SECURITY NOW!

Transcript of Episode #186

Listener Feedback #61

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-186.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-186-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 186 for March 5, 2009: Listener Feedback #61. This show is brought to you by listeners like you and your contributions. We couldn't do it without you. Thanks so much.

It's time for Security Now! with Steve Gibson, the show where we cover all things secure - privacy, online theft, that kind of thing. Steve is a real expert in the subject. Partly by force. I think some of your interest in security began when you were being attacked.

Steve Gibson: It started off defensively. So, yeah.

Leo: And of course you found spyware on your system, were the first to coin the phrase "spyware" and write the first antispyware program. You handed that off to pros. And I'm sure you're glad. That's a never-ending job.

Steve: Oh, boy, yes. It was becoming very clear quickly that this would, I mean, it would be a full-time job. And much as I think spyware is important, it's just sort of not my nature to go running around chasing after these weenies and try to figure out what's going on. I much prefer sort of the raw technology, let's do something new, than that. So the guys at Lavasoft sort of picked up the mantle and have done a good job with Ad-

Aware.

Leo: Ad-Aware, Spybot, and now of course everybody. You know, it's funny, I think for a long time antivirus companies were reluctant to say we will also fight spyware because they didn't - they thought, well, maybe this is commercial speech. It's not exactly a virus. I think there's now no question.

Steve: Well, yeah. Remember, too, that early on there was, like, Conducent and Aureate were two commercial companies that were installing this junk in people's machines that really infuriated people, yet they were not rogue malicious hackers. They were companies that someone had mistakenly given some venture capital to. So, yeah.

Leo: Right, right. Let's see here. We have...

Steve: We've got #186 for Q&A #61.

Leo: Questions. You've got questions; Steve's got answers. And some of them are about our great special that we did. I hope everybody enjoyed Gray-Haired Computing.

Steve: Oh, I got a lot of feedback about that. I included just a couple little things. We have a really interesting note from - that I think people are going to find fascinating from the Chief Technology Officer of PKWARE.

Leo: Oh, you're kidding.

Steve: ...who heard me talking about - who listens to Security Now! and heard me talking about ZIP encryption. So we get the whole story on that.

Leo: PKWARE. There's a legendary name in computing. Wow.

Steve: Well, and it was Phil Katz of PKWARE.

Leo: Phil Katz.

Steve: Phil Katz is PK. That's where that comes from. And he invented the ZIP file format. That was his.

Leo: Oh, that's fun. Steve Gibson, any news in the security world?

Steve: Unfortunately, we do have news.

Leo: I was afraid of that.

Steve: Yeah. We haven't had a bad Excel exploit for a while. So I guess maybe one was due. We have a new zero-day Excel remote code execution exploit which has been found in the wild. Anyone who has Office installed who opens one of these malicious email attachments that contains an Excel spreadsheet, even though it might be labeled something differently, I mean, so you get a link in the mail, in your email, it could have a name that makes it look like it's not an Excel spreadsheet because someone might be thinking, wait a minute, why is Aunt Mary sending me an Excel spreadsheet? But it ends up triggering Excel and makes a remote code exploit possible.

Leo: This is cross-platform, too, by the way, this...

Steve: Yes, I was going to say it's Office 2000, 2002, 2003, and 2007 on Windows. And over on the Mac side it's both Office 2004 and 2008. So this is clearly code which has been in Excel for quite a long time, and somebody finally found a way of glitching it and leveraging that to be able to get code to run on your machine. So right now Microsoft has acknowledged it. They've got their article, it's 968272, which says they know about this. And I'm sure that, I mean, I imagine that the second Tuesday of next month we will see a patch for this because they're saying it's only being used in targeted instances, which is to say, you know, when someone wants to get a certain executive, they'll send them email hoping to be able to take over their machine because that's the way these things always start. And before you know it they'll be spamming people en masse, trying to get this to happen.

And then Adobe has a couple problems. We mentioned a known problem that was not going to be patched until March 11 for version 9 of Acrobat, and a week later, March 18, for the earlier versions. Exploits are now in the wild, so this is now being actively exploited, as we expected it would be. So I just wanted to give people a heads-up that about a week from now, on March 11, you'll be able to get an update for Acrobat. And if you're an avid Acrobat user that's something you want to consider, certainly want to consider doing. And we'd also talked about last week the idea of putting it inside of Sandboxie, causing Acrobat to run in a sandbox. And if you open PDFs from your browser, and you've sandboxed the browser, then Acrobat is sandboxed by being part of the browser's sandbox. So I just wanted to caution people this is out there in the wild, and it's going to take a few weeks for Adobe to catch up.

Meanwhile there are some new problems with Flash. Adobe's Flash Player has multiple known problems - denial of service, information disclosure, clickjacking attacks, and remote code execution, a whole bunch of things. For this, however, there is an update. And my system notified me a couple days ago that there was a new version of Flash available. So anything that is over on Windows and Mac is 10.0.12.36 and earlier are troublesome. And over on Linux 10.0.15.3 and earlier are troublesome. And so what you need is just go to Adobe's site, or see if you can update your Flash Player, and also your player that's embedded in your browser, if you've got a Flash embedded there. You want 10.0.22.87. Or if you're still back on version 9 of Flash, you want 9.0.159.0 in order to get the Flash Player that's been patched to fix these multiple problems.

And the last little blurb I have is just to note that Facebook and MySpace are becoming a larger battleground for various types of exploits. They have, naturally, sophisticated APIs that are allowing these widgets to be created. And as unfortunately happens when we add complexity in technology, we inadvertently create opportunities and doorways for various kinds of exploits. So there are now - there's a worm that has sort of resurfaced called the Koobface worm, K-o-o-b-f-a-c-e, which is now back and active over in Facebook. It takes advantage of the whole social networking mode to send a note to people you know saying that you want them to see something, so click this link to go to on YouTube to see this funny video you've found. Well, that link, which you don't send, this worm that you've unfortunately acquired by doing this yourself, it sends them to a fake YouTube site that takes advantage of known problems in client systems to install malware and essentially to put a backdoor trojan on all the machines it infects. And it's spreading widely.

So it's the same story that we've run across before, is we get a new environment. Obviously Facebook and MySpace have been attracting a lot of people who are less computer and security savvy than they might otherwise be. They get this kind of a note from someone they know, and they go, oh, look, I want to see the funny video. And certainly it's not the first time they've received something like that. So it's another problem just to be aware of. And if you know people who are less security aware, you might mention that this kind of thing is happening on Facebook and MySpace and to be very circumspect, even if and when these things come from people they know. If it's in any way different from what they expect, they should be careful about it.

Leo: I've had this happen many times, or something similar happen many times, I think it was Koobface, on my Facebook. And what happens is sometimes people's sites, Facebooks get hacked. And so you get a message, as you said, you get a message from somebody you know, and it just has a link in it. And it looks completely...

Steve: Yup, generic.

Leo: ...generic and sensible. It doesn't look like something odd, necessarily. And so you're very tempted, because you feel like you're in a safe space, to click that link.

Steve: Yup.

Leo: Do you get infected the minute you click it?

Steve: Yes, well, you click it, and you go to a fake site, a fake YouTube site.

Leo: Oh, then they ask for the login. Or no. Oh, I know what it is. You download - you have to download a new player, an upgrade...

Steve: That's exactly what it is. Yes, it says, oh, you need to update your player in order to play this video. That's exactly what it is, yes.

Leo: Yeah, yeah, yeah, yeah. And then of course - see, this is a problem.

Steve: It's not a new player you're downloading.

Leo: Yeah, it's a bad guy.

Steve: It's definitely bad stuff.

Leo: Yeah, that's nasty. I've seen that's been going on for a while. It's happened to me several times. And what I'll do is I'll immediately send a note to that person saying your site has been hacked. You need to change your password. Somebody's putting out messages on your...

Steve: On your account.

Leo: ...under your name, and you should also tell everybody don't click that link because it's a bad link.

Steve: Right.

Leo: Well, we've put the word out.

Steve: Okay. A couple last little bits of errata. I did want to mention, I was listening to you talking about the issue that I have with the Kindle 2, and that is the non-replaceability of its battery. And I did want to mention that both of my prior Kindle 1s, during the year and a half I had them, had noticeably reduced battery life. And that unfortunately - and this goes, I mean, this is pertinent not to just Kindle users, but laptop users and iPod users and all kinds of users. Lithium ion technology not only has a cycle-life limitation, but just a shelf-life limitation of a few years. That is, even if you were an infrequent user of your Kindle, you would find that the battery itself ages, even when it's not being continually cycled. And after a few years, you'll see a reduced battery life.

So I'm really not happy that the Kindle 2 has switched over to this non-replaceable battery. Although you've got to wonder what the battery looks like in there because the Kindle 2 is so thin, it's clearly not something you're able to get off the shelf. So, but it's unfortunate that they chose for whatever reason not to do what they did before, which was allow it to be opened somehow, and for the user to be able to replace the battery. I was going to let mine get really bad and then, you know, like, just to see how bad it got. But it was both of them, because remember I dropped mine on the edge from about two feet, I mean, a little gentle drop on the carpet, and I was surprised that it completely whacked out the display. Amazon was wonderful about replacing it for me. But I remember even then that that first one's battery life was clearly shortened by just my regular usage at that point, after less than a year. And the second one was following the same path. So I'm a little worried. And I haven't yet brought my battery all the way down. But it's looking to me like this battery life is not as much longer as they touted for the Kindle 2. Have you...

Leo: Not at all. I was very disappointed with it. It's, you know, they said 25 percent greater? No.

Steve: No, I think it's...

Leo: I've depleted it in a night or two.

Steve: Yeah. And is that without using any text-to-speech?

Leo: Without text-to-speech, without any audio playback. I left the radio on, but you're kind of tempted to do so because it's a software switch, not a hardware switch.

Steve: Right, so it's a little less obvious when it's on. I have also discovered that it was on when I am not using it.

Leo: Well, and I leave it on because I subscribe to daily stuff. So...

Steve: Yeah, I mean, and that's another thing, too. I was hoping that when they put it under software control, they would give you an option of, like, have it...

Leo: Schedule.

Steve: Yes, exactly.

Leo: Good idea.

Steve: Have it wake up at - it know what time it is, presumably. Have it wake up and check for anything new and then immediately go back to sleep. That way, you know - and if it only did it - it doesn't even have to know what time it is. If it did it hourly, that would be fine. Or you choose how often.

Leo: Like a day, yeah. You know what, that's brilliant. And I bet you they'll do that in firmware because they haven't improved the battery life. It's a little disappointing.

Steve: No, I think not. And in fact my guess is that one of the reasons the page turn is faster, and image display - remember how long the first Kindle took, if you were switching to a page that had, like that was much more complex or had an image, it would kind of [groaning], I mean, it really struggled. It doesn't do that now. So my guess is they've juiced up the processor in order to get the next page up on the screen at the cost of some battery life.

And then someone somewhere posted some - I don't remember now where it was. But I made a note of it here to just mention that it's unfortunate that the Kindle didn't incorporate WiFi because had it, it could be international, if they wanted it to be. I mean, they argue...

Leo: Oh, good point.

Steve: It's only - it's the EVDO Sprint connection that is sort of locking it to domestic. Yet there's obviously a huge international interest in this. And had they given it WiFi, then although you wouldn't have the really cool cellular connectivity, you'd have Internet connectivity which would have allowed them to do the whole content management.

Leo: Oh, that's a very good point, yeah.

Steve: And then finally, several weeks ago I fumbled around, trying to come up with an explanation for hysteresis. And it's been bugging me ever since that I...

Leo: [Laughing] It all started because I read a definition out of an antique radio/television dictionary that Dane got for me.

Steve: And I thought of a perfect clear example. I was discussing keyboards with a friend of mine, and I realized that the snap of a key switch, when you press a button and it snaps, that's a perfect example of hysteresis.

Leo: Oh, really.

Steve: Because it's a nonlinear response. If you just had a spring where, as you increased the force on the spring, it goes down in proportion to the force, and then smoothly. And then as you bring the force back off, again you get displacement that is a direct proportion to the force, which is typical spring action. That's zero hysteresis. But if instead, like any keyboard that's got - not the mushy keyboards that have no hysteresis. But anything with a snap in it, when you reach a certain point suddenly something happens, that snap action is a - you've crossed a threshold. Suddenly the force required drops, and the key moves in. Now, even if you come back off to the same point where that snap occurred, the key stays down until you go much further back, where you've got the same sort of nonlinear event on the way back. And it's the separation of those two, the snap where it occurs down stroke, and the snap where it occurs up stroke. They're at different locations, and that gives you - if you were to plot force versus position, you would end up with not a straight line, not even a curved line. You actually end up with a loop because the force and position curve is different on the way back than it is on the way down. And that gives you a loop. And that's the characteristic hysteresis curve.

Leo: That's an excellent and accessible definition. Much better than the one in the book.

Steve: Or whatever I fumbled with. I don't remember what it was a couple weeks ago. It's just, oh, I thought afterwards, oh, that really clarified everything for everybody.

Leo: It's a hard thing to come up with an accessible definition of some of the abstract concepts that we deal with in computer science.

Steve: Well, and my problem is verbal. I mean, I could absolutely draw a diagram and wave my hands around. But the bulk of our listeners are...

Leo: Listening.

Steve: ...listeners. And so...

Leo: And that's my skill. I think if I have one skill it's that I can sometimes, not always, come up with a good analogy that makes this stuff make sense. But not always. I remember the time I was using sliced cheese to describe how audio is sampled. That didn't work so well.

Steve: Okay.

Leo: Not a recommendation [laughing]. So you try.

Steve: And on that note I will share a very short SpinRite comment.

Leo: Okay.

Steve: From Germany, a listener in Germany, Arne Klawitter. He just said, "Thank you/SpinRite." He said, "Just wanted to say THANKS" - all caps - "for recovering 32 gigs of data when nothing else helped. I tried so many other programs unsuccessfully. SpinRite really convinced me. And additionally, it is so easy and intuitive to use." That's because he's German, sort of has that strange interface.

Leo: It makes sense to me.

Steve: "So intuitive to use, great product." And then he signed it. So I just wanted to thank him for sharing his short, positive experience.

Leo: It makes sense to me, and I'm not German. It makes perfect - it's very clean and very simple. It's just the way, you know, it's the last DOS program I still use, I think. Now, ladies and gentlemen, without further fanfare or adieu - ado, not adieu, ado. Without doing no more, I'm going to read you our first question. Ladies and gentlemen, this comes to you from Listener Mike, with a cool retro-PC tip. He says: Hi, Steve. As you may have gone into retro mode in regard to old PC hardware - I guess this is another guy who heard our last episode, the Gray-Haired Computing episode - I just wanted to give you a tip in regard to the Retr0bright Project. Do you know about the Retr0bright Project? I haven't heard of this.

Steve: I do now.

Leo: It's Retr0bright with a zero, not an "o," Retr0bright.wikispaces.com. Or you can just Google Retr0bright with a zero. They offer a cool, as in cheap or free, solution to the annoying yellow color old PC equipment tends to get. Nice. Yeah, every one of my old beige boxes is now yellowed.

Steve: Yeah. And in fact it's distressing when you see what used to be a nice, like even a gray color will turn really this sort of gross dark orange-yellow color. What happened was, first of all I wanted to acknowledge that this mention by Mike is the first one I encountered. But as I was scanning through our mailbag I ran across, like, four or five other people. So I wanted to acknowledge them, too. Many people who knew that I was digging around back in nostalgia and old machines sent this news of this Retr0bright. And on the site they show some before and after shots. And it is phenomenal what this does. What happened was some chemical engineers got together, thanks to the Internet, and actually figured out what it was about the plastic that was causing it to yellow. Turns out it's the fire retardant chemistry which is mixed in, which is one of the reasons why monitors are among the worst yellowers of all.

Leo: In a few years they turn yellow.

Steve: Yeah. And so they've come up with a formula which, interestingly enough, UV is one of the things that causes the monitors to yellow. You may have noticed, like if your monitor is in a window, one side of it is, like, much yellower, the window-facing side, than the other. It's because of the UV that ends up - although I think glass really attenuates UV. But still there does seem to be some photo-based effect. Anyway, they use this formula with UV. One example was it was done in Arizona, and someone in the UK did it with a UV light in order to get enough UV. But it really returns the plastic to its original look, like a light grey where it had become dark yellow. They show a Commodore keyboard sort of before and after. And they've got both a low viscosity liquid and a gel to make it easier to apply this stuff. So I just wanted to share it with our listeners and to thank the other people who also mentioned it because I thought that was very cool. It does bring that - it takes that antique look off, which some cases you really want off because keyboards and computers and monitors really get kind of gross-looking over time.

Leo: Oh, yeah. And if you're going to collect this old stuff, you're really going to want to make the difference. That's really, really neat. Retr0bright. Question 2, Chad Young in Ann Arbor, Michigan. He mentions an Apple solution for WiFi guests: Apple recently added a guest networking feature - boy, that's smart - to their AirPort Extreme and Time Capsule devices. This appears to be based on their dual band technology wherein they utilize two individual radios in each device. That's MIMO. I think all MIMOs do that.

Steve: I was just going to say, that's entirely separate from the guest networking; right.

Leo: Oh, okay. I remember this being discussed in a past episode; I thought I'd pass it along. So what are - do you know what they're doing?

Steve: Yes. What Apple has done, they've really done it the right way. I wanted to bring this up because I don't think we've ever really talked about this notion of guest networking, although some router manufacturers over on the - like D-Link and Linksys and so forth have begun to offer such a feature; however, not always in the right way. Apple has done it exactly right. I don't want to say "as you'd expect." But I do think that Apple often does things right. And that is, they have, I mean, if you and I and our listeners were to design this, it's what we would end up with, which is it is an entirely separate configurable page for the AirPort Extreme and the Time Capsule where you give it its own SSID, its own crypto key, and similar configuration. And anyone who connects has no access to the LAN. They only have access out to the Internet. So it is, I mean, it's exactly what you would like if someone comes over and they need to use your WiFi. You could give it, you know, still use WPA2, maybe give it a long but simple-to-enter password because you're not going to use it, and anyone who does has no access to your network. They only have access to the Internet. I mean, it's beautiful. It's what you want.

Leo: Great way to do it. Now, we use - we have a wired router. Everybody's on the wired router. And then from that wired router we have a WiFi router for our guests. Is there a way to do something like that with a multi-router setup?

Steve: Well...

Leo: Or would we have to do that three-way router thing that we talked about?

Steve: You really need to do the three-way router. Otherwise you do - so in what you just explained, where you have your WiFi inside of another router, well, it also has access to the network upstream of it, which is your LAN.

Leo: Right.

Steve: So you really need to do a three-router configuration in order to really create isolation which is enforceable in the face of any kind of spoofing attacks. But the point I wanted to make about this is that don't assume that guest networking is always done the way Apple did it, which is correct. There are some guest networking features that you're really going to want to test out before you trust them. For example, some don't block access to the LAN. Some don't allow encryption at all, instead of having separate encryption. So there are guest networking features that are just lame. I mean, they're really - it's unfortunate they didn't do more. They just sort of said, oh, we're going to add that feature so we have a bullet-point on our box. So you really - your mileage may vary. I wanted to note to people that when they upgrade their firmware they may find that feature beginning to appear in later firmware. But absolutely check it out and make sure that, when you log in that way, understand what the criteria are because it may not be something that you want to use, certainly not something you want to leave on all the time. Whereas with Apple's configuration you could absolutely set up basically a parallel WiFi environment that anyone can use where they don't get access to the rest of your internal LAN and other machines, only get access to the 'Net, which is just beautiful. It's the way it should be.

Leo: That's excellent. That's really, yeah, that is very bright. I guess we'll have to just get - the easiest thing to do is get an AirPort Extreme.

Steve: I'm actually considering that myself, Leo.

Leo: They're not cheap.

Steve: Why not? It's just a beautiful solution.

Leo: Yeah. I wish they were less expensive. Gary S. Martin in Tehachapi, California shares a "gray-haired memory." Are you gray-haired? I can't tell.

Steve: Oh, yeah.

Leo: I'm gray.

Steve: Oh, it's gray. It went gray, I don't know when that was, but it sort of snuck up on me. I've looked at some pictures that aren't that old, it's like, hey, it wasn't gray then. But sure is now.

Leo: I sure would love to do another one of those specials with Ray Maxwell.

Steve: Well, and I have to say, Leo, we got a phenomenally positive response, much more than I expected. And interestingly, they were really of two classes. There was the nostalgic response, which Gary's going to share with us here in a second. But then there was also, like, the 20-something response, from people who never had an experience before Windows 98. And it's like they don't remember any of this stuff. So it was really fascinating to our younger listeners for much the same reason. Or, I mean, for an entirely different reason, which is, wow, core? What's core?

Leo: What's that?

Steve: Yeah.

Leo: Oh, the good old days. Well, Gary says: Your special Gray-Haired Computing episode brought back a lot of memories. The first computer I ever saw in person was a PDP-8 at Cabrillo High School, Lompoc, California. I was there for a chess meet. A nerd after my own heart. And it was in their computer lab. They had a Lunar Lander game. Oh, yeah, I remember this. They loaded it from paper tape. It displayed vertical velocity and altitude on the front panel lights. You used a front panel toggle switch to turn on the main engine at the right time to make the vertical velocity go to zero at zero altitude. I believe that it printed fuel, speed, and altitude to the teletype - it did, I remember this - as the game ran. It was the first computer game I ever played. And what a waste of paper.

Steve: Well, and you know, it's interesting because I'm planning to do a set of little, short, easy-to-enter programs for the PDP-8. You know, I call them "toggle toys." And Gary's note made me realize, that's a perfect, a perfect example. For some weird reason I've always loved the Lunar Lander game because it's very simple, yet it's very tricky.

Leo: Yeah.

Steve: You turn the thrust of your little spaceship on and off. And you need to watch your altitude and your height. And of course the goal is not to crash. And so it involves, like, turning your engine on and off just enough to bring the altitude down without increasing your speed, and then a little more at the end there in order to slow yourself down when you're just a little bit above ground, and then touch down. And it's - what I love about is it's pure, simple calculus. It's just integration and, you know, velocity and force. So conceptually very simple. But, you know, it makes an interesting sort of - just difficult enough sort of problem. So, and what I remember is where it would print out velocity, altitude, and fuel. And then it would say, like you'd put in a number from one to nine, how hard you want to burn the engines. And then it would put out velocity - or zero. And then it would do again velocity, altitude and fuel. And so you'd be looking at those, trying to judge as time was going by, as it printed out each iteration. It's amazing. Back then we just thought that was just way cool.

Leo: I think it's actually going to be - your little toys will be a very useful way to learn programming also because in a very kind of basic way you can see how you do what you do. And I think it's not a bad thing to learn how to program at the machine level, to understand what the machine's doing before you get to higher level languages.

Steve: Well, and in fact there are - it's surprising how much PDP-8 resource is on the 'Net. The PDP-8 is specifically often chosen by - like in Assembly language and machine architecture classes, specifically because it is so simple. It's got six instructions, and then some math, a single sort of math instruction, and an I/O instruction, for a total of eight. Because there were only three bits for the opcode. And so it not only - it's the restriction of it that makes it interesting. It makes it easy to learn because there isn't much to learn. But then the way they built it, it's like, okay, how are you going to solve this problem with only two twigs and a toothpick?

Leo: Right, right. Very cool. All right. We have our next question from New Zealand. Bill in Auckland wonders about mixing security on web pages. I see this alert a lot. I'm a web developer, and on a project I'm working on I discovered something strange that deals with mixing secure pages and nonsecure pages. Most web developers know if you have a secure page, and you have an item on the page that points to a nonsecure URL - an image, for instance - the browser gives you that warning, you have a mix of secure and nonsecure items on here. As I understand it, the reason is the potential for information leakage. That's a good thing to notify people about.

I recently discovered if you have a nonsecure page and embed an iFrame that points to a secure page, you don't see this warning. Uh-oh. Also in this scenario the end user has no way of knowing that the page in the iFrame is secure since of course the padlock doesn't appear at the bottom of the browser. To me, I don't see the difference in security between a secure page with insecure assets and a nonsecure page with secure assets. Wouldn't both be susceptible to information leakage? And if not, why? I know from some casual tests, if you try and get information like cookies from the secure page in the iFrame using scripting, that Firefox will return nothing, not giving the outer page access to the secure page's cookie information; and IE will generate an access violation warning. But if I were a determined hacker, I'm sure I could find a way around this. What do you think?

Steve: Well, there's a couple things going on here. First of all, we know how the basic model of a web browser is, that it retrieves the HTML content for the page. And that content may then contain URLs of additional page assets, which it then needs to go and fetch. The way URLs can be written is so-called "relative URLs" or "absolute URLs." That is, oftentimes you'll have a page where the assets on the page just say, for example, \image\reddot.gif, for example. And so what happens is the browser knows to append the HTTP and the flavor of HTTP, either not secure or secure; also the www dot, or whatever the domain is. And it assumes, then, that if the secondary URL begins with a backslash, that that means start at the root of that domain. If it doesn't, if it just said image/reddot.gif, then it would assume that it was relative, that that was relative to the location of the page that's being displayed. So there's a relatively complex set of semantics that browsers universally agree upon. And this was spelled out a decade ago, so it's well established.

The beauty of not having to say http:// and so forth for all of the resources on a page is that, if you choose to display the page nonsecure, that is, with just regular HTTP, then all of the assets will be loaded nonsecure, and so everything is of uniform security. If, however, you did go to https:// and brought that page up, then if the assets did not themselves say https or not, that is, if they just left that part off and said /image/reddot.gif, then the browser, because it uses the whole front part of a URL, or reuses it from the page where the asset is located, it will automatically make connections of the same security as the page. So web designers that have been around a bit take advantage of that to deliberately prevent those kinds of mixed content messages.

What happens is, if the browser, for example, has brought you a secure page, then you tell it, oh, I want you to show this image, and you explicitly say http:// without the "s." If the page is secure, and you're telling the browser I want you to load this thing not secure, oftentimes it is generally configurable to suppress it. But the default setting is the browser will warn you about mixed content. And that's something that users go, uh, oh, wait a minute. Because, you know, they're often wanting to be on a secure page for a reason. And if they get something that pops up and is confusing them, they may just wander away and say, oh, I'm not going any further with this, or I'm not giving you the information that your site is saying that it wants.

Now, the second thing happening is iFrames. iFrames have been controversial because they're powerful and, unfortunately, very exploitable for malicious purpose. In the normal case, which we were just discussing, you load HTML, and then the assets that that page requests are not other pages. That is, they're GIFs and JPGs and cascading style sheets, other non-HTML assets. An iFrame allows you to embed an entire whole page within a page. That is, "I" stands for "inline." So essentially you create, you define a rectangular region, and you say to the web browser, go get another page, HTTP, an HTML page, and render whatever that says in the frame.

Well, it's powerful and useful. But once again, every time you hear me say "powerful and useful" you think, uh-oh, maybe a mild Gibsonian response there because it means there's opportunities for exploitation. And of course iFrames have been a big problem for exactly this reason. Consequently, web browser designers, contemporary web browser designers, have taken a great deal of time and trouble to isolate the iFrame from having any access outside of itself. That is, they'll say okay, fine, we're going to let you display this. But we're going to be very careful to restrict what can be done inside that frame.

So my advice is avoid them if it's at all possible. I had to use an iFrame in the cookie project that I worked on, which we'll be getting back to as soon as I finish the DNS project, because it turns out that there are some mishandling of cookies in iFrames. And the cookie forensics technology that I've got surfaces those and demonstrates when your browser is not doing the right thing with cookies, is allowing iFrame leakage. And in fact there's an obscure bug in Opera that we discovered, I'm not even sure they fixed it yet, where it's possible by doing some strange redirections in an iFrame to confuse Opera and get around its management of that. So they're best avoided, if possible, because they're just an opportunity for problems.

Leo: But they're so tempting. They solve...

Steve: Well, that's because they're so useful.

Leo: Yeah, they solve problems so easily, yeah. But it is really a bad idea, I think. It's bad design. Wayne in Waldorf, Maryland disagrees about what "mini" might mean. He says: In 185 you said "mini" referred to the instruction set, as in mini computer. I let it go until I heard you say that again in 185A. I disagree. Remember that computers used to refer to the room-filling monstrosities. The only way a PDP-8 could come close to that was if the room were a smallish closet. The minimal instruction set type computer was a RISC system, Reduced Instruction Set Computer. Yes, a PDP-8 may have had fewer instructions than an IBM 360, but a PDP-11 had a fair number of instructions - MOV, CLR, OR, XOR, AND, various types of branches along with a byte version of the same thing. And yet that PDP-11 was considered a mini computer. Yeah, I always thought the mini computer was mini because it was smaller than a mainframe.

Steve: Yeah, I mean, I read the document in Gordon Bell of DEC...

Leo: Of Digital Equipment Corp. Yeah, he's the guy.

Steve: Yes. I've read the document in Gordon's own, I mean, that he wrote, where he was explaining that what they were aiming at was a minimal computer. And when you look at the PDP-8, I mean, I'm amazed, I mean, the thing has no high-level integrated circuits. It's just ANDs and OR gates and flip-flops that put this whole thing together. So, I mean, it is truly a minimal computer. What Wayne is getting confused about is this notion of RISC versus CISC, as it's called, a Reduced Instruction Set Computer and a Complex Instruction Set Computer. There the idea was that there was a tendency, as these machines evolved in the early days, and again this is - I've gotten so much insight from reading the original working papers, literally the design papers of the PDP-8 and the PDP-11, which Gordon has published on his own site, where you can really go back in time and remember what these people were thinking. And I read him saying that, when they were going from the PDP-8 to the PDP-11, which was sort of the logical progression from them, he wrote that because virtually all of these machines are being programmed in Assembly language - which, you know, you don't hear that anymore these days.

Leo: No.

Steve: He said, we want to make these machines easy to understand, and easy and enjoyable to program. So they were literally deliberately designing a complex instruction set machine that was a joy to program. And frankly, it's why I can't wait to retire, when I'm 85, and get my hands on one of these, which is why I've been purchasing some of these old machines. I have PDP-11s now, and VAXes even, because the VAX is a 32-bit extension of the 16-bit PDP-11, because oh, my god, they just - they created the most beautiful instruction set. Specifically because they knew that that's what people were using.

Well, of course what they found was that many of these complex instructions were difficult to implement, and expensive to implement, and turned out not being used, especially by compilers. The compilers weren't taking advantage of the power of those instructions. So what's evolved is sort of a different philosophy of using more simple instructions which can be executed more quickly, instead of more singular, more complex instructions which take many cycles to execute. If you're a programmer talking at the machine level, you'd rather have more powerful instructions and need fewer of them to express your intent. If you're a compiler, what's really more useful there is to have fewer, faster instructions that you can mix around in a larger number of combinations. And that's what the whole RISC instruction set approach does. So it's reduced, but it's reduced in computers that are no longer mini, they're generally pretty maxi machines.

Leo: Maxi. Maxi, not mini. Tyler Gurney in Orem, Utah has need of Elaine's help: Steve, Security Now! is my favorite netcast. I anxiously look forward to it every week. I'm a SpinRite owner since v5, and I've used it many times. I'm a proud lurker in your newsgroups, looking forward to CryptoLink, as well as results from the DNS project, the cookies project, the testimonials database, et cetera. I have about eight hours of audio that I need transcribed. I was wondering if Elaine would be a good choice. Do you give out her contact information?

Steve: You know, we've run across this. Normally I just send, when people request it, I bounce email back to them with the URL of Elaine's site. But I thought I would just give her a moment in the sun because she does such a spectacular job of these transcripts every week. Most people, I don't know how many people read the transcripts, but they are painstakingly accurate. I interact with her often several times where she'll need clarification on a term. She uses the 'Net extensively to get the spelling of everything right. When we were first talking, she was doing medical conference transcribing where she's getting every medical terminology correct. I mean, so I love the accuracy of what she does. And I just found her by Googling, and I just thank my lucky stars that it was she who I ended up contacting. I think I did because she has a web-based form that she had at the time, and I was able to fill it out, and she got back to me the next day.

Leo: That's cool. See, have a website.

Steve: Anyway, she's On-Site Media, On-SiteMedia.com, for anybody who ever has a transcription need. I just can't recommend her enough. She's just terrific.

Leo: She's good.

Steve: And she's blushing right now. [Awww.]

Leo: Former - she's a court reporter, so she knows how to keep up. [This is not accurate, although I did study court reporting]. Now, we've been using, as a trial for TWiT, this service, it's called Pods in Print at PodsinPrint.com. You can now get TWiT transcripts, just This Week in Tech show transcripts, as well as Futures in Biotech from them. And let us know what you think about this. These guys, they are a human transcription service. I think they go to India, though, and I'm not sure how they do it. But they're not, you know, they're not handcrafted as much as Elaine's are. But it gives you something to read along, and often can give you some more information as you're listening. So Pods in Print, we're probably going to do more shows with them because - inspired by you.

Steve: Well, I know that Elaine is not the cheapest service around.

Leo: Yeah, we can't afford her for all our shows, I'm afraid.

Steve: Right. David Lawrence asked me once what I was paying, and I think he was using some farmed-out, out-of-the-country service that was much cheaper. But there's just, I mean, I care about the quality. There's no way I would consider anything, so...

Leo: Well, and your show, I think also, every word counts. On TWiT, every word does not count. I can tell you right now. Many words should just be left out.

Let's talk about ZIP security. Joe Sturonas in Milwaukee, Wisconsin: Steve and Leo, I'm the Chief Technology Officer of PKWARE - love it - inventors of the ZIP file format. As such I'm writing in response to the Security Now! Episode 184, where you commented that you heard ZIP uses AES now. I happen to know a little something about the ZIP file format. PKWARE is the custodian of the APPNOTE - that's at PKWARE.com/appnote.txt - that defines the ZIP file format.

To answer your question directly, as the inventors of the ZIP format, we have updated the standard format to support strong encryption capabilities using industry standard encryption. Yay. Combining AES with X.509 digital certificates provides sufficiently durable file protection that is every bit as strong as PGP. Support has been added to ZIP files with AES using either a symmetric key, a passphrase; or asymmetric key, which is X.509 V3 digital certificates; or both simultaneously. Additional security is provided using a digital signature, SHA-1 or SHA-2, to provide authentication. Signatures can be applied to individual members of the archive and/or to the entire archive itself.

Strong security is built on top of the standard ZIP invented by our founder, Phil Katz, so all the capabilities people have been using for over 20 years are still there. And now you can strongly protect files as easily as ZIPing. So you ZIP and encrypt. ZIP encryption provides a platform-neutral solution which can be used across all platforms. What other technologies have been able to stand the test of time for over 20 years in this industry? Okay, other than SpinRite.

Strong security in ZIP was developed as a hybrid cryptosystem. A hybrid cryptosystem uses an asymmetric public-private key combination to encrypt a symmetric key that is used to encrypt the files. That's what PGP does; right?

Steve: Yup, and that's - and we've talked about that approach many times.

Leo: It's a good way to do it.

Steve: It's exactly the way to do it.

Leo: Since the hybrid approach applies the compute-intensive asymmetric encryption to only the small symmetric key, it consumes minimal processing while providing fast, effective encryption with standard symmetric algorithms like AES. So the body of the message is using a symmetric key, but the symmetric key is passed using an asymmetric public-private system.

Steve: Right. So if you were specifying only public key encryption, it handles the symmetric aspect of that transparently for you sort of underneath that. You don't need to worry about it.

Leo: PKWARE's solution for strong security is available in SecureZIP, an advanced version of the familiar PKZIP program that includes this strong encryption support as part of the standard feature set. SecureZIP Express for Windows is free for noncommercial use. Right on.

Steve: Which is really neat, yes.

Leo: Yeah. It also includes a Wizard to get a free digital certificate, as well. SecureZIP.com is the URL for that. He says it runs on z/OS, which is a mainframe operating system; i5/OS, which is for AS/400; UNIX; Linux; Windows Server and Windows Desktop. I don't see Mac in there.

Steve: I know, I was wondering if maybe being covered by UNIX would do it. But probably not.

Leo: You know, Apple builds a ZIP and UNZIP into its OS. And probably they felt there's not going to be much of a market in that case on the Mac side?

Steve: Yeah, but XP has this miserable zipping folder thing, too. In fact, I just heard Paul denigrating it before we began recording our podcast.

Leo: Right. We use IZArc, or he recommended 7-Zip. But neither of them do this. Now, it's interesting because it sounds like this is part of the standard. So presumably it could be implemented by other...

Steve: Right. It is now part of the standard.

Leo: Yeah. Thanks for your unrelenting persistence to never miss a week of Security Now!. Keep up the good work. Joe Sturonas, CTO of PKWARE. That's so cool. It's good to hear about them. That's a great company.

Steve: Well, yes. They've been around forever. In fact, I licensed their 16-bit toolkit 20 years ago in order to add inflate and deflate capabilities to something I was doing. I don't remember now what it was. But, yeah, I mean, these are the guys. And I really think it's very cool that SecureZIP Express is available for free for noncommercial use.

Leo: Isn't that awesome.

Steve: I'm going to check it out. And in this new approach we've talked about of doing some more, a little bit less techie and a little more practical application episodes, if it looks like it's warranted, I'm going to consider doing a Security Now! on exactly what the features are and how they work.

Leo: Excellent.

Steve: Yes.

Leo: Remember before ZIP it was ARC?

Steve: Yeah.

Leo: And I don't know what there was before ARC.

Steve: And in fact there were some - was it, oh, boy...

Leo: There's RAR.

Steve: SEA, SEA.

Leo: SEA, Self-Extracting Archive. But that was a ZIP format, I think.

Steve: But I thought that there was some battle over patents or formats or something. I remember there was a big, messy legal battle.

Leo: Oh, yes. Yes.

Steve: I don't remember, maybe it was between ARC and ZIP. And one way or another that got resolved, and we ended up with ZIP.

Leo: Thank goodness there's a standard, de facto. That was a case where the market really did a good job in settling it. The market is not notoriously good at creating standards, however.

Steve: No.

Leo: Tom Stewart in Arva, Ontario, Canada wonders about the security of open source software. I'm ready to put on the gloves on this one. Hi, Steve. Love your Security Now! podcast. My question relates to security and open source. I probably don't understand the open source process. My impression is the source code is made available to other parties who could then modify it to suit their needs. That's correct. How are standards maintained for open source software? And what is to prevent a version of Linux or other open source software, even security-focused applications, from having malware inserted and being distributed? Would it not be easy to put in a trojan which could become active at any future point? Thanks for the show. Like many other users I purchased SpinRite in appreciation for your efforts to put this show together. Well, that's a nice way to donate.

Steve: Yeah.

Leo: I have benefited from both listening to the podcast and running SpinRite on my failed hard drive. Well, what do you say on that?

Steve: Well, I have a much more pro open source position, relative to his question, than you might expect. I think...

Leo: Oh, no. I wasn't going to box with you.

Steve: Oh, oh.

Leo: I wouldn't box with you. I would box with him on this one.

Steve: The way the process normally works, as I have seen it, is you've got a couple custodians of the master source archive, and a lot of people who are involved who come up with basically little patches. I mean, once the program is pretty much established, there aren't lots of changes you normally need to make to it. And anything that's changed is done in a highly collaborative mode. So, for example, someone will submit a patch which says, okay, change the following lines from this to this. And then a lot of people will look at it, they'll scrutinize it, and if it looks like it does the right thing, it'll get checked in.

Now, certainly errors happen everywhere. They happen in closed source software. They happen in open source software. So my feeling is the process has every bit as much chance to prevent a problem as a closed source approach. Now, the one thing you don't want to do is to accept a large software system which is open source from somebody who says, oh, hey, I've improved Linux. Here's Joe's Linux. Install this on your system and use it and trust it. Because there Joe has gone off...

Leo: Well, first - there's a couple of issues. First of all, you can't call it Linux.

Steve: Right.

Leo: Because one thing Linus did do is preserve the copyright. So if it says it's Linux, I mean, I guess some guy could be a rogue Linux, but it wouldn't take long before the world would beat a path to his door.

Steve: Well, and I'm saying, you know, I didn't mean like in a major production environment where he's gone into business. But it's like, hey, you know, take my version. Because the source is open, it is absolutely easy for someone to exploit it for malicious purposes. And the point is they probably can't get that back into the master source tree, but they could certainly generate their own evil version and then try to get people that they know, or maybe don't know, to somehow run it, and you don't know what you're running. It's been one of the arguments, for example, against my, for me, publishing the source for my security-related tools because I was concerned somebody could take one - I mean, they're all very small and tight. The fact that they're in

Assembler would tend to hamper most people from messing with it. But still there are lots of people who know Assembler, especially given something that's working. My concern was somebody could take it, mutate it into something bad, and say oh, you know, here's a copy of GRC's whatever.

Leo: Right. The difference is, in order to do that he'd have to make it open source. So you could look at it, and you could see what he did. I think open source is always more secure. But you have to consider where you're getting software from in any case, in any event.

Steve: I guess I would say that with the openness of open source comes responsibility to make sure that the availability of the source code isn't misused.

Leo: Well, I would submit that you have to be careful who you take applications from in any event.

Steve: Right.

Leo: Any application could be malicious, open or closed. The advantage...

Steve: Yes. Because mistakes happen.

Leo: Yeah. Open source doesn't make it more prone to that. I can take your source code, disassemble it, and stick a bad thing in it fairly easily - that's not a hard thing to do - and then reassemble it. And then say, hey, I've got Leo's version of SpinRite.

Steve: Right.

Leo: You'd be stupid to take it from me. So I don't think open source makes that easier by any means. It's a trivial thing to add a branch to disassembled code that goes to a bad thing and rebuild it. That's not hard to do. The advantage is it's open. So if you're getting open source code from somebody who's made it closed, he's violating the license. Don't take it from him. If it's open, presumably people are looking at the source code saying, what's he doing here?

Steve: Right, and running a big comparison on...

Leo: And you can do it if you want, yeah, I mean, you can look at it. And just take it from reliable sources. But any time you take software from anybody, take it from reliable sources. That's always a risk. I don't think open makes it any more risky. In fact, I think it makes it less risky.

David Popovich, IT Support in Stuart, Florida, wanted some GoToMyPC clarification.

He says: Steve, I've been listening and learning from you for years. My question is about GoToMyPC. Leo's ads often say you can use GoToMyPC anywhere, and because of SSL it's secure. A few weeks ago I heard Leo say you could use it at an Internet caf. In fact, I just said it, at a shaky Internet caf on Sunset Strip. This immediately bothered me because I wanted to hear from you as to why this is considered safe. If there's a key logging program on the Internet caf computers - well, okay. You know what, there could be somebody looking over your shoulder, too. There could be somebody ready to hit you in the head. That doesn't make it safe. GoToMyPC is not going to protect you against everything.

A compromised system would be like handing the bad guys the keys to your entire system. GoToMyPC is only as secure as the system you run it from; right? Of course, running it from your own laptop in a WiFi hotspot through SSL is okay - see, the guy is misunderstanding. I'm not saying it's safe in every case. There could be a camera over your shoulder. You using your own laptop, it's not safe because it could be looking at your keystrokes. I'm saying the SSL is like a VPN. VPN's not safe in that context either.

Steve: Just you refuse to get to the end of this question.

Leo: All right, all right. I mean, it's - okay. Of course running it from your own laptop with a WiFi hotspot through SSL is okay. No, it's not. Same problem. Somebody could photograph you. But the advertisement doesn't clarify this at all. So using GoToMyPC on relatives' computers, library computers, et cetera, would all be taboo; right? If the answer to my query is not favorable to GoToMyPC, then I suspect I may not hear a reply to this during your podcast. I realize they're an advertiser. If that's the case, I know your time is precious, but I would love to get the answer, even if by email. Thanks to you and Leo for the great insight into today's wild, wild west online. Steve?

Steve: So I completely agree with you, Leo. And David is asking for something that even my own program, CryptoLink, won't protect you from. That is, I mean, the message here, though, the reason I wanted to present the question, is that it's a perfect example of understanding the threat model that we've talked about, what it is you're protected from, and what it is you're not. So...

Leo: But he's right to be aware of keystroke loggers, absolutely.

Steve: Exactly. And so certainly it's the case that, if you were at an Internet caf, and you were going to use this system from somebody else's machine that you can't vouch for, or even, Leo, as you said, from your own laptop, but you might have somebody watching you enter your username and password, then that's part of the threat model that you need to understand. So what a VPN system is offering is protection against a class of problems. And certainly there are things it's not protecting you from. For example, if GoToMyPC offered, like, the use of the VeriSign tokens, for example, the little football or the credit cards, a one-time password system, that would change the threat model, change its security in a different way than if you're using a static password. And that's something certainly our listeners well understand.

Leo: Yeah, yeah. I mean, software can only protect you from so much. Nothing's going to protect you if the ceiling falls in. Let's see. Riley Willcox in Truckee, California wonders about sniffing email content: Hi, Steve and Leo. Thanks for the great show, keeps my brain stimulated. I try to be a safe emailer and connect to my mail servers via secure protocols. However, once email is sent out from these servers, it's no longer protected. For really critical emails I can encrypt the content, but I'd rather do that only when necessary. Few people are actually set up to receive PGP or GPG-encrypted email. My question is, if an email is not encrypted, how hard would it be for someone to grab that content between my email server and the destination server? What techniques would an attacker use? That's a good thing - again, a threat model - good thing to keep in mind.

Steve: Yeah. And I thought this was a great question because a lot of people, I think, have a hard time visualizing the inner workings of an ISP, a datacenter, like what is the Internet? They see it as a box on their desk that they've plugged things into, and then sort of magically disappears. The question then being, well, okay, are there wires hanging outside of the ISP that some guy dressed in a telephone uniform with spiky shoes could climb the phone pole and connect to. So I wanted to sort of paint a picture for people about what happens to their data once it goes wherever it goes.

As he says, if he were to use a secure connection to the servers, for example using SSL-encapsulated email, then there's no way for anyone to see his email as it's leaving his machine, as it's getting essentially all the way to the ISP's server; whereas as we know, SSL, because it does endpoint-to-endpoint encryption, it would then decrypt it as it's sitting on a server. So now it's on the server in plaintext format. And it's going to remain so from then on.

So the threat really is not, I think, to any degree that's substantial, from random people. Certainly the ISP, which is to say the ISP's employees, do have access to all that. I mean, the networking level people, the people in the network operating center, the people who tend the servers, the people who are making sure the connections stay up and running, I mean, they're in the middle of the data. So they have access to it. They have access to everyone's email that is in this form, that is not maintained and packaged in its own encrypted envelope as you're sending it to its destination. So if nothing else I guess the threat is watered down, unless for some reason you were targeted. In which case, if there was some reason for an ISP or an employee, a rogue employee, to get all of the email or traffic or something to or from one specific customer, they have access to it. I mean, that's their job. This data is there, and nothing protects it.

Leo: You know what I liken it to is the post office. When you send a - think of it as sending a postcard. It's going through the post office completely available to anybody who wants to look at it. And the mail carrier and everybody. We trust them.

Steve: Or we just trust in their disinterest. And in fact that's one - so that leads me to the other threat, which unfortunately is Big Brother. We've been through, post 9/11, an increase in surveillance. We don't know to what degree it affects us individually. We pretty much - I hope it pretty much doesn't affect me. That is, I have no reason to believe that it would. But we know that there have been technologies employed where ISPs have had devices installed that literally read everything, that are filtering for keywords and performing matches. And this was FBI technology at one point that was discussed and apparently deployed.

Leo: Carnivore, yeah.

Steve: Carnivore, exactly. So that's really a problem. We do know that law enforcement is unhappy about VPN, the growing use of VPN technology, because it blinds them to the actions of bad guys. I'm not happy about blinding anyone to the actions of bad guys. But I also want to blind malicious hackers and nongovernmental bad guys. So unfortunately law enforcement loses in the process. It would be nice if only good guys protected themselves, but everybody can because it's powerful technology.

So I guess, anyway, to wrap this question, the idea is that this stuff is vulnerable. I think what we have to assume is that the people who are charged with tending it don't care, whether that's Big Brother, who might be scanning everything, or random ISP employees that have better things to do than read random people's email. We know that random people's email does get read sometimes. Sometimes it makes the news. Most often it doesn't. So I guess the answer is, if it's really the case that you don't want your mail read, you need to go to some measure to protect it until its recipient receives it. And that's either using something like PGP, or SecureZIP, as we just learned about, or some technology like that.

Leo: I guess the other difference is the post office there's federal laws protecting you and protecting your privacy. Which I don't think there is in email.

Steve: Sure, wiretapping laws would...

Leo: Oh, maybe, yeah.

Steve: Yeah.

Leo: Okay. I mean, I assume that every email that I send could be read unless I encrypt it. Period.

Steve: Leo, mine are boring. I'm just...

Leo: Yeah, go ahead and read 'em.

Steve: Someone's reading that, it's like, oh, well, then that means they're not doing something else, so...

Leo: But you could say the same thing about any phone call that you make.

Steve: Right.

Leo: Unless you explicitly enforce privacy, you don't have any.

Steve: I've sometimes been cautioned, when I'm having lunch with my attorney, to keep my voice down because it tends to go up. And I need that reminder. I'm glad for that reminder because you don't know who's at the table next to you. I mean, probably no one. Probably I'm just annoying them rather than interesting them.

Leo: Well, that's it. I think relying on disinterest is like relying on security through obscurity. It works most of the time. But not all the time. And the problem is that somebody who is interested probably doesn't have your best interests at heart.

Steve: Right.

Leo: Last question, from Pierre in Canada. Pierre wonders about the wisdom of defragging today's huge hard drives: Hi, Steve. This week a colleague told me about an article I then read on the Internet and asked me what I thought about it. I wasn't sure what to answer him because the article seems to make so much sense. So I thought, hmm, who would know better than the maker of SpinRite? What we're wondering is, according to this article, with hard drives getting bigger and bigger, defragging could have more negative impact on the drive than positive impact.

The two arguments are that, first of all, with a drive of, say, 500 gigs or a terabyte, the time it takes is getting so long, that's a lot of work on a drive for a very long period of time. The stress could reduce the lifetime of the drive. Secondly, hard drive transfer rates are getting so fast that the gain is not significant. Why bother to defrag a drive if you don't need to? So what do you think about this? Good job with SpinRite. I'm a programmer working in an IT department. We use SpinRite every day. Wow.

Steve: Well, that's an interesting question. One of the things that we learned with SpinRite 6, when I first incorporated real-time S.M.A.R.T. monitoring, where SpinRite is periodically polling the drive's S.M.A.R.T. data, one of the things SpinRite shows and watches is drive temperature. And we learned that a lot of drives, especially laptop drives, easily get overheated. Many people have had SpinRite stop, as it will, and warn them that their laptop drive is now at the manufacturer's upper limit of temperature. And sometimes this even happens with people who have desktop machines. Sometimes desktop machines that were designed for an earlier generation, smaller, and less power-hungry drive, people will add a drive to the existing enclosure or replace a smaller drive with a bigger one, which may draw more power and generate more heat. So the ventilation, which was adequate for the cooler drive, isn't for the hotter, more power-hungry drive. We've seen that a lot. So the only real downside I can see is you really want to make sure that your drive is not overheating. Unfortunately it's not easy to tell that. I mean, SpinRite has it built in, but defragging programs don't.

The other thing is, I would, if the question is I'm worried that defragging the drive might cause it to fail, then my reaction is, whoa, you absolutely want to solve that problem first. That is, one way or another you want to be in a situation where a drive failing cannot really hurt you badly. Which means you either backup enough; or you've got a RAID configuration so you've got some redundancy, so if a drive dies you're able to survive that. Because, believe me, I mean, I'm in the business. Drives do die. SpinRite can fix a lot of them; but, you know, sometimes when the heads fall off or they seize up or they burn out or something happens mechanically, there's nothing any software could possibly do. So...

Leo: Do you think you need to defrag drives these days? Is the speed sufficient that it's not worth it?

Steve: Well, it's interesting, the...

Leo: [Indiscernible] time is always going to be an issue; isn't it?

Steve: The transfer rates have gone up, but the hunger for data, I think, has gone up just as fast, if not faster, than the transfer rates. I mean, our systems don't feel particularly faster than they used to. I mean, when Windows 7 boots up in less agonizingly long time than Vista, it's like, wow, that's faster. But we used to turn our machines on, and they were booted by the time the CRTs warmed up. Those days are long gone. So my sense is that drives are faster, but everything is bigger. And so the bigness is completely offsetting the increase in data transfer rate. I still defrag.

Leo: You do. How often?

Steve: I don't do it fanatically. I saw that my defragger was running, that is, the services, I've got two services that run with the third-party defragger that I'm using. And I thought, oh, that's dumb, and I went and manually turned them off. But it reminded me that I hadn't defragged in a long time. My routine is, when I'm - I'm running a RAID, so I've got redundancy all the time. But every so often I want to take a snapshot. So what I'll do is I'll just go through a housecleaning. I will empty caches. I'll look at the size of my - I'll sort by file sizes, get rid of a bunch of junk that I've just sort of accumulated through daily operation, so sort of trim the system down to a minimum working set size. Then I defrag it. Then I make an image of it. And so that's sort of just an afternoon, or maybe a couple hours of when the mood hits me I decide, okay, it's time to do a little maintenance. I get rid of everything I can first, then I defrag it, then I make an image, and then I feel good.

Leo: Good. I'm glad you feel good. I've been for a long time kind of one of those guys who says defragging is voodoo; or, not voodoo, but it's overrated. Do it every few months, that's all you need to do. And...

Steve: It also has the advantage that it's a little bit of a poor man's SpinRite because...

Leo: Right, it does check it, doesn't it, yeah.

Steve: It does make the drive go and move stuff around. And moving stuff around, it makes it read it. And if the drive discovers sectors that are beginning to be problematical, the drive can relocate those to better physical sectors. So it's not just logically moving it, it's actually physically retiring sectors that are bad. You don't get any guarantees that it hasn't missed spots, which is where SpinRite comes in because it does a whole drive read, essentially, and re-verification. But it has a beneficial effect, I think, that is probably greater than its deleterious effect, so long as you're not overheating.

Leo: All right. So keep it cool, man.

Steve: Yeah.

Leo: Steve, a great 12 questions, as always. We love our listeners. You guys are smart and always raise interesting issues, and we're so glad you write. Go to Security Now!'s web page, GRC.com/feedback, to submit questions for our next Q&A episode, two episodes hence. Of course, while you're at GRC, buy a copy of SpinRite, download all of those great programs, check out the show notes, the 16KB versions of the podcast, and all of that stuff, too. It's a great website: GRC.com. And Steve, we'll see you next week.

Steve: Talk to you then, Leo. Thanks.