba on successful completion, so.

Leo: I just love it that you do that. I think that's cool. You have a bunch of other sounds, too, I know.

Steve: Oh, and actually I've got some really nice synthetic voice sounds that monitor my network here. And it's like I've got one that says, "Your primary Internet link has gone down."

Leo: Oh, geez.

Steve: And "Your primary Internet link has come up." And then I also have this - I wrote some tools that monitor workstations that are doing, like, AV compression. And it'll say, "AV workstation is now idle," because I'm working and I want to know when another machine is done so I can go over and...

Leo: Oh, that's good. We need to do all of that stuff.

Steve: Yeah, it's really cool.

Leo: That's really good. And you wrote those scripts in...

Steve: In Assembler.

Leo: In Assembler, of course.

Steve: Of course. What else?

Leo: Question one, Alesia Ketchie, who is otherwise anonymous - okay - is very upset that Microsoft uninstalled her new antivirus program. Steve and Leo, she writes, I turned on my computer a few days ago, and I got a message saying that Microsoft MSRT had removed AV 2009 from my computer. So now I don't have an antivirus installed. I tried to download another copy of AV 2009, but I couldn't remember where I got it. Can you tell me - this is not - this is a joke.

Steve: No. I'm not kidding you.

Leo: This is a joke.

Steve: I am not kidding you, Leo.

Leo: This is a joke. Can you tell me where to find it, or recommend a free AV program?

Steve: [Sighing]

Leo: Oh, boy.

Steve: Isn't that perfect?

Leo: I can't believe it.

Steve: I mean, maybe she's pulling our leg. If so, I take my - I tip my cap, my hat to her...

Leo: It's pretty funny.

Steve: ...whatever I'm tipping, something. I'm tipsy. I just thought this was fantastic.

Leo: Oh, my goodness.

Steve: Of course AV 2009 is a virus.

Leo: Yeah.

Steve: It's bad.

Leo: Yeah.

Steve: And a lot of people have been getting it. And MSRT has been removing it from a lot of machines. So in case that Alesia is serious, we're not laughing at you, we're laughing with you.

Leo: Yes, because you're not alone. There are many, many, many people who've fallen for this. I get - literally I get this call on the radio show all the time.

Steve: Yes. Yes. So do not go looking for another copy of it. Actually it'll probably find you, without you having to look for it, and happily crawl into your computer. It is malicious. It's good that Microsoft MSRT removed it. And Leo, you're probably more on top of what's the best free AV program to recommend, so I defer to you.

Leo: I'll tell you a couple of things. First of all, the guys, the clever fellows at AV 2009 are now calling it AV, in some cases, AV 360, or 360 Protection. Because what they're doing is they're trying to trick you into thinking it's Norton because Norton Antivirus 2009 and Norton 360. So you may see other names for this. The way you get it is you will go to what you think is a normal website, and all of a sudden you'll see a popup on your screen that says, wait a minute, you have some spyware or viruses on your system. We'd like to check. Click okay. By the way, if you click okay or cancel, no matter what you click, the same thing happens. You get sent to a site where it says - it shows you a phony clicker going [indiscernible]. And I know it's funny because I've seen it on my iPhone, and I've seen it on my Mac, neither of which get any of these viruses. And so it goes click click click click click click click. And it says, yeah, yup, yup, you're infected. But we've got the free solution for you. Click this, download it. You download an executable and install it. And that's of course AV 2009 or AV 360, and it is a virus.

So there's a good program to get rid of this, Malwarebytes.org. They've been doing a good job keeping up with the latest iterations of this. Malwarebytes.org. A lot of people have good success removing it with this tool. MSRT will also remove it. And for an antivirus, you know, I personally don't think it's - I think it's a foolish economy to get a free antivirus when a better one is 30 bucks a year. But there are good free ones. AVG is a good one. AntiVir is from Free-AV.com. And there's Avast!. Those are the three biggest names. But frankly, I would spend 30 bucks and get a better one that's faster and more accurate.

Steve: And is going to be supported by its company and receiving viral signature updates in a timely fashion.

Leo: The reason AVG gives it away is because they want you to upgrade to their paid version.

Steve: Right.

Leo: So their free version, mostly it's just slow. It doesn't scan quite as fast. It uses more system resources. I like NOD32. That's what I use. But there's a lot of good choices out there. And I certainly think that, you know, spending 30 bucks a year is not too much to spend for an antivirus.

Moving along, question number two. This is from Jan Hertsens. Another Yubico question. He wonders whether Yubico might be missing the obvious. We're talking about the YubiKey security dongle. In your last podcast you mentioned the need to trust the verifier of the key to be able to authenticate a key. So why not have the key use public key asymmetric encryption? You supply the user with his public key via download or whatever, keep the private key on the device. This way any entity can independently verify a token, but can't fake it. Trust No One? That seems like a good idea. Would that work?

Steve: Well, let's talk about it. First of all, there are so many - so much feedback about Yubico and YubiKeys that we received that, I mean, I know we talk about it a lot. But it's just it's captured our listeners' imagination. So I thought this was an interesting notion. There are a couple problems with the idea. And so let's work through how such a thing would work. If the YubiKey, or some future variant of it, or a hypothetical dongle like that were to use asymmetric encryption, then the idea would be that one side of the asymmetry of the key, that is in this case we would call it the private key because it would be kept private, although I keep reminding users that they are interchangeable. One undoes what the other does, essentially.

So in this dongle would be the private, asymmetric key. And no force on earth could cause it to disclose the key. So the whole idea would be you need to prove that you own the thing that contains this private key. So that means that it would have to accept something that was encrypted with your public key that anyone could have access to, and then decrypt it using the other key, the private key, and then demonstrate that it had done so by returning that decrypted thing. So, for example, you go to a website that has access somehow to your public key. Like when you originally authenticated yourself with the website, you said hi, you know, I'm Steve Gibson. Here is my public Yubico - we need to have a different name, though, because I don't want to confuse people. It is not a Yubico.

Leo: This is some other Yubico-like dongle.

Steve: Yes. Here's the public key associated with my account and with my authentication device, and that's the key, so to speak. And this is who I am. So please authenticate me against this public key. So you would go through whatever it is you do to authenticate yourself the first time, giving the site that, you know, your public side of your authentication device. So then, every time you logged in, the site would, for example, just generate what we call a nonce, an n-o-n-c-e, basically a pseudorandom blob which it only ever uses once. It would encrypt that with your public key, that it and anybody could do. But the only way to decrypt it is with your private key. So you would then provide that that encrypted nonce to your authentication device. And it would decrypt it using the private key that it will never divulge. It will only do the work that that key allows it to do. And then it would return the decrypted nonce to the site, confirming to the site that whatever it is, you've got the matching private key that matches the public key.

Leo: It's like signing. It verifies your identity.

Steve: Yes. And this concept works. Here's two problems. First is, it means that you need data going into this device. That is, that encrypted blob has to go into the device. Which means that it can no longer just be a keyboard. And one of the elegant things about Yubico's solution with the YubiKey is it's just a keyboard. When you hold your finger over the little contact, it spits out the next little blurch of stuff.

Leo: It's not that smart, in other words. It's just a little...

Steve: Well, and that's part two, is it is really not smart. It's very inexpensive because it doesn't take much to do that. Suddenly now we're asking it to do public key encryption, which is very processor intensive. And so it would radically change the form factor, the cost structure and essentially the technology that you would need to have in there. Which is not to say that it can't be done. There's crypto chips all over the place. So it could certainly be done. Somehow you would have to get the data into the key. The only way I could think of doing that when I was brainstorming is there is communication for the lights on a USB keyboard, you know, the Scroll Lock, Caps Lock, and Num Lock. And so it is possible for the computer to get data into a keyboard by using those. And I've not looked at the protocol closely. There may be even a wider channel than just three lights' worth. But certainly you could serialize the key's binary bits in order to send that into the key, in order to get the data in. It would then process it and then spit out the matching crypto.

So it's certainly possible to do - I think you could solve these problems. I don't know if it could be as inexpensive as the YubiKey, and maybe in the future there will be something like this. As far as I know, it doesn't exist now. But I completely agree. This is cool because it does solve the problem of a third party. You give anyone who you want to be able to authenticate you your public key, and nobody who doesn't have your private key can do so.

Leo: Yeah, I love it. It's risky because you're carrying your private key around on a device.

Steve: It's actually no riskier than the YubiKey. The YubiKey has a secret key also.

Leo: That's true.

Steve: It just encrypts a counter in order to spit it out.

Leo: Well, and here's one other thing. The way typically your public key/private key works, isn't there a passphrase or some other way of matching yourself to the private key? I guess [indiscernible].

Steve: You could certainly add...

Leo: ...[indiscernible] signing, there is.

Steve: Yeah. You could certainly add authentication. And you could also even do it - I wonder if you could do it at your end, where you have to type in your passphrase that stays local; so you're authorizing the key to do the decryption, and that way we've now made it multifactor, too.

Leo: I think really the issue is totally cost. I mean, now you're putting a processor in here.

Steve: Yeah, it's, well, yeah, yeah.

Leo: You know, prices are falling. Someday somebody's going to do this. It's just how do you get it down to the nickel.

Steve: It certainly makes sense. And it's a cool thing to imagine that you've got your private key locked up in this little thing on your key ring, and you can - now you no longer need a third party. See, the point of the third party is they know your secret key.

Leo: Right.

Steve: But if you use public key crypto, you don't need that third party. You just need more power at your end.

Leo: Although ultimately - at least PGP works with a chain of trust. You always have a third party who's validating that you are you. Otherwise you can make up a key. I'm Steve Gibson. See, I've got the key that says it.

Steve: Well, no.

Leo: Somebody has to verify that that really is Steve Gibson's key.

Steve: Well, remember that I said we originally need to go through some sort of authentication.

Leo: Yeah. There's always trust somewhere, is what I'm saying.

Steve: When I'm giving them my private - when I'm giving them my public key. That is, the matching public key. And so you could imagine that this thing, if you, like, hold the button down for a long time, it spits out the public key that it's willing to give to anyone, anytime. But and that way, you know, it's like it's got them both. But it will never release the private key. It will only use the private key to perform a decryption operation on your behalf.

Leo: Right.

Steve: Anyway, it could absolutely work. It just, you know, it would require more processing power in the key. You really don't want to, like, let the key out even into the CPU. You could say, well, I've plugged my key into a PC. Modern PCs could do that in a blink of an eye. Yes. But then as soon as you let the private key out of the hyper-strength YubiKey, future YubiKey thing, then it's subject to compromise. So you never want to let it out. You only feed something in for it to do the work on. I mean, that's certainly an upgrade to the concept.

Leo: Thinking on - I have a question, thinking on this note.

Steve: Yeah.

Leo: Because I use PGP to create a public/private key pair that I use to sign all my mail so that people know it's my mail. Now, of course, anybody could do that, make a key that says it's, you know, Leo Laporte.

Steve: Actually all they know is that it came from your machine. Right? Somebody else could use your machine and be signing mail from you.

Leo: No, no. You could create - because of the nature of PGP, anybody can create a key with any address. But what you do is you ask people do sign it. So people who know it's you would then sign the key. And so the trust goes up. It's called a chain of trust. And it is kind of a flaw in the PGP system, unlike a certification system. If you get a certificate, you prove to Thawte or VeriSign that you are you and at least prove that you have that email address because they send a cert to that email address. Not with PGP. You're generating the key locally. So there's no central authority. But at the same time there's no central authentication, either.

But what you do is you ask people to sign it. In fact, they have key-signing parties where you'll go, and a hundred geeks will bring their PGP key, and I'll look at your driver's license, say yeah, that's you, and I'll sign your key. And then it says, well, Leo Laporte says it's him, and Steve Gibson says it's him, I guess it must be him. It's not tied to the machine. I carry my PGP private key with me.

Steve: The problem is we're mixing up certificates with...

Leo: Exactly. This is a different system.

Steve: Right, right, right. And so this is a certificate-free system where we're simply using the...

Leo: You know what it's like? It's a self-signed certificate.

Steve: Well, no, it's not a certificate at all. It's just crypto.

Leo: It is.

Steve: It's raw crypto.

Leo: But I'm saying, in terms of trust, it's like a self - you can make a certificate for yourself, self-signed.

Steve: Absolutely.

Leo: But no third party has verified it in any way.

Steve: Right.

Leo: That's what these are like.

Steve: So the idea is, if this next-generation key, if you bought one, it would have two modes of operation. You hold the button down for a long time, so that you don't do it inadvertently, and out comes your public key. So it's able to dispense your public key whenever you need it. You can write it down. You can store it in a text file. You could use to enter it into a website. Doesn't matter. Because, I mean, that's freely offered.

Leo: It's public, yeah.

Steve: It's public. Then as you normally touch it, it will wait for a something to come in. And when that whatever that is, that is an encrypted something, it will apply your private key to decrypt it and then type it back out. I mean, and that - if that system existed, it would be a tremendous authentication tool. And what it proves is, it proves you're in physical possession of that object which contains that private key.

Leo: Precisely, my friend.

Steve: Yeah.

Leo: Darren Tieu in Redwood City, California wonders whether a VPN is really necessary. Do I really need a VPN, he says? I'm using an open WiFi in a hotel. Is a VPN necessary? If I'm transmitting information I don't want anybody to see, shouldn't that be on an SSL connection anyway, you know, a bank, Amazon, whatever? Any sensitive information should never be transmitted over a non-SSL site, whether it's through a VPN connection or not.

I really don't see the need to spend money for a VPN connection. I don't care if people in the hotel know I'm on ESPN.com to see if the Cal men's basketball team beat Stanford. I don't care if people are reading my - oh, I'm sorry. I care if people are reading my email, but Gmail is through an SSL connection, so they can't see my email anyway, with or without a VPN. If I'm checking my balance at a bank site SSL, people can't see the traffic. I understand I need a VPN if I connect to a work network, but that's different than surfing on the web. In terms of protecting someone from hacking into my computer, a firewall protects me, but not a VPN connection. Am I correct?

Steve: Absolutely.

Leo: The rub is, not everything is SSL.

Steve: And even, as we know, Gmail is not SSL unless you explicitly start your session with HTTPS in Gmail. Many people just do Gmail.com, which defaults to non-SSL. You're taken briefly into a secure session for your login, and then you revert to nonsecure. So all of your Gmail is passing in the clear. It is certainly the case that, if someone is really vigilant with what they're doing and whether they're secure or not from moment to moment, I agree with Darren that a VPN in an open WiFi scenario is not necessary for typical use of the web, if he doesn't care what he's doing. But if he - most email is just standard IMAP or POP or SMTP, which is not over an encrypted connection. So email is classic for just being totally readable and, arguably, somewhat private, especially login credentials that are often easily captured, username and password for login.

So again, if you're really careful, then I agree, something like HotSpotVPN we were talking about, and this is probably what triggered Darren's question, where you wanted to be protected in an open environment, you know, you decide if you want it or not. Remember that two weeks ago we had the question from someone who was for some reason, or for a limited length of time, being forced to work in an employer's office that had open WiFi, and he didn't want to stir things up and cause a lot of ruffled feathers. So he was asking us how can I protect myself? I'm a Security Now! listener. I understand the importance, the danger of open WiFi. I want to protect myself. So it's like, there, something like HotSpotVPN which is very economical and available on one- or two-day contracts, or a week, made a lot of sense for him. If Darren is feeling VPN-hostile, then fine. I mean, I completely agree with him. You have to be vigilant, though, moment to moment, action to action, and aware that unless you are over a secure connection, anyone can see what you're doing.

Leo: You can in the Gmail settings now turn on Always Use HTTPS in the browser.

Steve: Good. I thought I remembered that they made some change.

Leo: Yeah. And that's highly recommended. I think the real problem is a lot of Internet service providers' email is in the clear. The password's sent in the clear. And so that's kind of the nightmare scenario, where you check your email, and then you leave, but the password's been sent in the clear. Now the guy's got access to your email until you change your email password. And how often do you do that?

Steve: Right.

Leo: And once somebody has access to your email, I mean, there are all sorts of threats. So, yeah, if you're vigilant. But that's always the case. If you're vigilant you don't have to worry.

Larry Strope of Streamwood, Illinois is describing his love/hate relationship with NoScript. Dear Steve and Leo, I've been with you guys since the beginning. I've enjoyed all the various topics you've covered over the years. Some topics have done fly-bys on my senior-sized, shrinking brain. Most of them have stuck in some fashion or another. All good stuff, mind you. And I've been a SpinRite user since v2 or 3. Wow. Can't exactly recall which. I still have an Apple II+. Will SpinRite work on that? Just kidding, just kidding, but not kidding about having an Apple II+. That goes back to the time before consumer hard drives, a.k.a. "The Chronicles of Apple."

My topic is NoScript - this is almost a Shakespearean sonnet here - and my love/hate relationship with it. I love the added protection it offers against script baddies, but I hate the added time spent in trying to decide which of the listed blockages I need to clear that will allow me to, say, push a button required to continue or accept or submit, among the list of allows that extends from the bottom to the top of the screen. When I look at the list, I have no idea at all which item is controlling the button I am trying to use. And frequently I find that certain links contained on a page won't work at all without making additional allowances. I usually end up in frustration saying to hell with it and allowing the entire page. That's what I always do. But doesn't that kind of defeat the purpose of NoScript?

Steve: No.

Leo: One would think that a modicum of common sense would prevail in these situations. After all, if I have chosen to sign up for a newsletter or register or make a purchase from a site I believe - a key word here - to be trustworthy, then one would think allowing the entire page would be safe. The offset is, however, that overall trustworthiness of websites has diminished considerably over the years, and a sense of wariness is usually present in the subconscious (clicking with your fingers crossed behind your back). This is particularly true when you see things like DoubleClick and Google Analytics showing up in the list. I suppose this has been more of a rant than a question. But do I need more caution? Less caution? A therapist? Thanks for a great series. What a great letter.

Steve: Yeah, I really liked it.

Leo: So this is what I had. That's the same problem I have. But I always just say, oh, allow the site.

Steve: Yeah. For me, okay, again, the recurring theme we have here is know the risks and know your options. And so I have no problem with Larry saying allow the whole page. NoScript is one tool that I haven't abandoned after I've turned off the popup notices, which really are annoying. I think the default for that should be the other way around. But I understand why a new NoScript user might want to have those.

For me, every time I run across a site that's got a problem, it's like, oh, okay, fine. And I just go down, and I allow it. I'm not running a personal firewall that manages all of my outbound traffic. I'm not running an antivirus. There are many different security solutions that get in my way more than I'm willing to tolerate, but NoScript is not one of them. I'm so cognizant of the tremendous ease of stumbling onto a site that hurts me without any ability to control it. I mean, I can control not clicking on links in email. I can control having a sense of responsibility for what software I've got installed on my machine and how it's phoning home and what it's doing. But I can't control ahead of time what a page that I'm about to receive when I click on a link is going to do to me.

So there's a real threshold in my thinking about scripting. And, you know, we all know that I've been anti-scripting for a long time. Clearly, the 'Net needs scripting more and more. And I don't - I can't argue against the functionality the scripting provides. It's so useful. Unfortunately, with that usefulness comes exploitability. So for me, I guess I would say think of NoScript as a tool, a useful tool that you can use any way you want it. But it's better to have it than not at all.

I just - who knows what kind of junk, I mean, remember the first Q&A was with, I think, Alesia, who got AV 2009 installed on her machine. Well, the way that original popup happened was scripting. The site she went to used scripting to produce a popup which had malicious intent, which then induced her, as you said, doesn't matter whether she clicks yes or no because a script is behind that, that took her to the website that performed a fake AV test. All of that is the fault of scripting. If scripting was disabled, she would have never had that trouble and would not have had that malware installed on her machine.

So I'm really bullish on NoScript. I just think it's the right tool. Again, I'm not telling everyone to run with it all the way up and fight with a site. It's up to you. But having that choice is what NoScript gives you.

Leo: Actually, I don't know if NoScript would have protected you against AV 2009. The way it works is, at least in one - in the instance I'm aware of is that somebody's website is hacked. Their .ht access file is modified to check the referrer. If the referrer comes from Google or Yahoo!, it replaces the page that you would - and this is all happening server-side - replaces the page that you would get with another page that says you've been hacked. So it - and that could also...

Steve: So it's not a popup window.

Leo: It looks like a popup window. It's just a sized HTML page.

Steve: That requires scripting.

Leo: You can't just size a page, huh?

Steve: No.

Leo: Okay. Yeah, because it does, yeah, I guess in order to trick you it needs to look like a dialogue box.

Steve: An OS dialogue.

Leo: Yeah. So that's done by scripting. Okay, yeah, you're right. Of course anybody who falls for that probably isn't running NoScript. Do you, so when you get to a site, do you say Trust All, or do you say one by one, okay, you can allow that script, allow that script?

Steve: I've certainly seen sites where the page has resources coming from all over hell and gone. I mean, it just, you know, you - there's George. You click the little blocked S for Script, and up pops a menu of just it lists all of these different domains that the site, that the page is trying to pull from. Well, first of all, there's a good clue that this is a sophisticated page. I mean, I would look askance at a page that was doing that. Most sites don't give you a huge list of domains. But I normally just allow the main page. And for me, 99.9 percent of the time it works.

Leo: Yeah. Yeah.

Steve: Or you could just say allow everything. Again...

Leo: I'm not saying allow everything, but I'm saying allow this site.

Steve: Yes, yes, allow this - allow everything on this page.

Leo: Right.

Steve: And again, even doing that, even that gives you a chance. It says wait a minute, you know, you've seen most of the page. Something's not working. I mean, my point is you've arrived, and you can evaluate. If you had scripting enabled by default, blanket scripting, you don't have a chance to evaluate. You don't have any opportunity to make a decision about whether you want to go further or not. I mean, and when you click a link, you're blind. You don't know where you're going.

Leo: Right.

Steve: Until you get there.

Leo: What a world, what a world.

Steve: Yeah, well, it's why we have a podcast.

Leo: Alistair Kidd in "Larbert," wherever that is, suggests that Steve's AxCrypt encryption advice requires a knuckle rap. Dear Steve, from the last SN Q&A when talking about email encryption, you mentioned AxCrypt." And he quotes you. He says, "So I would say" - you say - "So I would say, for somebody who just wants to occasionally send something encrypted, you just encrypt the file and email it and a little AxDecrypt program to a friend, or tell your friend to download AxDecrypt, which is also free." But saying you could send a binary attachment, expecting the recipient to run it? Doesn't that contradict all sensible advice about email attachments? I guess it does. If I got an email saying that I had to run an attachment to decrypt it, it would go straight into the bit bucket. Same with an email containing a link asking me to download an executable. Yours in flippancy, Alistair. P.S.: Any thoughts about Microsoft's Fix It button idea? Ill-considered? Rank rotten? P.P.S.: Love the show to bits. That is a good point.

Steve: Yeah. And I don't disagree with it at all. My assumption is that, if you're sending someone an encrypted file, you know them. It's your attorney. It's somebody you have some sort of a relationship with that requires encryption, and that that's a little bit off of the normal beaten path. So I would have no problem if someone said, hey, I encrypted this with this program, go get it. Especially if they were a trusted friend. Go get it from the site or put AxDecrypt, Google it, into the - run a Google query, find it yourself, I mean, whatever. Or you've arranged to have the program before, and it's what you guys use when you're sharing things back and forth. So, I mean, I guess I understand his position. But it seems to me that it's - the idea is it's someone you have a relationship with, rather than spam being sent out saying, hey, here's an attachment, go here to decrypt it, and the spam is coming from someone you don't trust and have no knowledge of.

Leo: Right, right.

Steve: By the way, there was some dialogue after this in GRC's Security Now! newsgroup indicating that there are flavors of ZIP which use AES encryption, which I was unaware of. And apparently not all flavors of, but some. But that's an interesting notion, too. So here's a mainstream ZIP program, and I don't know if it's the ZIP built into Windows. I will try to do some research about this because it would be terrific, for example, if there was a platform-neutral ZIP format where encrypting the ZIP actually did encryption, rather than putting the weak password protection on ZIPs that they used to.

Leo: Yeah, used to be really crackable, yeah.

Steve: Used to be very easy to crack it, yes.

Leo: Yeah. I use PGP. I actually use GNU Privacy Guard, which is an open-source PGP. And I sign all my messages with it. And I publish my public key. So if somebody wants to encrypt mail to me, they have my key. And they just use it, and they encrypt it, and they send it to me.

Steve: Yup.

Leo: I wish more people did that, but it's just too geeky. Nobody does it. Jonathan in Roseville, California resists the "YubiKey for static password" notion. Steve, you've been talking about the YubiKey from Yubico for a while now. I really like the idea. I use an RSA key for one of the sites I use, but the YubiKey would be much more convenient except that I usually get the RSA key over the phone. However, you've mentioned using it in static password mode, as well. I listened to your episode going into detail about the YubiKey, and I didn't hear anything that makes it sound any more secure than writing down a strong password and entering it on the keyboard accurately every time. You would be vulnerable to key loggers still, and this time it would matter because it is not a one-time password. Would this be more secure than just storing the key in a text file on a thumb drive?

For my wireless router I use one of your 64 hex character Perfect Passwords as an AES encryption key, then store that string in a text file, because you're not going to type it, on a removable drive. I then copy and paste it into the appropriate field when setting up a new wireless client. Is there something I'm not thinking of? I could even encrypt that file. That would add security. But I use this drive every day for school, and I don't worry about using it. I'd love to know what you think about this. Thanks for the netcast. Exactly what I love to listen to - in-depth technical discussions of IT issues.

Steve: Okay. Once again, I bring this up because there's so much traffic in our feedback about the YubiKey. And I've thought about why there's the amount of people questioning this notion of static password. And I think that I should have said something when we first discussed this that I never said because I sort of took it for granted more than I should have. And so I want to make it very clear, sort of once and for all, that when you change the YubiKey from its one-time password mode to static, you've completely changed everything. I mean, everything that was special and originally really cool about the one-time password mode of the YubiKey is changed. It's gone. Now what you have is something completely different. I mean, you can literally - and I wanted to be really clear about this - think of it as two entirely separate devices. You know, they live in one piece of plastic, and it's one or the other at a time. But, I mean, they really are completely separate.

So we've had a lot of listeners questioning the security of the static password. Well, I'm glad they're questioning it because they've been listening to the show. It means they understood what the nature of the coolness of the whole concept of a one-time password. He mentioned his RSA fob. I've got the little eInk credit card in my wallet which I'm able to use interchangeably with my football if I'm away from home. You know, I mean, all of these one-time password schemes are uniquely secure because they don't use the same thing twice. So they're fundamentally different from anything which is static.

And so I completely agree with Jonathan's comment that the YubiKey in static mode is nowhere near as secure as the YubiKey in one-time password mode. But it's not intending to be. It's not saying it is. It's a very different model. And once again we need to kind of come back to the security model and just understand what that is. So now with this static password mode we're generating 64 characters from a 16-character alphabet which has the entropy, the randomness of 256 bits, when the YubiKey is configured in static password mode with a maximum-length output.

So what it is, is useful for what it is. It's 64 characters, the same 64 every time. You can use it as your WPA key. You can use it as a password on a website. Every time you use it, you need to understand that it's not one-time password, it's the same password every time. But what you're getting is you're getting its extreme length and the difficulty of memorizing it as its benefit. And the ease of entering that monster-long, 64-character thing every single time, the ease of entering it just by touching your finger on the button.

So there are scenarios where you could argue that the risk is very low. For example, in preboot authentication with TrueCrypt, where there's no OS running, there's no Internet connectivity, you're like pre- all of those problems that you typically have. And you could add your own password after that in order to create a second factor, the YubiKey being one, with its monster-long thing, and then something you add to it. So again, very much like NoScript where, yes, NoScript has some problems, but it gives you some leverage. I think the YubiKey in static password mode, let's not think of it as the end-all, be-all solution. Let's think of it as, okay, that's useful for a certain domain of solutions different from what the YubiKey in its one-time password mode offers.

Leo: Okay. That makes perfect sense to me, as long as you understand the distinction. Right?

Steve: My rant is over now.

Leo: Okay.

Steve: They really are separate solution domains.

Leo: Yes. It's important to understand.

Steve: And this one little bit of plastic is able to function either way, which is I think very cool.

[Talking simultaneously]

Steve: ...applications for each.

Leo: He's right that you could store it on a USB key. You could do all sorts of things with it.

Steve: Well, in fact, until I did this, switched one of my YubiKeys to static mode and used it as my WPA password, which I have, when someone came over - I use my own Perfect Passwords just like he does, and I had that on a file in a thumb drive. And so when someone came over and wanted to get online with their laptop, I'd give them the thumb drive. They would get the file, open the file, copy and paste it into the password box twice, and then they were online. Now it's cooler. I've got...

Leo: Yeah, just give them that, and they go, boom.

Steve: Exactly. They go, like, what the heck is that? I say, ah, well, you're over here in the house of magic and mystery, so what do you expect? You know, we've got yabba-dabba-do going off all the time. It's loony over here. And so I stick this little sliver of black plastic in and touch it, and it goes zoop, and then do it a second time, and bang, they're now online. It's very cool.

So, yes, understanding where and how it makes sense to use it reduces this to a tool. Neither of them are perfect because, for example, we saw that the YubiKey in one-time password mode, being symmetric, requires a third party to authenticate. Okay. So that's useful for a certain domain of problems. And the static system doesn't require a third party, doesn't have public key encryption. But what it does is, is offers you a very complex string which is safe, you could argue is safe to use in an environment where keystroke logging is not a real threat.

Leo: Right. Mike Silvers in Salisbury, Maryland wonders why his NAT router isn't protecting him. Steve, I'm an avid listener and a computer consultant on the east shore of Maryland. I have a question about the visibility of internal LAN IP addresses through a NAT router. I visited the site IP-Lookup.net. When you reach the page, it gives you information about your WAN IP address and ownership of that address. What concerns me is the little link under the WAN IP address. When you click on the link, it shows the internal LAN address of my Apple Mac. I thought the NAT router would shield the outside world from determining my internal IP structure. How do they bust through the NAT? Should I be concerned about this? This is an old trick.

Steve: And I have one word, Leo.

Leo: Yeah. I know what word it's going to be, too.

Steve: Uh-huh. It's the S word.

Leo: Yup.

Steve: Yup. Scripting. Scripting.

Leo: In other words, the computer, your computer knows its IP address. So it just writes a script that says publish the IP address.

Steve: Yup, it's funny, when I was reading this, I said, I think, I mean, I knew what the answer was, just as you did. I went to IP-Lookup.net under Firefox, and it completely failed. It showed me my WAN IP. It did not show me my LAN IP. In fact, the link for that didn't work at all. It was completely nonfunctional. So then I said, okay, gee, I've got to enable scripting. So I deliberately enabled scripting. Even then it didn't work. And it turns out it's because it uses actual Java, and I've deliberately not installed Java for Firefox's use because I just don't think I need it, and I don't. However, I opened up IE and gave it a try under IE, and it worked perfectly. So it's like, yes, thank you anyway. It's scripting.

Leo: This is an old trick sites used to use to say your privacy has been compromised, and it would do all this Java script stuff and...

Steve: Even worse, what many sites did was, on IE, was they would show you the contents of your hard drive.

Leo: Right, right. I remember that.

Steve: Oh, my god, I can't tell you how many times, I mean, how much email Greg has answered where they use ShieldsUP!, and we'd say you're secure, and they'd say, hey, I went to this site, and it showed me the contents of my C: drive. It's like, yes, because your browser was told to show the contents of your drive. The site just gave your browser a link that said show C:\ and the browser does.

Leo: Just to make this clear, there's stuff that happens on the server side, and there's stuff that happens on your browser, on what's called client side. And your browser knows all that stuff. So your browser can be told to display that information. It doesn't mean it's sending it back to those guys, either; right? I guess it could.

Steve: Correct. Good question. It certainly could. I don't know in this case if the Java sent it - or JavaScript because either can do it - sent it to their server, and then they displayed it, or whether it just displayed it locally.

Leo: Oh, I'm sure it just displayed it locally. But my question is - yeah, because that's all it needs to do. but my question is, could it be used to - could you query, I mean, we'd have to ask a JavaScript wizard, I guess. I'm sure there - the theory is that it's sandboxed, and it's prevented from doing stuff like that. But there might be ways around that.

Steve: Yeah. I would be surprised if you couldn't incorporate the local IP, for example, into a URL query.

Leo: Exactly, something [indiscernible] like that.

Steve: And so have your browser request a resource that had the IP embedded in it, and the server could then capture that request and decode the IP from it.

Leo: You'd make a good hacker, Steve.

Steve: Well, I've got a lot of that technology over at GRC for, like, tracking users who don't have cookies enabled, where I'm wanting to offer them services and, like, keep track of who they are. I use that in the cookie forensics stuff that is soon to be made public in order to allow people to have everything blocked, yet I still maintain a relationship with them.

Leo: Jack Scharf has a good question from Longmont, Colorado. How is he supposed to know that auto updates aren't trojans? He says on Microsoft, whether it's Windows or Defender, Apple with iTunes or Safari, Adobe with Acrobat, Intuit and any other software vendor, alerts me to install automatic updates, how do I know it's really that company? Maybe it's just a popup saying you need a new update, and it's a trojan horse. With all the hacking and spoofing going on, this seems like an avenue into unsuspecting computers which is far overripe. What are vendors doing to prevent this? What can users do?

Steve: Well, it's a great question. Really the security answer is, if you're running programs on your computer, you have implicitly trusted them. That is, you've installed software from Microsoft, Windows and Defender, or Apple iTunes, Adobe Acrobat, TurboTax from Intuit. The point is you're assuming that they're going to behave themselves honorably. And in running them, essentially they have the full run of the computer. From that standpoint it's sort of a little surprising we're not having more problems than we are with programs misbehaving. Of course, if Apple iTunes did something bad, thanks to the communication network we now have with the Internet, the world would know about it very quickly. So vendors are doing the best job they can not to cause problems for their users because they recognize that it directly affects their bottom line.

Leo: Oh, but I think he's saying what it somebody posed as iTunes and said, I'm iTunes, I'd like to update.

Steve: Well, then you've got something in your computer which is working against your interests.

Leo: That can only happen because you've got somebody running on your system already, yeah.

Steve: Yeah. Now, the interesting threat is, what if somebody, for example, used a DNS spoof so that when iTunes tried to get an update from Apple, it actually got an update from a malicious site? So there they're intercepting a valid software update, and essentially commandeering it in order to get malware installed in your machine. And we know, for example, that Microsoft understood that problem, and so they've gone to some measures to cryptographically sign and protect all the Windows downloading stuff that is going on. And we hope that Apple and Adobe and Intuit and the others are doing the same, although there's no guarantee that any random company that wants to do automatic updates is taking every kind of security measures that they can.

Leo: Okay. And I guess in a way that's what the Antivirus 2009 thing was, which is essentially trying to pose as a dialogue box from a legitimate company to trick you into downloading something else.

Steve: Using a little bit of scripting, followed by a lot of social engineering. I mean, that was largely a social engineering hack, convincing people that, oh, look, you're infected, click this to get disinfected.

Leo: You'd know if it were, I mean, it would be hard to spoof one of those Microsoft or Apple windows; right? I would think. Maybe not. I guess if it's a Windows window, anybody can draw it.

Steve: Yeah, code running on your machine can look like anything it wants to, essentially. But by that time you've got code running on your machine. I guess the danger is...

Leo: I'm worried about scripting.

Steve: Yes, intercepting valid software's automatic update maneuvers and, like, getting your own stuff in instead. And you could imagine, if they weren't - if software vendors were not careful about the way they were protecting their own automatic update system, that could be a problem.

Leo: Right. Hasn't happened yet, to my knowledge. Todd in New York shares a very interesting VPN story that raised some questions. He says, guys, I love the show, loyal listener, so on, so forth. I have a service that is offered through my condo, Verizon Avenue, and I have been suspect for some time that they were throttling my service. Oh, his ISP is through his condo. I supposedly have 1MB down - which isn't very much - but for some time I've had certain music services, Rhapsody, that once worked quite well in this building, until all of a sudden I started to have horrible buffering issues where songs would stop in the middle multiple times in the song. I chalked this up to something that had changed on the side of the music provider, but today I learned otherwise.

I've been working freelance to set up an Astaro Security Gateway, yay, for a client that I consult for. In testing out the Astaro appliance that I installed in the client's office this past weekend, I noticed something odd. I was connected into their network over VPN, with all of my Internet traffic routed through the VPN, when, forgetting that I was still connected, I fired up the aforementioned music service. To my amazement it worked fine. So he's on the same connection. It's just VPNed.

Steve: Yup.

Leo: Songs flowed like butter without one hiccup, which had been more than commonplace before. It was to the point where literally I could not listen to a streaming song for more than 30 seconds before it would stop as the rest of the song buffered. Now, through the VPN, it was great. I could even download stuff in the background, still playing songs in the foreground. This has been a problem for a while, so I really noticed the improvement. So in order to test, I dropped the VPN connection, and my song-playing ability immediately dropped back to the abysmal zone. VPN back on, all was good.

So, two questions. First of all, is Verizon throttling to push their faster FIOS service? If so, tsk tsk. And secondly, does a secure SSL tunnel get around this problem? Obviously going through a tunnel I'm getting reduced throughput. After all, I have the same amount of bandwidth that I'm pushing through another network and then back out. But is the fact that this traffic is encrypted getting around their bandwidth-limiting countermeasures?

I love the show. SpinRite's saved me on numerous occasions. And Leo is the only reason I'm still an IT professional. Thanks for your insight. P.S.: The Astaro Security Gateway is a godsend and has made me and my client very happy and safe campers. We like to hear that. That's great. So that's a really good test. Bad bandwidth, run on a VPN, it's better.

Steve: Isn't that interesting. So what this - first of all, to answer his question, he's absolutely correct that if an ISP or anyone out on the Internet were trying to do application-specific throttling, where they're going to throttle only some traffic and not others, they need to see the traffic. And you cannot see traffic in a VPN because every packet is individually encrypted, so it looks like just - it looks like literally, as we've talked about often, encryption is noise. It is absolutely random noise. And so the VPN wrapper encrypts basically all of the packet, including the source and destination port. So the particular protocol that you're transferring, whether it's email or web or streaming audio, any ports associated with those protocols are obscured completely within the encrypted VPN tunnel. So if - and we don't know for sure that this is going on, that Verizon is doing this. But if they wanted to, they would bandwidth-limit based on some aspects of the traffic that their deep packet inspection would be identifying. And so they would hamper that. They are unable to do so if that traffic is being enclosed in a VPN tunnel. And, I mean, based on the evidence that Todd has shared with us...

Leo: Pretty clear, yeah.

Steve: It really looks like that's what's going on.

Leo: They're probably using some sort of packet sniffing or, you know, actually on streaming audio it's a different port. So they could even be watching the port.

Steve: That's all it would take. It would be saying, okay, we're just going to, like, give low priority to this port, to traffic on this port. Sure, we'll let people transit it. But if we're busy, and we've got other things to do, we'll just drop some of those packets and let them worry about getting them resent later, which is all it takes to throttle and cause problems for that kind of traffic.

Leo: That happens to our streams. We have an audio stream on port 80 as well as port 8000, so that just kind of eliminates that because you don't usually throttle port 80. That's the web surfing port. But our video, our TWiT Live video is on a different port. Although it's got some smart technology in there that will change ports, depending on how it's being handled. It ultimately ends up on port 80, if it has to.

John Ratzlaff in Candler, North Carolina wonders about eInk. I wonder, too. eInk. Hi, Leo and Steve. You have mentioned the new Kindle 2. You said it had 16 levels of gray. How are they doing that? My understanding of how eInk works is it consists of microscopic balls, white on one side, black on the other, which rotate in place according to the charge applied, which would result in either black or white. How do you get levels of gray? Dithering? Yes. Right?

Steve: It's interesting. If we're talking about, well, first of all there's not just one kind of eInk. There are a number of different technologies. The early stuff did use bicolored spheres...

Leo: Oh, really.

Steve: ...which were black on one side, white on the other. And they were suspended in an oil suspension and then rotated electrostatically so that their white front - the white side, the white hemisphere was aimed at the reader, or the black hemisphere was. So that was that technology. However, in the case of the current eInk that is used in the Sony readers and also in the Kindles, they use an entirely different approach. They have a high number of little black particles. And those black particles are pushed to the front of the screen or pulled to the back. And so you get a somewhat lower contrast ratio. But that's the technology they use is a very high number of black particles within each pixel region.

And in fact if you've ever had an occasion to look at the Sony or the Apple screen, for example, through a jeweler's loop or a good magnifying glass, you can see sort of dust. It's like not all the black particles obeyed their instruction to go to the back of the bus and get out of sight often. In fact, you and I have talked about how there sometimes is a ghost left behind. When you turn the page you can see sort of a dim ghost of the contents of the prior page. And that's literally - it's a little bit sort of reminiscent of phosphorescence and the way phosphor fades. But it's just that the greatest percentage of the particles did make the migration, but some didn't because it's sort of a statistical thing. So what they're doing in the case of the Sony reader and the Kindle is they're deliberately pulling different percentages of the particles away by having carefully designed their technology so that they're able to get shades of particle propagation within a single pixel. So it's a cool technology.

Leo: Very, very interesting. See, I never really was aware of how they did it. I thought it was that charged ball thing, too. So it's more like an Etch-A-Sketch.

Steve: Yes. It's very much like an Etch-A-Sketch.

Leo: With static.

Steve: With electrostatics, yeah.

Leo: Also, frankly, a little bit like how a laser printer works. Isn't it?

Steve: Uh, yes.

Leo: I mean, in effect, because it uses - it charges the drum, which attracts the toner.

Steve: And, exactly, and a laser printer has ink in the form of so-called "toner," which is super small little black particles.

Leo: Which everybody who has ever gotten it on his hands knows.

Steve: Very much like that.

Leo: So, yeah, they charge it. It attracts the particles in the places that you want. And of course a laser printer can do really good grayscale. We'll get ours in a week. We'll give you our review.

Steve: Yes.

Leo: Probably not in time for the next Security Now!, but the one after. Regina Gannaway, Great Mills, Maryland, wonders if the Sony PRS-505 is more secure than a Kindle? It's Sony's answer to the Kindle. Actually it predates the Kindle. Dear Steve and Leo, love the show. On Episode 183 you both expressed your love for the Kindle and your not-so-glowing opinion of the Sony eBook Reader. I have a security-related question, reason, rather, why I prefer the Sony eBook Reader to the Kindle. My main reason for purchasing the eBook Reader from Sony is to have a compact way to read documents for my job. They're usually in Microsoft Word format or PDF format. Putting them on my Sony eBook Reader helps reduce the weight of how much I have to carry around to read in those spare moments here and there, not to mention saving trees. I also convert PowerPoint documents to PDF, read them on the reader, as well. Big lifesaver there.

My Sony software allows me to just drag and drop these formats onto my PRS-505 reader. The document never leaves my control. My understanding is the Kindle requires users to send the document to Kindle.com, where they convert it to their proprietary format. Then they will send a newly formatted document back to the Kindle device. I can't do that. I work for a government agency, and I can't email documents to a third party to be converted to work on the eBook Reader. I have to be able to maintain control of the documents for various reasons. Usually it's proprietary information. Therefore I like the fact that I can transfer a document to my eBook Reader without losing control over it. Don't you think this is a good reason to recommend the Sony eBook Reader?

Steve: Well, I know you know the answer to this, Leo.

Leo: I do.

Steve: Yup. And I just wanted to put Regina's mind at rest. You can do exactly the same thing with the Kindle that you do with a Sony.

Leo: You've done a lot of research on getting stuff onto both the Kindle and the Sony.

Steve: Yeah. When you plug the Kindle, which has a USB connection, into your computer, it switches it into drive mode, and it looks not like a reader, but just like a drive, just like a thumb drive to your computer. And all your documents are there. You can browse them with Explorer. You can easily drag and drop documents onto the Kindle. Then when you pull it off and essentially disconnect it from the computer, it looks through to see what you've done and registers them and puts them on the Kindle's table of contents. So essentially you're able, again, to do exactly with the Kindle what you can with the Sony.

Leo: Now, the Kindle doesn't read PDF format, but you can convert the PDF to the AZW format that Kindle uses. What do you recommend? Have you tried that? What do you recommend to do that?

Steve: There's a - gee, it's been so long now since I looked. There's Mobipocket is the format.

Leo: That's it, okay.

Steve: And the Mobi - there's a Mobipocket - last it was a version 4.something. And that was the authoring tool, and it does a great job, and it's what Amazon uses. And they bought Mobi at the beginning of their whole eBook move. And so, yes, you're absolutely able to create content for the Sony. I mean for the Kindle.

Leo: Okay. And it just requires downloading some software to get...

Steve: Yup. But Regina's concern was that she was losing control of it for legal reasons. And in this case at no point is it out of her control.

Leo: We liked - we didn't pan the eBook Reader, by any means.

Steve: No, no, not at all. I mean, we both had both versions of the Sony and loved it until the Kindle came out.

Leo: And it's mostly because of the wireless, right, that we like the Kindle better.

Steve: Specifically because of the wireless. I mean, for me that did it. And in fact there was an article in this week's Economist where they were wondering whether eBooks might be the salvation of newspapers because so many people are reading newspapers now on their Kindle because of the WiFi. I mean, the fact that you can just, I mean, well, the wireless, the cell system, that you can turn it on and get all your magazines and newspapers updated in the morning and then read them all day. It's just spectacular.

Leo: They keep adding - they just added The New Yorker, which was really my one true dream.

Steve: Oh, no kidding, I didn't know they had The New Yorker, oh.

Leo: Yeah. That's really - we get - we subscribe to it. But between my wife and my daughter, I never really get a copy of it. When Abby, you know, she's in France. But when she was home she would just literally, as soon as it would come, she would take it upstairs, and we'd never see it again. Which I really couldn't very well complain about. I mean...

Steve: No, because she's...

Leo: She's reading The New Yorker. That's great, yeah.

Steve: Yeah.

Leo: But so now my wife does kind of the same thing. But now I have my copy on my Kindle. And I get the Atlantic on it, and Salon.

Steve: Yeah, I want The Economist. The Economist is still...

Leo: I would love it. But, you know, the fact that The New Yorker has jumped indicates to me that others will. You know, there was a great study. Somebody did a little math on the costs of printing The New York Times, and estimated for this - The Times has said we have about 800,000 people who've subscribed for more than two years. This is the dedicated group, the regular subscribers. Somebody estimated that the cost for this 800,000 subscribers was about $600 million a year in paper, ink, and trucks.

Steve: Wow.

Leo: And then figured, you know, you could give each one of them a Kindle and deliver it to the Kindle, it would cost half as much in one year. It would cost $300 million.

Steve: Wow.

Leo: Yeah. That's the underline on this. I mean, I prefer a paper. But...

Steve: And newsprint is just gross, too. I mean...

Leo: I mean, I like reading it. I like spreading it on the table and reading it. But, you know, those times, it's just too expensive.

Steve: Yeah.

Leo: Mark Smith in San Luis Obispo, California, makes some very good points about WPA WiFi security. Our last question, Steve. He says, Hi Steve and Leo. I just finished Episode 182. You guys really harped on open access points, complaining about the security or lack thereof when you don't have WPA turned on. I would argue that every link in the path between you and whomever you're talking to - mail server, web server, IRC, whatever - is untrusted, not just the final wireless link. If there's something you consider to be sensitive that should be protected, protect it at the application layer: SMTP over TLS, HTTPS, IMAPS, et cetera. If you're already doing this, then the data on the wireless link is already protected. This is kind of like our question about the VPN.

Steve: A little bit, yeah.

Leo: There are other things that WPA gets you, most notably access control to your access point, for which it's very well suited. But in many cases an open AP is precisely what you want: a coffee shop, a visitors network in an office. If you're correctly protecting your application layer, you shouldn't be afraid of using an open access point. Let me know if you disagree or if I'm missing something.

Steve: Well, again, like I said, Mark Smith makes some very good points about WPA WiFi security. And the only lesson here for our listeners is, once again, be aware of your situation. For example, Starbucks, in my case, dropped their T-Mobile relationship, which ended officially at the end of the year, in favor of AT&T. There was an overlap of about, I don't know, four or five months when they announced that they were going to be switching to AT&T service. T-Mobile was secure. AT&T is not. It's wide open. And, you know, I've been thinking, you know, I ought to just put a WiFi sniffer on and hang out at Starbucks for a few hours and just sort of see what's going on, just as a little bit of a - as a lesson to why open WiFi is a problem. Because I bet, with all the UCI students that are there, it would be an eye-opening bit of sniffing of traffic. And I don't know that any of them are aware that there is no encryption. They have to log into the AT&T. And they may think, oh, I'm logging in, I'm encrypted. But there's no encryption being used on the radio now in Starbucks stores. Which I think is a real problem.

Again, if people knew, that's fine. But it's matching up the vulnerability and the environment to what it is that you're using that environment for. And so I understand Mark's point that WPA is - obviously it's super useful for authentication because it's very strong authentication. And open WiFi has a place, as long as the dangers are properly understood.

Leo: Yeah, that's the key. And I think people, as you say, people don't really. So I use WPA for those two reasons. One, to keep people from using my access point.

Steve: Right.

Leo: And there's a real problem with that because, if your neighbor is using your access point and doing something illegal, you're liable.

Steve: Right.

Leo: The ISP's going to contact you, not him.

Steve: Well, and we've talked about how that's something where MAC address filtering, which is available probably now on all consumer routers, MAC address filtering, if you really wanted to leave your WiFi open for some reason, MAC address filtering would prevent inadvertent use. But it would not prevent someone deliberately saying, oh, look, I really want to use Leo's open WiFi. And gee, he seems to be only protecting it with MAC address filtering. So they just capture some packets out of the air and use - switch their adapter to use one of the authorized MACs. And then they're on your network.

Leo: Right. And then of course the encryption, in case I forget to encrypt at the application layer.

Steve: Yeah. Again, it seems to me - okay, so here's the rule. If you have no reason not to be running WPA, run WPA. If you have no reason not to encrypt...

Leo: Right, why not?

Steve: Exactly. Absolutely encrypt. So I would say it's always a good thing. It gives you authentication. It protects your information. You don't have to worry about whether you're running over a separate TLS, SSL, HTTPS, IMAPS, you know, blah blah blah. That's, you know, the wireless aspect of your work is encrypted. And we know that having the most security possible is more security than not.

Leo: Than not.

Steve: And so, and again, sure, if there's an application where you can't have encryption - and maybe Mark is also reacting to the question that was asked a couple weeks ago by the guy who was having to work in an office where their WiFi was open. So he was saying, look, I don't want to give all of them a hard time. I just want to protect myself because I recognize somebody outside could be sniffing my traffic, and I'd rather they not. So for him, using a VPN temporarily made sense. And so he was in an environment where he could not use WPA security. So I would say, if you can, by all means do it. Aside from a little bit of inconvenience for having to give everyone the WPA key, it just makes sense to do it.

Leo: I will give you another example of risk. FTP, unless you're using secure FTP, SFTP, you're sending passwords in the clear. And a lot of spy cam, you know, I used to use a program that would send a picture of me every 15 seconds to the server, and you could watch me. Before we were doing all this. Most of those don't use SFTP. So you're sending every 15 seconds, you're sending the password to your server in the clear. And in fact some of the uploading that we do is not - I do SFTP and SSH whenever I can. But some of the servers we use, the commercial servers that we upload our podcasts to, for example, don't use SFTP. So if I used them over an open WiFi access point, I would be - people could go in and delete podcasts and all sorts of stuff.

Steve: Well, yeah. I think there's two classifications of attack. There's opportunistic attack, and there's directed and focused. Opportunistic would be somebody sitting with a laptop at Starbucks, sucking in all the traffic and seeing what they can collect. And then a directed attack is somebody who decides, hey, I'm going to get Leo. I mean, for whatever reason, I want to do that. And so they'd arrange to sniff all the traffic on your various links, looking for that one time when, for whatever reason, somebody was doing a transfer who didn't know how to bring up an SSH tunnel first, or something like that, and they said aha, we caught some information in the clear. Now we're going to be able to leverage that in order to go further. So again, being as secure as you can all the time I think is what's most prudent.

Leo: I agree. Why not?

Steve: Yeah, exactly, why not? We have the tools. Use them. We have NoScript. Turn it on; turn it off when it's in your way.

Leo: I can think of a reason not to use that. But that's another matter entirely, for another day. Ladies and gentlemen, thank you so much. Steve Gibson, he's done it again, another marathon episode. But there was a lot to talk about today. Eat it in two bites if it's too much for you to digest in one. You will find transcripts online at GRC.com, along with 16KB versions of this file to save bandwidth. And of course all of Steve's great software, starting with SpinRite, the world's best disk recovery and maintenance utility, but also free stuff like ShieldsUP!, Shoot The Messenger, DCOMbobulator, Wizmo, and on and on and on. GRC.

Steve: Oh, and Leo, a very cool tool coming. The DNS benchmark I'm working on, that's where all my time has been going when I'm not on eBay.

Leo: When is SpinRite for PDP-8 coming out? That's what I want to know.

Steve: It's, yeah. This little DNS benchmark is going to be a popular gizmo.

Leo: Oh, neat.

Steve: It's just - you run it. It sucks in a list of, well, it contains a list of all the known public DNS servers and instantly, well, very quickly profiles their performance versus all the ones that you're currently using, and ranks them with graphs and statistics and things. Oh, it's going to be very cool.

Leo: You probably program like some people do crossword puzzles. It's just your recreation.

Steve: Yeah. I just love it. I mean, I just, yeah.

Leo: That's what you do.

Steve: I really do, yeah.

Leo: Always great, Steve. Thanks so much.

Steve: Even ancient computers with wacky octal codes and 12 bits. It's like, whoa.

Leo: I can't wait till that episode. That's one week...

Steve: It's going to be a great episode.

Leo: Yeah. That'll be a week from today, Thursday the 26th at 2:00 p.m. Pacific Time, 5:00 p.m. Eastern Time on Live.TWiT.tv. But we will also take the audio of it and put it out as a Security Now! Special.

Steve: That's a perfect idea.

Leo: Yeah. Why not? Why not? Thanks everybody. We'll see you next time on Security Now!.