Listener Feedback #30

Description: Steve and Leo discuss questions asked by listeners of their 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 they have previously discussed.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-122.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-122-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 122 for December 13, 2007: Listener Feedback #30. This show and the entire TWiT broadcast network is brought to you by donations from listeners like you. Thanks.

This is Security Now!, the podcast where we teach you how to protect yourself, your security, and your privacy, as we learned last week. Steve Gibson is here from Irvine. Hi, Steve.

Steve Gibson: Hello, Leo, great to be back with you.

Leo: Good to talk to you again. We've got - it's a question-and-answer session. So we're going to do something fun this time, aren't we.

Steve: Well, yeah. We actually had some great questions that I just really liked a lot. And we sort of - we had a couple questions pulling up the rear that were really fun. And the first one I ran across, I had to just call it the Horrifying Show-Stopper Question of the Week, which is our last one, #12. And then someone had a great idea which I thought, well, this is the Great Tip of the Week. And someone else had a really fun, anecdotal story about Perfect Paper Passwords. So I thought, well, that's the Perfect PPP Quandary of the Week. So those will be our final three, will be that little lineup.

Leo: I like that. That'll be a lot of fun. And we don't have any addenda from our privacy

discussion, except that I think we probably sent a chill down the spine of more than a few people. But it's good to be aware of, and that's the point. It's not that necessarily there's anything you can do. And certainly are some things you can do, but ultimately you may not have a lot of choice without dropping off the grid. But at least it's good to be aware of and make those choices where you can.

Steve: Exactly.

Leo: Any SpinRite news?

Steve: I did have one real short little fun anecdote from a lifelong Mac user.

Leo: Uh-oh.

Steve: And of course, you know, SpinRite - yeah. SpinRite has always been PC only. And in fact it's tied to the PC because it still uses the BIOS, the Basic I/O System, BIOS, that was part of the original IBM PC specification. And of course the Mac, the Intel Macs, they use something called EFI, which it serves the same purpose. That stands for Extensible Firmware Interface. But the two are not compatible. But anyway...

Leo: There's no INT 13 call in EFI.

Steve: Precisely. And believe me, SpinRite uses that like crazy.

Leo: But there must be some comparable call.

Steve: Yes. Well, I have to say there has to be, although I've never even gotten around to looking at it.

Leo: And you'd better, because I think EFI is not something Apple made up, it's a Microsoft Intel specification.

Steve: That's exactly right.

Leo: So eventually PCs are going to be using it, too, I guess.

Steve: There may be ultimately some migration.

Leo: Although, you know, this is what keeps PCs from evolving is this install base, is all these people like you who've written to BIOS, and they can't change it. So maybe they never do change it, I don't know.

Steve: So anyway, Phillip Shiver, he describes himself, he says, as a lifelong Macintosh user and PC technician. "I want to say thanks for SpinRite. When nothing in the Mac OS world can recover lost data..."

Leo: That's true.

Steve: "...I depend upon any available PC and SpinRite to fix Mac drives."

Leo: There's no analog to SpinRite in the Mac world. There are data recovery utilities. There are disk, you know - because SpinRite doesn't work at the file system level. So there are file system recovery tools.

Steve: Correct.

Leo: There are unerase tools. But as far as I know, there is no disk-level tool like SpinRite on the Mac side.

Steve: He says, "I started using it at work in 1992 to" - so what's that, 15, 16 years - "to recover data on" - get this - "on full-height 5MB drives, and also from floppies. And SpinRite has been making me look like a hero ever since. No IT department or serious Mac technician should be without SpinRite in their toolbox."

Leo: But you do need a PC to hook it up to.

Steve: Yup. He says, "Now, if we could just get a Mac-bootable native version to enable the use of SpinRite for regular maintenance, my world would approach nirvana." So his best regards, Phillip Shiver. So for what it's worth - or, I'm sorry, Shivers. Phillip Shiver. Noted.

Leo: Yes. But there's nothing we can do.

Steve: Not at the moment, but...

Leo: There's got to be an analog because of course that's what things like VMware and Parallels do is they map those INT 13 calls and other BIOS calls to EFI.

Steve: I'm sure there is. And the fact is, with SpinRite 6, I removed a lot of SpinRite's BIOS dependence, but not all of it. So I - it is in Assembler, so at some point, probably 6.something or 7, I will make that jump. And it'd be really fun to be able to make a bootable ISO that just - from which you burn a CD, and it just boots on a Mac.

Leo: Well, you do make one for PCs.

Steve: Yes.

Leo: That's what the PC does. But we should be cross-platform. It should say, oh, it's EFI. I'll do it this way. Oh, it's BIOS. I'll do it this way. You can do that, Steve.

Steve: I can do it, Leo.

Leo: All right, here's what we're going to do. Anybody who has an easy way to map an INT 13 call to EFI, you know, one we do for EFI, send me an email, and I'll make sure Steve sees it. You don't have to field these. No, no, no, you don't have to field these. But there's got to be somebody who's sitting there saying, well, is it just the INT 13 call or the other calls you need?

Steve: It's INT 13 screen and keyboard. So it's...

Leo: Oh, it's a screen and keyboard, yeah.

Steve: Keyboard, screen, and the drive.

Leo: I'm sure the screen and keyboard's pretty straightforward. I mean, they have a pretty good toolkit for on the Mac side. And I used to write Assembly language code for the Mac using the toolkit. So it's not undoable.

Steve: Yeah, maybe even a little translation layer, you know, [indiscernible] the drive or something. Anyway...

Leo: You would make a lot of Mac users very happy, Steve.

Steve: It'll happen.

Leo: Really?

Steve: Oh, yeah. But no one knows when.

Leo: It may be your grandson that does it. Okay, it's that kind of open-ended. Shall we get right to the questions?

Steve: We should. I did want to ask our listeners, if anyone had intended to go check out my Kindle review over on Amazon but forgot last week, I wanted to remind people that it's snipurl/skr [snipurl.com/skr], stands for Steve's Kindle Review. I would love if people - if we could use the clout of the size of our listenership to help me boost this review up in popularity so that people can actually see it. It's many pages down. And the ones that were there first are the ones that typically get seen. So there's really no way for an unseen review ever to get itself voted higher unless something like this happens. So...

Leo: And yours is very thoughtful, and you took the time to review it after you had used it for a week, unlike many of these reviews, some of whom have never used it at all.

Steve: Well, the reason I wrote this thing was that there were more people giving it bad reviews and one star than positives. And none of the people with one star own it. I mean, it's not like...

Leo: It's like politics. It was, like, political.

Steve: Yeah, it was like, well, I just know I hate this because it doesn't do, you know, HD-DVD. And it's like, yes, that's not what it is. It's an eBook reader. So, I mean, there were just - it just drove me nuts that something which I think is going to change the industry is just getting panned by people who are posting reviews when they don't own it. I mean, so it's not a review. So anyway, I wanted to ask our listeners if they would go to snipurl/skr [snipurl.com/skr]...

Leo: How did you get that special skr?

Steve: The snipurl.com site. Oh, I wasn't saying dotcom, snipurl.com/skr. It allows you - it'll assign you a random token, or you can give it one. And if it hasn't already been used by someone, then it'll assign one that you provide. So it's just - it's very handy because it allows you...

Leo: I didn't know you could give it your own. I thought you had to just take the random one it assigned you. That's really great.

Steve: Isn't that neat? Yeah, it's really nice. So again, it's snipurl.com - I think I forgot to be saying dotcom - snipurl.com/skr. And if you...

Leo: I don't know if it's me, Steve, but I have tried now, and there is no button I can click to vote for you. I see the text that says, if you found this commentary helpful, click the button below to help its ranking.

Steve: Could you have clicked already? Because I think if you click it once, it won't allow you to do it again.

Leo: Oh. Well, I don't think so.

Steve: Yeah, I don't think so.

Leo: Maybe I did. I mean, maybe I read your review, and I was unconscious and clicked it. Let me try it on Internet Explorer on the Mac. No, I don't see, you know, it says, "Was this review helpful to you?" And there's no button. There's something - I'll come back. There's something wrong with Amazon right now. Maybe it's just me. So but, you know, I'm getting my Kindle any day now, and I will add my thoughts, too. But like you, I want to use

it for a while.

Steve: Yes.

Leo: And we're both serious eBook readers. So we know what to expect. We've used this screen before on the Sony.

Steve: Yes. And Leo, the thing you said to me, the reason, the motivation you had to buy it - it was either you said it to me, or maybe I listened to it on TWiT or on one of your other podcasts, but you said it's the connectivity. You subscribe to a whole bunch of newspapers. You've got them lying all over the place. And the idea of the connectivity - and I've got to tell you that it is really comforting. It's something about the fact that it's just - it's connected into the world. And I've subscribed to a magazine and two newspapers - no, I'm sorry, a newspaper and two magazines. And it's really cool. They're just there in the morning.

Leo: Yeah. That's what I'm very interested in trying out. And I'm going to cancel my Chronicle and Wall Street Journal and, I mean, I have so many - I wish The New Yorker came out on this. Man.

Steve: Well, and I read The Economist every week.

Leo: Is The Economist on it?

Steve: No, not yet. But I have the same wish you did. I mean, if The Economist is on it, bang.

Leo: I'd buy that.

Steve: I'm canceling my paper subscription instantly.

Leo: In a heartbeat. And I am going to get The Nation on it. But I wish, you know, and maybe these magazines are sitting back and watching to see. But I would love to see The New Yorker and The Economist. And then I wouldn't feel so guilty about all that paper I waste every week. Because sometimes I don't get to read it, like most of the time.

All right. I am going to read the questions. Now's the time. Listener Feedback #30, starting with Perry Harris of Bluffton, Indiana. He needs to be more careful when he sits down. Does he have one of those exercise balls he sits on, too? He says, we've used RSA SecurID key fobs and credit card fobs for a while now. We've learned that both need to be treated with care. The key fobs can stop working if you bang them around with keys on your key ring. The wallet card should never be put in your wallet - uh-oh. That's where mine is. If you place your wallet in your back pocket, the act of sitting on the wallet causes enough pressure to curve it and cards within, and that breaks the card by cracking the LCD - actually it's an eInk screen. I don't know if VeriSign's card has the same problem or not.

Steve: And that's...

Leo: Oh, his are LCD.

Steve: Exactly, that's why I brought this up is RSA's cards are - you have a glass LCD screen.

Leo: Oh. [Indiscernible].

Steve: Yes. And I wanted to make sure that people knew that VeriSign's cards are different. And mine's in my wallet, and I plop my butt down on hard chairs without a second thought. I mean, it is amazing that this VeriSign card is as flexible as it is. It is, I mean, there's nothing about it that is stiff or unlike a regular credit card. And I've had mine in my wallet. I take it out and show it to people every so often. And it just - it is really robust. So I wanted Perry to know that the VeriSign cards are in fact different from the RSA cards in that you really can carry it in your wallet with confidence.

Leo: You know, it's not good to sit on your wallet, though. You can get sciatica.

Steve: Okay, well...

Leo: Keep your wallet in your hip pocket.

Steve: Good idea.

Leo: Unless you have a really thin wallet. Let's see. I have a thick wallet. I have lots of crap in it. All right. So, good, that's good to know. There's a difference between the two cards.

Steve: It really is, and it really is flexible. That VeriSign card is really built to last. And they have a three-year warranty on it. So if you did do something, they'd replace it for you.

Leo: I love mine. I don't use the fob anymore because the card's with me always.

Steve: Yup, exactly.

Leo: That's the whole point, I can keep it in my wallet.

Steve: We call it the football, Leo, by the way. That is the official name now. In fact, in these questions you'll see people referring to it as the "football," which is I guess the name we came up with. I think it's perfect.

Leo: I like it. Yay. Jeff Parsons of Issaquah, Washington had a PayPal/DoubleClick observation. We talked about this a couple of weeks ago, very big deal. Couldn't PayPal be even sneakier by linking their Apply Now button straight to the target PayPal page, instead

of having an image file like a GIF on that target page that's hosted by DoubleClick? See, that's not how it's - well, anyway. The URL to that image file could be similarly encoded with a PayPal user-specific ID, which DoubleClick could record and then simply return the expected image. Assuming browsers transmit cookie data for image GET requests just as they do for user-initiated GET requests, it seems there are many ways the privacy of even the most vigilant user would be at risk.

Steve: Yeah, there was a couple things here. First of all, Jeff has described the common practice of putting DoubleClick-sourced ads or images on a destination page, which then causes your browser to go fetch that image. And we know that, unless explicitly disabled, the so-called third-party requests, that is, requests out to a third-party server, are cookie-based requests. It also turns out that the vast majority of people are allowing their browsers to do this. I have some technology on the GRC site which has been in place now for quite a while, I think at least a year, which is generating statistics on the number of people who are browsing with third-party cookies enabled. And it's something like 89 percent of all the people who come to GRC, whom you would expect to be a little more security conscious and privacy conscious than your typical Internet user, they've got third-party cookies enabled. So it is the case that what Jeff is describing is the typical way that this is being done. However, the only reason I can see that PayPal would be explicitly linking to DoubleClick, as we described in our discussion, what, three weeks ago, is that they want to avoid users who disable third-party cookies by making it a first-party request, which is, as we described, the whole point of this.

Leo: Yeah, yeah. But I guess you could do other sneaky things. Because we talked about how you could cut the URL up and hand-write the URL and avoid that redirect. But nobody's going to do that. I don't even do that.

Steve: Yes. And I think Jeff's point was that you could also put some data in the image URL, like the user-specific ID stuff could be in the image URL, so it's not just a generic ad fetch, it's an ad fetch with additional information, which is absolutely true. There's a little bit of a worrisome next-generation feature that's not well known yet. It's in the next HTML spec. At least it's in 5.0. I'm not sure if it's in any of the 4.x specs. But it's an attribute in a link tag, in a standard tag, called "ping." And believe it or not, Leo, it is specifically for acknowledging to third parties when you the user have clicked on an object on an HTML page. It literally sends a ping fetch off to another third-party site, just saying, oh, just to let you know this link was clicked. It's like, oh, my goodness. Well, so much for privacy.

Leo: You know, they've just done everything they can to make it easier for them to invade our privacy. Corby in Reno, Nevada isn't so sure that twice is better than once. I had this question, too, actually. I'm glad Corby asked.

Steve: Well, Corby - well, yeah, okay. You read it.

Leo: In Episode 120 you answered a question regarding double encryption. You were asked if encrypting a file twice would provide twice the protection. I don't think your answer was entirely correct. I have no proof, but I'm willing to bet if you encrypt a file with Key A, and then again with Key B, the resulting file would be no more secure than a single encryption. I would guess that a brute force attack would find a third key - oh, that's interesting - that would decrypt the message. If two very weak keys were used to encrypt the original message, I would guess that a brute force attack would reveal a weak third key that would decrypt the cipher text.

And in the same vein, Mat Ludlam of Wybridge, London, England says: If we take some clear text and encrypt it with Key X and then Key Y, one way of decrypting it is to reverse the process. But isn't it possible that Key Z would work alone? Or Key Zed in his case. For example, XOR functions - actually that's right. For simple XOR functions, this logic works. But I've never looked into how proper encryption algorithms work.

And also Carlos - boy, you got a lot of feedback. So Carlos Gonzalez in San Jose, Costa Rica - all over the world, I might add - says in Listener 120 - Listener Feedback 29 you answered a question from a listener regarding the use of double encryption in order to protect from a brute-force attack. You and Leo endorsed the idea, saying it would make it more difficult to brute-force decrypt the text since you'd have to figure out both keys, and you wouldn't know if you've got the first key since you get random stuff back. Seems to me if you're using symmetric encryption to double-encrypt a text, first with Key A, then with Key B, the resulting text would be decryptable by using a new key, let's call it C, which would be equal to A XOR B. It's kind of interesting. It's kind of the same thing that Corby was suspecting without the mathematics behind it. So double-encrypting a text doesn't make it any harder to decrypt using brute-force methods. It just creates a new, unique key which will decrypt the text entirely. So you're right back where you were. Is that true?

Steve: No. It turns out - well, but so many people asked the question. First of all, I was delighted that people were thinking...

Leo: They're using their thinking caps.

Steve: Exactly. I mean, obviously it was a question and answer that interested a lot of people because I just chose three out of a great many more people who suggested the same sort of idea. What we know about symmetric encryption is that it is fundamentally different than, for example, the encryption we talked about early on in the Security Now! series when we were talking about the weak encryption that was provided by WEP, and actually is the same encryption that is used by WPA, the good WiFi encryption, except that the algorithm has been fixed so that it's not broken.

But our listeners may remember that back then the way that encryption worked was that there is a cipher known as RC4, which is a pseudorandom data generator. So you give it a key, and you get it started, and this thing produces a series of pseudorandom bytes which are so good when you use it correctly that it takes, it's estimated, as I remember it's about a billion bytes, that is, a gigabyte of data, even to be able to determine that it's not completely random. So it's really, really random. And we know that if you mix random data with nonrandom data, the result is random data. That is, if you XOR, which is the operation that a couple of these guys talked about, basically it randomly inverts bits in your data. And that results in something just as random as the original random data, even though it sort of, well, it does have the plaintext encoded in it, such that, naturally, if you reinvert the same bits, you'll get your data back. Well, that is a form of symmetric encryption, but that's a symmetric stream cipher, or symmetric stream encryption, where you mix, using XOR, you mix your data with something pseudorandom. And then when you mix it again, you get your data back.

Block encryption, that is, a block cipher, which is for example what Rijndael is using - and I don't remember now whether the person who posed the question specified which type of cipher he was using. I assumed he was using a block cipher. Block ciphers work completely differently. And the math of them, which we can explain sort of at a high level, shows why this notion of double encrypting really does work. Remember, as we talked about symmetric block ciphers, you have a chunk of data that is relatively wide in terms of its bit size. In the case of Rijndael, it's 128 bits. And so you put 128 bits in. And what comes out is a different 128 bits. And what

the cipher guarantees is that, for every single possible 128-bit combination, you're going to get a different 128-bit combination out. And there's a one-to-one relationship. That is, you can't get something out twice when, well, there are no missing patterns. That is, for all 128 possible combinations in, you're going to get a different 128 bits out for every possible combination in. And the key on the cipher, that is, the key that you give the cipher, determines which one of that many keyed mappings there are.

Okay, but if we stand back a minute, look at the number of possible mappings. If we had, say, all zeroes, and we put that in, that's going to map to one pattern of 128 bits. Then we have the - I'm sorry, I sort of started off wrong there. The idea is, if we know that we have 2 to the 128th possible patterns that we can put in, the question is, how many mappings are there? Well, the number of mappings is 2 to the 128th factorial, because for any pattern in, the output might be any one of the other possible 2 to the 128th bit patterns. Then for the next pattern in, the output might be any one of those except the first one. So we have 2 to the 128th times 2 to the 128th minus 1, times 2 to the 128th minus 2. In other words, it's 2 to the power of 128 factorial possible mappings. Which is an astronomically large number.

Leo: Yeah, no kidding.

Steve: I mean, it's just phenomenally large, how many possible mappings there are. But consider the key. The key is, for example, the maximum key that Rijndael uses, it can use either 128-bit, 192, or 256-bit key. Well, we know that that, of course, has 2 to the 256 different combinations. Well, that number is far smaller. In fact, you know, 2 to the 128 is just 2 to the 128 times 2 to the 128. So essentially it's like there are 2 to the 126 factorial mappings missing. So my point is that the 256-bit key can only access a tiny, tiny, tiny fraction of the possible mappings between the input and the output of the cipher. So the fact is, what the cipher is doing is, driven by the key, the key is able to select a minuscule number of possible mappings, meaning that there is just a vanishingly small chance that double encrypting with different keys would be equivalent to a single encryption with a third key.

Leo: I'll take your word for it because I didn't follow that at all.

Steve: It's like the chance would be 1 over 2 to the 126 factorial. I mean, it's just ridiculously small. So double encrypting works.

Leo: Right. And aren't you sorry you asked, Corby and Mat and Carlos. Joel Bialek in Syracuse, New York is looking for a new image: A long time back there was a topic you and Leo mentioned briefly I'm really interested in. You and Leo were discussing how you make images, snapshots of your systems when you first build them, then routinely make further backup images as you go. You mentioned specific software that you preferred. And for the life of me I can't remember which episode it was, and I can't find it. And don't worry, we know what it is. I've used backup systems like Maxtor's OneTouch, but I don't like having all the other software running on my system, you know, the .NET, HD Retrospect, et cetera. I'd love something that could either be parsed to DVDs or, better yet, an external USB hard drive that doesn't require other programs running in the background all the time. All that being said, what are your top three recommendations regarding this type of software? Would Windows Home Server work?

Steve: Well, the program we referred to, Joel, is still my absolute favorite. And that's called Drive Snapshot. That's PC based, so it's Windows only. Drive Snapshot is just a program which you run whenever you want to make a snapshot, so it's not running all the time. It's not going to be able to unwind changes or move back in time, any of that. You need to tell it when you

want to have it make a snapshot. What I like about it is that it does a very good job of compressing the file system. You are able to tell it if you want to limit the size of the individual image files. So, for example, you could tell it to make them no bigger than 4.7 gigabytes, in which case if your image was larger than that, it would split it into multiple files for burning to DVD. Another cool thing is you can mount those snapshots. So if you had a snapshot from another system, or an earlier snapshot and you've deleted some files, you could literally double-click the snapshot file, open it using Drive Snapshot, and browse it just like you would any folder hierarchy and file hierarchy in Windows Explorer. And then finally, you run this thing - if you did have your system die, you can run it from a DOS prompt, that is, the lowest level available OS, and it will reconstitute an entire partition from the snapshot. So I really like the program. I use it all the time when I'm configuring laptops and want to be able to step back if I make a misstep. I really recommend it.

Leo: I was just doing some math, I'm sorry, the calculator started talking.

Steve: I didn't hear it.

Leo: I know you didn't hear it. You know, that's DriveSnapshot.de, if you want to know more about that. And I recommend it, too. I use it, too. The problem with that as a backup solution is you have to image everything. And you know, a lot of times we think of backups as backing up everything. And I think that's kind of because business does it that way. And so we get it in our heads. And business does it that way because they need to restore fast. Every minute that you're down is lost revenue. So in business, where you have drones that are paid to do this kind of thing, it's okay to spend more time at the front end so you spend less time at the back end. I think it's the opposite for end users. The less time you spend backing up, the more likely you are to do it. And should disaster happen, it doesn't matter if it takes longer to get back up as long as you can get back up. So I don't recommend backing up Windows and all the applications. So I do like you do, Steve. I make an image of my first install because I don't want to go through the Windows install ever again.

Steve: Right.

Leo: Including the activation. Just restore that. But...

Steve: And the 89 security patches now for XP.

Leo: Right. The problem is that that is out of date almost immediately because there's another security patch next week. And so it's hard to keep an image file up to date. So what I do is I make that so I can do a quick install, and then I backup my data just by copying it to the hard drive. And actually I use Second Copy, which does it automatically, does a synchronize. On the Mac I use ChronoSync. These are background copiers. They do run in the background. They're very small. And basically every hour or so they say, hey, is there any change between the main drive and the backup drive? If there is, copy that stuff over, too. So you always kind of have a backup.

And that, to me, imaging is too hard. You're not going to do that. You do that once a week, so what happens if it goes down halfway through the week, you've lost stuff. Imaging is really more for that first build of the operating system. And remember, it's going to get out of date pretty quickly. So if you want to keep applications up to date, if you want to keep

the operating system up to date, every maybe six months you might want to strip it down, reinstall, add the updates, and make a new image. That's the problem with this is you have to do this pretty - you have to be pretty careful about it and do it...

Steve: I should mention, too, along the lines of a file-by-file backup, I have another program that I love. It's called FileBack PC. If you just Google FileBack PC, you will find it. And it's one I've used for years. I like it. And in fact, I've got it monitoring my Assembly language subdirectory and...

Leo: You don't want to lose one iota of that programming.

Steve: Exactly. And what I love about it, too, is you're able to tell it how many copies of each file you want to maintain; which files you want to maintain using file extension filters; and, for example, what limit on number of copies within a certain length of time. It's got very powerful backup description options. So, for example, I'm able to say I want to keep 20, up to 20 previous instances of my Assembly language files, no closer together than an hour apart, no more than 10 a day kind of thing. And it just - that way I know that my work is always going to be safe.

Leo: I have to get that. That's from MaxOutput.com, and it's 55 bucks.

Steve: It really is good, Leo. I've used it for years, and I just really like it.

Leo: Yeah, I use Second Copy, which is about 30 or 40 bucks. And it's not as flexible, though. This looks much more - this is a more serious product.

Steve: It's a serious tool. And it also understands about fileshares. It's able to link and to log onto remote drives over Windows networking with no problem.

Leo: Yeah, see, that's what I do. I map the network attached storage drive to my Windows machine and just automatically backup to that. So it's not even the same machine. But, yeah, this is where an external USB drive is so useful. But as you get more serious, network-attached storage really is the best way to back up. And then don't forget an offsite backup. Because if you, you know, look what happened to Francis Ford Coppola. They stole his computer and his backup drive. So he had nothing. So I use Carbonite to backup online. It's a sponsor of the radio show, but I use that. And now I have, like, boy, I have, like, 18 different ways I can restore. But you've got to do that.

Steve: You're safe.

Leo: That's the way you have peace of mind. James Wilcox of Rapid City, South Dakota wants to keep his router. I don't want to lose my router. I don't want to. Thanks for such a terrific podcast. Thank you, James, for listening. I have a football on my keychain that I love showing off to friends and family as I explain multifactor authentication and watch their eyes glaze over. Really great for the holidays. Anyway, I was doing a little reading on IPv6, and I had a question. This guy must be great at parties. According to MS's help file in

XP, part of the problem with IPv4 is it didn't anticipate such a large demand for IP addresses. NAT routers, of course, get around that issue. But when IPv6 comes around with its 3.48 times 10 to the 38th addresses, won't routers be obsolete? If that's true, then I guess our handy routers as firewalls will go away, won't they? Maybe that's a good plug for Astaro. First of all, where is, you know, is IPv6 imminent? And what happens if it comes out?

Steve: I don't know. I have said that it's never going to happen, and then I get a bunch of hate mail from people saying, oh, it's already happening, don't you know what's going on, you're clueless, and things like that. The problem is that the real incentive for it was largely this IP space depletion and consumption, which has not happened because of NAT routers, exactly as James Wilcox here suggests. My sense is that routers are so good for security that clearly, if we originally had 128-bit IP addresses, which is what IPv6 gives us, had we had 128-bit IP addresses, routers may have never happened in the first place. But now that they have, and they do such a good job of protecting us on the 'Net, I'll be surprised if, even when we do get 128-bit IP addresses, if that ever happens, I'd be surprised if they go away because they happened, and they're inexpensive, and it's just sort of a nice place for you to plug everything in. I mean, if you didn't have a router, then we'd go back to having a hub or a switch. Which is not as smart. But those don't cost that much less now than routers do. So I think this is something - I don't think James needs to worry about losing his router. I think he'll always be able to have a router.

Leo: Yeah, and I think many routers handle IPv6. So it's not like your router is, I mean, you'll still have a router. This router may be obsolete. My router can do IPv6, I think.

Steve: Yes. And of course that was a big deal in Vista. And I think XP has a v6 stack also. So again, it's sort of happening, but it requires that our ISPs basically swap out a lot of the hardware that they've been using.

Leo: A lot of infrastructure, yeah.

Steve: Yup, a lot of infrastructure has to be upgraded. And there's not a great deal of pressure to do it.

Leo: I think it'll happen. I think it is happening. I think it's gradual. The question is, how long before IPv6 is obsoleted? And that may take forever. Who knows how long that'll take.

Steve: That'll never happen because IPv4 is already accommodated in a little corner of IPv6. So you can keep using IPv4, and the IPv6 transport will translate back and forth.

Leo: See, that's the real question, is when are they going to phase out v4, and they're not.

Steve: Right, right.

Leo: Jim Bassett in Pleasant Hill, California also uses the Patelco Credit Union. We were talking about that on Listener Feedback 29. He said he wanted to follow up on the first

question regarding multifactor authentication at Patelco. The questioner explained that Patelco was using reverse DNS to authenticate logons. He discussed the trouble with relying too heavily on that. Jim wants to point out Patelco allows you the option to not keep a trusted provider on file, thus requiring you to receive a new password via email or SMS for each log-in. Wow. Would this be - well, all right. Would this be a more acceptable use of this type of multifactor authentication? I, too, am a Patelco member and will be implementing this as well and would like your feedback. I regularly access my account from both home and work. I for sure would not have my work ISP as a trusted provider - rightfully so, I think. And I'm thinking of doing the same for my home provider, Comcast, despite the minor inconvenience of needing to get a new password via email each time I access my account.

Steve: It's interesting. So essentially what's happening is we apparently had a system where a username and password could once be used from anywhere. Then they added this notion of wanting to, well, I guess actually it's not even a notion, it's a government regulation is coming downstream saying you must have multifactor authentication. So they're saying, okay, we're going to use reverse DNS for multifactor authentication. We recognize, though, that it's not a safe practice, that is, it's not as strong another factor as necessary. But what Jim is saying is that, in response to that, you can tell Patelco to never trust anyone, and that will force them to send you a password via email or SMS every single time, which, yes, I think is much more secure. So that's what I would do, as you said, Leo. It's certainly a bit of a hassle. But unless you're logging onto your credit union site all the time, I would think it's probably worth the security.

Leo: Via SMS isn't too bad of a problem. And, you know, it just comes to - I've noticed when I use this, Bank of America will do this as its secondary form of authentication. It's instant. So you press the button, bzzz, there's your phone, there's the number. I think that's probably a good way to do it. In fact, I would like to do it that way.

Aaron Mashburn, writing from the Republic of Panama, has an SSL question: The recent PPP episodes, Perfect Paper Passwords, sparked my interest in crypto. So I went back and redownloaded episodes 31 through 37 as a refresher course. Good idea, by the way. That's a good thing to keep in mind. We've covered these subjects, you know, kind of the primers on these subjects in past episodes. So if you go back you can look at those. He says as he listened to Episode 37: A question popped into my mind. I understand an SSL certificate verifies the identity of my server. But is the cert used at all in the encryption process for the SSL?

Steve: Well, it was a neat question. And in fact we've had a number of questions like this such that I've made a note in our upcoming show plan to specifically talk about SSL. We've talked about the need for it. We've talked about the certificate side. But we've never actually looked at the protocol itself. And as is so often the case, it's extremely understandable. So I want to give it an entire episode in the future.

In the meantime I wanted to respond to Aaron's question and say that, yes, there are aspects of the certificate which are used for the HTTPS connection. Specifically, what the remote server does is it provides the client with its certificate which has been signed by the certificate authority, which as we know allows the client to verify the signature. It also provides - it's the public key of its public/private key pair. It provides the public key so the client can then choose a random number, uses a pseudorandom number to obtain a symmetric key which they will use for bulk encryption during their conversation. You cannot use the public key for bulk encryption because it's much too time-consuming to do that. So you use the public key just to encrypt the symmetric key.

So the remote server provides the client with its public key. The client encrypts the randomly chosen symmetric key using the server's public key and sends it back. And the beauty of public key crypto, which is what Aaron relearned in episodes 31 through 37 of Security Now!, is that someone could see that, could sniff the wire and see that going by. They could literally see the publicly encrypted symmetric key chosen by the client and be absolutely unable to decrypt it. Only the server that has the matching secret key in this public and private key pair is able to decrypt that which was encrypted with its public key in order to obtain the symmetric key which they then use for communicating. So that's how that works. And we're going to do an episode that really - that covers this because there's a lot more features and options in it. And we're just depending upon SSL for so much these days.

Leo: Oh, yeah. Oh, yeah, all the time. Jack's a little worried about being spied on in Australia. He says: Is it possible to strip the security of HTTPS - that SSL we were just talking about - and decode the encryption so that it could be read by the Internet service provider or other government agencies in Australia?

Steve: And the answer is a resounding no. You have to be careful that, for example, your ISP has not given your browser a certificate authority and is able to proxy your HTTPS connections. But...

Leo: But you would know that because the certificate would say your ISP's name and not the site's name.

Steve: Yes. It would show that the ISP had signed that site certificate, which it would be creating on the fly, rather than, for example, VeriSign having signed it. So you absolutely, if you were at a site and you verified the chain of signing authority for the certificate, and you see that it goes back to one of the very common root certificate providers, then there is no way, given that, that anybody sniffing the wire is able to, as Jack said, strip off the security. I mean, that's the whole point of SSL. And it is a well-known public standard with no ways around it.

Leo: Clarify for me, though. Okay, so I'm at work, and I know that my Internet is going through a server at work, and that they're probably doing this certificate shuffle here. So I'm on Amazon.com. I've got a certificate. If I get info or look at the security on the page, will it still say, the certificate, will it say Amazon.com, or will it say my office?

Steve: It would say Amazon.com.

Leo: It would still say Amazon.com.

Steve: And then it would show that it was signed by your company. And so...

Leo: So it's the signing is what you want to look at.

Steve: Yes. You want to see who it is that signed the certificate. And...

Leo: Now, you're screwed if you work for VeriSign.

Steve: That's a very good point. I never thought about that.

Leo: [Indiscernible] VeriSign, and you can't tell if it's the right, you know, if it's...

Steve: Yes, that's a very good point. They could certainly be - and we assume that they're not. We don't know they're not, but I've never thought about that before. But you're right. Since they're the signing authority, they could be signing certificates that they're making on the fly, and everyone's browser would accept them. That's a very interesting hack, Leo.

Leo: Now, it's different in each browser to get that information. But in most browsers you can right-click on the page and say, what, View Certificate or...

Steve: You'd look at the page properties, and then View Certificate. And then you'll see the first certificate. And then you could look at details. And it'll sort of show you normally a hierarchy of signatures going back to some root. And so it's the root that has signed all the certificates in the chain of trust. And if that goes back to someone like VeriSign or eTrust or one of these - or Thawte, for example, then those are mainstream certificate-signing authorities, and you know that the site you're going to had its certificate signed by them.

Leo: Got it. Got it. Zacc Cooley in Gilbert, Arizona wants the world to stop spinning: I was wondering how the new solid state hard drives - we just got one at the Lab, by the way, 64 gigs, 2,500 bucks.

Steve: Yeah.

Leo: I was wondering how the new solid state hard drives are going to change how we as consumers back up our data. Do the bits in a solid state hard drive function like the bits on the spinning platter hard drive in that they can become corrupted? Hell, yeah. Will I still need SpinRite to use on my solid state drive over time? Will we need to defrag them? Will we need to back up the data on solid state hard drives as much as we do now? I've got questions. Is there a battery inside like motherboard batteries that keeps the data in there than could die and cause data loss? What, Steve, tell me, what?

Steve: Well, there's essentially no relationship between solid state hard drives and physical hard drives.

Leo: Well, they store bits.

Steve: Well, okay, exactly. But no, in terms of the technology, they are absolutely different. So, for example, there's no need to defrag them because...

Leo: Oh, interesting, really.

Steve: Yes. There is no physical seeking going on.

Leo: Oh, so it doesn't matter. It's a read cycle from any address is the same.

Steve: Exactly.

Leo: Oh, interesting.

Steve: There's no battery inside. What they use is an electrostatic technology where essentially it's possible to strand electrons on a little tiny bit of metal that's insulated. And the electrostatic influence of the electrons is sensed on the other side of the insulator, and that's what determines if it's a one or a zero. So basically sort of you squirt the electrons, you force them through the insulator, and they get stranded there. Or you suck them out, and then they're no longer there. So it's a very different technology, completely different from a hard drive.

The problem is that that process of squirting electrons through the insulator, known as tunneling, it actually creates some physical damage and some wear over time. So there is a problem with this technology in terms of the number of times you can write to it. You can read to it easily just by saying - just by querying whether or not this little transistor is on or off. But the process of writing is just a little tiny bit destructive to the system. The reason writing is so slow is that it actually requires a higher voltage in order to inject the electrons. And that requires something called a charge pump to charge itself up in order to cause this injection to occur. And that's why writing to these non-volatile RAMs is much slower than reading from them.

But the other thing that happens is, if you write to them over and over and over, they die. So they don't die fast. It's like on the order of 10 to the 5 write cycles, so like 100,000 write cycles. But not infinite. Hard drives are infinite. That is, it doesn't hurt them in any way to change the data on them. It actually hurts non-volatile memory to change its data. So in order to mitigate the damage, non-volatile RAM has a technology that spreads the actual writing around the surface of the RAM. So that even if you are reading and writing the same area, that is, the same address of the RAM over and over and over, it's actually occurring in a distributed fashion across different physical areas of the RAM. They do that in order to spread out the damage caused by writing to it. So you really don't want to defrag because that's needlessly writing to a technology which has a limited number of write cycles. You certainly don't want to run SpinRite on it because SpinRite just writes like crazy. I mean, that's good for hard drives. It's bad for non-volatile, solid state hard drives.

Leo: So don't run SpinRite on it.

Steve: You don't want to run SpinRite.

Leo: Would it work? I mean, it's an IDE interface, isn't it?

Steve: It does work. Mark Thompson, my buddy at AnalogX, years ago he was curious about whether non-volatile RAM really was hurt by writing. So he used a PCMCIA EEPROM, a non-volatile memory. And he set it up as a Windows swap drive.

Leo: Oh, boy.

Steve: Windows killed it in one hour.

Leo: Whoa.

Steve: Just killed it dead.

Leo: There you go.

Steve: It just was game over.

Leo: He's got money to burn, that kid.

Steve: He's crazy.

Leo: So but that's interesting because ReadyBoost, this ReadyBoost and ReadyDrive technology that Microsoft Vista supports is solid state. But it's not writing to it a lot. It writes to it once, then it uses it as a cache.

Steve: Exactly. It's very much - reading is no problem. And also you'll notice that other mature technologies, they will use RAM; and only very seldomly do they flush that out to the non-volatile. So mature technologies understand that you cannot write to these things all the time. You must - you can read from them easily, but writing is something you want to tend not to do. And as you said, Leo, it's still very expensive.

Leo: Oh, yeah. It's not - we're way off from this being common.

Steve: Yes.

Leo: I was surprised how expensive that was. All right, Steve. Are you ready? It's time for the Perfect PPP Quandary of the Week. Sean Reiser of Astoria, New York says: First off, this is not a story about a flaw in the PPP system, just a reminder of the saying "because there is no patch for human stupidity." PPP in and of itself is excellent. We just need to eliminate humans from the equation. PEBKAC, baby. I've been piloting an implementation of Perfect Paper Passwords on a corporate site to replace RSA key fobs - oh, really, neat - and encountered a problem I didn't expect. Once we expanded the pilot beyond the techs into some of the business units, I noticed users tend to take advantage of the blank backside of the paper to note little things like the site URL, their username, their password. Oh, boy. No matter how much education we did, even if the users were specifically told not to write this information on the cards, there were always users who did it. Barring lobotomies or summary execution, I really don't know how to handle this. I'd appreciate any ideas you have. That's, I mean, but what are you going to do about that?

Steve: Yeah. The only thing I could think, I mean, I was thinking, well, that's interesting. So essentially by writing their username and password on the back of their PPP card...

Leo: It's right there.

Steve: ...they've completely scrapped all the security that is available if they lose physical access to their card.

Leo: Especially if they're crossing off the PPPs as they use them. You even know what the next PPP is.

Steve: Yup. And so the idea of something that they know and something that they have, well, if a bad guy gets a hold of it, then he knows what they know, and you've lost all of your multifactor strength. The only thing I could think was to put - first I was thinking, okay, well, how about if you print it double-sided, and you just put like a cross-hatch on the backside? People might still write over that. So then I was thinking, okay, there's nothing that would prevent you from making double-sided PPP cards. That is...

Leo: Oh, good idea.

Steve: ...instead of just printing one web page that contains three, print two, so that you have cards 1, 2, 3, and 4, 5, 6 on the second page, which will print out, when you print it double-sided, on the back side. So now the card...

Leo: People would still write it, though, in the little bit of space there.

Steve: I know.

Leo: You've got to laminate them. Laminate them. Then they couldn't - they'd have to use a grease pencil.

Steve: That's probably a good idea, Leo.

Leo: Just laminate them.

Steve: So it just will not take ink.

Leo: Right.

Steve: There you go.

Leo: I don't know what - take the pens away from them. I don't know what you can do. There must be, you know, it's funny, I have great sympathy for IT professionals. And I know how much they hate users. And it's hard.

Steve: The real world is a real problem.

Leo: Yeah. And you go in, and I'm sure every IT professional goes in, you know, gung ho, excited about their users, they want to help them, and they slowly grind you down. You start out with all the goodwill in the world, but slowly they grind you down. And I don't know a single IT pro who doesn't have a horror story or two or three or four. Are you ready for the Great Tip of the Week?

Steve: Yeah.

Leo: This is from Patrick on Montreal: Perhaps you've already had a slew of people tell you this, but just in case no one did...

Steve: And nobody did, by the way, Patrick.

Leo: You're the one. An easy way of accessing the Flash Player Settings Manager is to right-click a Flash object on a web page, then choose Settings, which brings up the basic settings panel on which there is an Advanced button right on the first tab there. It'll take you directly to the right Adobe/Macromedia page which contains the Flash Player Settings Manager. Also might be worth mentioning that settings should be adjusted by accessing the panel while browsing in Firefox as well as IE because they are browser specific, I guess.

Steve: Yes, exactly. There's a different plug-in for the Firefox from Adobe than there is for the IE. And I didn't verify this, but I'm assuming Patrick is saying that they're not sharing settings on a single machine. So you need to go there both. But I really liked his suggestion because it's, I mean, many people wrote to say, hey, my bank is using Flash cookies. I've never heard of Flash cookies before, but that's what they're doing. So our mentioning this and talking about this to our listeners a couple weeks ago brought this to the fore. And of course people want to have control over what their machine is doing and the trackability. They may have been people who were disabling third-party cookies, but never even thought or knew about Flash cookies.

So I liked Patrick's note because it makes it very easy. In order to check this out, I just went to MSNBC.com, and I went to CNN.com. Both of them, I mean, just assuming they would have Flash stuff. And sure enough, Flash began jumping around on their page, doing animation. And I just right-clicked, hit Settings, and then clicked the Advanced button. And that took me to the Adobe page, just exactly as Patrick said. So I loved his idea.

Leo: Excellent. Very, very. All right. It is time for our last question of the day from Brian W., also in Montreal. He asks the Horrifying Show-Stopper Question of the Week: I've been a listener since Episode 1, and I love the show. Thank you, Brian. I know you guys have covered keystroke loggers in the past. I am one of the probably millions that love using wireless keyboards, but I never considered the security risk. Yeah. I saw a Black Hat presentation on wireless keyloggers. What really worried me was that the encryption, if you could call it that, on Microsoft's wireless keyboards was a 1-bit shift register. Is this, like, industry wide?

Steve: Well, I did some research after Brian...

Leo: Somebody asked me this on the radio show, you know? And I didn't know. And I'm glad Brian asked this question.

Steve: Yup. Get a load of this. It's not a 1-bit shift register. It's a 1-byte static byte that is XORed with the data from the keyboard.

Leo: So would that be pretty easy to reverse engineer?

Steve: Leo, it'd be hard not to reverse engineer. It is horrifying. It's horrifying.

Leo: And this is true not just for Microsoft, but do other keyboards do it this way?

Steve: Well, apparently Logitech has recognized that this is a problem that's sooner or later going to get exposed. Microsoft's wireless keyboards do this. The 1000 series and the 2000 series have been examined. The 3000 and the 4000 have not been. But it appears to be the same for them. Logitech has, like, a secure connect...

Leo: They have an encrypted keyboard, yeah.

Steve: Yeah. And so they're boasting about that. But the extremely popular Microsoft keyboards, during the so-called "association phase," the keyboard chooses a random byte, one byte of randomness, and provides it to the reader. Then the keystrokes you type are XORed with that one byte. Which means, as we know, there are 256 possible combinations of one byte, that the one byte can have. All you have to do is suck in a bunch of characters, you know, wait a few minutes for someone to type 20 or 30, and then in a heartbeat you could check every possible byte. One of them will turn what they're typing into English or clear text or whatever language they're typing in. In that case, at that point, their keyboard is decrypted for all intents and purposes, deciphered. What this means, of course, is that in a situation where people are within sniffing distance, radio distance of a keyboard, you absolutely have to consider that it is not safe. Keyboards are using a low frequency, 27MHz, which is extremely easy to receive, meaning that in an apartment building, neighbors who have a wireless keyboard could have everything they're typing trivially decrypted, if it's at least on these Microsoft Series 1000 and 2000 keyboards, and probably other keyboards. So it's definitely a concern.

Leo: Good to know. And so the Logitech ones that stay encrypted, do you know what technique they use?

Steve: I don't, and it's something definitely worth some research. We know that it's actually not very easy to perform really good encryption against man-in-the-middle attacks. It's absolutely possible, and several times we've talked about the technologies to enable that. But you've got to have some serious work being done on each end in order for a conversation in the clear that can be monitored, as any radio conversation can be, to have that kind of a conversation secure. It's definitely possible to do it. But you've got to really want to. So I'm wondering if Logitech's approach is way secure, or just less hard to crack. But again, it'd be - I don't know if you could make anything less hard to crack than what Microsoft has done, which is just choose a single byte and XOR your data with it. It's like, why even bother?

Leo: Well, in their defense, they don't say it's an encrypted secure keyboard, do they?

Steve: No, they don't.

Leo: And so it's probably more to prevent crosstalk from other keyboards in the same area.

Steve: It doesn't, no, it doesn't do that. And it turns out that there is now technology on the 'Net that will simultaneously record and decrypt from all the keyboards within range at the same time. Because the keyboards do have a unique identifier that allows you to disambiguate the data coming in from all the different keyboards. I mean, it's unbelievable.

Leo: That's crazy. They can log everybody in the whole office without even touching their computers.

Steve: Yes, just put on your tinfoil cap and connect a wire to it and send it in to your keyboard receiver, and you can monitor what everybody's typing.

Leo: Unbelievable. Well, we've come to the end of another fantastic and fascinating edition of your questions, Steve's answers. We thank you, Steve. It's great stuff. I want to remind everybody to go to Steve's site, GRC.com. That's where you'll find 16KB versions of every single episode ever of Security Now!, all 122 of them. You also have transcripts, which is great. Do you have transcripts for every episode?

Steve: Every single one. I had Elaine go back, and she started from #1 and came forward.

Leo: Oh, that's great. And lots of great free stuff, including ShieldsUP; PPP, the Perfect Paper Passwords, some sample implementations and so forth; his great security forums; lots of simple and useful utilities like Wizmo; and of course everybody's favorite hard drive maintenance and recovery utility, but don't use it on your Flash drives, SpinRite. You can use it on your spinning - well, it's SpinRite. You don't have FlashRite.

Steve: No, not doing FlashRite yet.

Leo: SpinRite. GRC.com. Steve, so much fun. Thank you so much. I appreciate your time.

Steve: And I yours, Leo.

Leo: Are you going to take Christmas off? What are we going to do here? We going to keep going?

Steve: We're going to keep going. We're not going to miss a single week.

Leo: You just want to catch up with TWiT.

Steve: Even though - yeah, I'm going to pass them up this time finally. Even though you're going to be off in Egypt for a couple weeks; right?

Leo: I will. Christmas Eve I'm leaving, which is just five days from now. I'm going to leave to go to Providence, spend Christmas with my Mom and sister and my family. And then all of us, my family, the four of us are going to go to Egypt on the 27th. We'll be in Cairo. If we have any Security Now! listeners in Cairo, say hello. I'll be the one on the donkey. Or the camel, I'm not sure. And then we come back January 6th. So we'll get back, the network itself will get back in the swing of things kind of slowly after the 6th. I probably will miss at least two TWiTs or three TWiTs, which means you'll be ahead by then. Finally. And then we're doing a MacWorld, which is January 15th through 18th, we're doing a live podcast every day there in the West Conference Hall. If you come up, we'll do a Security Now! up there, if you come up.

Steve: Oh, I'll think about that.

Leo: I can get you in.

Steve: Cool.

Leo: I have friends.

Steve: I like that.

Leo: Yeah, it'll be fun. And then I do want to remind everybody that we do have that holiday coupon still for ScotteVest merchandise. Anything on the ScotteVest site, as you check out, use the coupon code LEO, and you'll get 20 percent off the top, right there. It does not include the TWiT merchandise available through ThinkGeek, just the ScotteVest merchandise at ScotteVest.com. 20 percent off, use the coupon code LEO. And that's just our little way of saying thank you for being such great listeners all year long.

And I would be remiss if I didn't thank the folks who donate. Because I think people hear the ads, on this show especially we've had full sponsorship. We're sold out. We'll never have more than two ads. And we have been sold out all year. And I think probably some people think, oh, well, they don't need our money. We do, because sponsorship frankly does not cover the costs. The money from sponsorships goes to the hosts. The costs, the day-to-day costs are coming from your donations, things like the server, the rent, Dane's salary comes from your donations because those are consistent, and advertising is not.

So please, we do appreciate your donations. Don't stop. Keep them up. We thank you so much for your support. Just go to TWiT.tv and click those links. Even $2 a month is plenty. That's enough just to show you care. All right, Steve. Have a great week. We'll talk again, boy, we're going to be getting close to Christmas when we talk next. It'll be the 20th, 12/20.

Steve: Thanks, Leo.

Leo: All right. Take care.