Listener Feedback #112

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-290.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-290-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode 290, recorded March 1, 2011: Your questions, Steve's answers, #112.

It's time for Security Now!, the show that covers your security online; your privacy, too. Here he is, the man of the moment, the man of the hour, Steve Gibson. Actually the man of the hour and a half.

Steve Gibson: I actually got email from someone who was rather irate because I think maybe we broke a record. No, we couldn't have broken a record. It was, I think, like 101 minutes last week.

Leo: Oh, not even close to a record.

Steve: And he said, "Come on, Steve. I only have so much time." And he said, you know, "Cut out the fluff and condense the podcast." It's like, okay, well, I heard you. So I actually responded to his email. And I said, "Okay, I just want to let you know you've been heard." But we have fun with the podcast. We don't rehearse this. So it's not possible for us to exactly plan everything. We do have some fun diverging from time to time.

Leo: I'll give him his money back.

Steve: There you go.

Leo: Actually, this is the least fluffy of all the shows. So if he thinks this show is fluffy...

Steve: That's a very good point.

Leo: ...I got bad news for you. Most of the other stuff is longer and fluffier. But, no, one point he does have is that the shows have been getting longer and longer. And I don't know what to say about that. We could make them shorter, I guess. I don't know. I don't know. I'll tell you what. If I heard from a huge number of people, I might. But most of the time all I hear is, we don't mind, we like it long. We want more.

Steve: And I think in our case, from the feedback I get, certainly people are loving the fact that they're getting a lot of factual stuff; that they, even people who've been around for a long time, are running across tidbits that they hadn't encountered before. But also they do appreciate it from an entertainment value standpoint. So they're listening to have some fun with us.

Leo: I think that's the best way to approach this is just figure we're in the background. You're listening. You're learning as you've got some company in the room. It's this, or you could be listening to Dr. Laura. So you take - she's off the air. But you take your pick. It's just like radio. It's just keeping you company. And in this case you learn a little bit of something. So those of you who are tuning in live, we will do MacBreak Weekly tomorrow, along with the iPad announcement. And Steve's very kindly moved us over. Normally we do this show on Wednesdays at 11:00 a.m. Pacific, 2:00 p.m. Eastern on live.twit.tv. I'll tweet out to let Security Now! fans know that we're going to begin early. This is a Q&A episode.

Steve: Yeah. The big news from the week, which I finally had to tweet to everyone that I knew about this because all of our listeners were sending me tweets, saying hey, Steve, have you heard about the LastPass cross-site scripting vulnerability which was uncovered? And so in order to, like, stem the tide of everyone making sure I knew, I sent out a tweet saying, yes, I know about it, and we're going to deal with it in detail in this week's podcast, which we will do here in a few minutes.

Leo: Oh, okay. So that's coming up in just a little bit. That's good. All right. It's nice, yeah, to give them - because people want you to cover the latest, but of course we don't want to cover it until we have something to say about it. But we will talk about it. Okay, Steven. Let's see here. What do you want to start with? The security news?

Steve: I'm going to tell Greg, my tech support guy, about that. I let him do, like, computer PC fixing stuff sort of on the side. And this would be fantastic, I would think, for him invoicing customers. And I was thinking also for collecting payment.

Leo: I used it for years until I had Lisa and a staff to do this for me, and it got more

complicated. But it's just great. I love it. FreshBooks is so cool.

Steve: So we do have Service Pack 1 for Windows 7 - and also I think that also covers Server 2008, if I recall - has been released. And not without some problems. There have been instances where some security software, third-party security software has caused SP1 problems. And, I mean, there's been enough of a buzz that I found it interesting. Brian Krebs, our illustrious security blogger and researcher, came to the conclusion, based on everything that he had seen, that if you've been keeping up with security patches all along, as I'm sure all of our listeners have, his feeling is, since all SP1 is, is everything that's happened so far, don't bother. And, for example, maybe only use SP1 if you're doing a fresh install. You'd certainly want to do that because you'd install the original Windows 7, then immediately SP1, which probably saves about three hours' worth of ridiculous, endless, individual security patch updates, and then updates to those updates, and updates to those updates and so on.

My feeling is, based on our experience, Microsoft won't let us not install SP1 in the long term. Ultimately they seem to get a little antsy about service packs that haven't been installed and start bugging us more and more. Even if you told them, don't bother me about this anymore, it sort of comes back. And then it's like, okay, why am I being bothered about this? Well, you really need to install the service pack. So I would say there's no hurry about installing it. You get no substantial new features. Nothing about it is necessary, especially, well, given the fact that you've been keeping yourself up to date.

Leo: That's exactly what Paul Thurrott said, too. We covered it a little bit last week. He's in agreement. But if you haven't been keeping it up to date, here's a chance to get them all at once.

Steve: Oh, yes, and save yourself a substantial amount of time.

Leo: A lot of IT guys do this. If you go to Windows Update, and you look at the catalog, you could download this service pack as a standalone file. So if you've got a bunch of machines that you haven't been updating, or you're an IT guy, and you want to update the whole office, that's the way to do it. You put it on a CD and then go around and update all the machines. You can get the full update.

Steve: Yeah, it's big. I think it's, like, half a gig, 500-something megs, as I recall, for the full monty.

Leo: It is better to use Windows Update, Paul said, because Windows Update will check for dependencies. So that if that machine needs something before Windows 7 SP1 is installed, it'll get that first so you don't have some oddball situation going on. So that's the best way to do it.

Steve: Yeah, I will be installing another instance of Win7 in a machine shortly. And so I was really delighted. In fact, what I got, because I am a MSDN member, I just got Windows 7 with SP1 already prebundled. So saves a lot of time.

Leo: You bet.

Steve: In the news, something just sort of joggled my security filter. In Cheshire, in the U.K., keystroke loggers were found on library computers. They don't know how long they've been there. They don't know who installed them. They don't know who's been back to dump them. But these were hardware blobs that we've talked about years ago, inline in the keyboard. And so what they've done as a policy is they've changed their physical structure so that the keyboards are plugged into the front of the machines, so they must be USB keyboards. They're probably plugged into a front USB port rather than in the back, just so that it's more obvious if there's a blob that's been plugged in between the keyboard and the computer.

But the point of my raising this is I just wanted to sort of remind, just as a little general sort of signpost reminder, that, as a general rule, you cannot find less security anywhere than in a publicly accessible PC. A library computer - and we have talked about this before, but I just wanted to reiterate because I think it's so important. A computer that you don't have pretty much constant oversight over, you just have no way of knowing what its short- and long-term history has been. And there's no safe way to use it. Even - and this is a perfect example - even if you were using SSL, you had HTTPS Everywhere, you turned Facebook's HTTPS enforcement on - although we're going to talk about why that doesn't really work, either. I mean, even with all that, if you were to log on or purchase something and enter your name and your address and your credit card information, and there was a physical keystroke logger between your keyboard and your computer, it would capture all of the things you're typing in. And anyone looking at a log could easily parse out your name, your address, your billing address for the credit card, and the credit card number. Or your login credentials. And somewhere you're typically typing in a URL, and so they could see you doing that, so they would know what site those credentials applied to.

So just by looking at the log of keystrokes, you pretty much, I mean, that's very powerful, which of course is why some bad people took the trouble to put keystroke loggers on those library computers. So I just wanted to sort of, as just a little signpost reminder, if you have some use for a library or a kiosk or some public access PC, just have your guard up past the red level. There's just really nothing safe you could do except completely passive browsing, where you're just putting things into Google and clicking on links and being passive and not getting sucked into providing any information on an inward direction.

Leo: One thing I like, though, more and more on browsers you'll see, or I guess websites do it, too, "Don't check this if you're on a public machine." They are at least kind of warning you, don't save that password, things like that.

Steve: Yes, and that's really, really good to see, that we're beginning to get this kind of pervasive understanding of what the dangers are. Basically it's once upon a time, a company would have been reluctant to do that because they would have felt they were discouraging people from something that they wanted to do.

Leo: Or scaring them.

Steve: Precisely. Now it's like, oh, they're being responsible to remind us that this is a danger because everyone really understands that's the case. Speaking of which, there was an interesting blurb from our old friend Robert Graham. Robert Graham was one of the founders of Network ICE, that did the BlackICE firewall that was very popular years ago. And he poked his head up in public, commenting about the new Intel Thunderbolt 10Gbps I/O bus, which the latest Apple laptop computers are famously going to be having installed in them for the first time. And what Robert mentioned, I just wanted to note here, because that's what we do; and that is, in exactly the same fashion as the FireWire bus is a security concern, so is Thunderbolt. Essentially, Thunderbolt is the PCIe bus, which is now the bus that links all of our components together in our PCs. It is that bus serialized.

Leo: I didn't know that. That's interesting.

Steve: Yeah. And what that means is that a device that is on Thunderbolt has DMA, Direct Memory Access, to your machine's RAM. So we did talk about this years ago with FireWire. There were some exploits. In fact, HBGary, the company, the government contracting security firm that's been in the news a lot lately, they provided the government with a device, a FireWire-based device, which you could just plug into any FireWire-based machine, and it would instantly clone the current state of its memory that allowed people who had this device, for example, to steal cryptographic keys that were in use at the time, directly out of the machine's memory. We've talked about, for example, in the past, lowering the temperature of RAM immediately after turning a machine off, like spraying it with Freon in order to...

Leo: To hold it, yes.

Steve: In order to hold it, yeah. And it's surprisingly effective. And we talked also about how, due to the fact, the way cryptographic algorithms work, you take the key, and you do something called a "key expansion" where, for example, in the case of AES, which we covered on an entire podcast before, you take a 128-bit or 256-bit AES key, you run it through a key expansion, which takes that relatively modest number of bits and algorithmically expands it into a block of information. It does that because the symmetric cipher is iterative, and it has to do what it does, for example, either 11 cycles in the case of a 128-bit AES key or 14 cycles in the case of a 256-bit AES key. So in order to feed the cryptographic algorithm 14 times, you need much more data than just 256 bits, which is the source key. So you sort of - you expand it, and that's called building a key schedule.

Well, the problem with that is that, in doing that, in scheduling the key out to full size, which is what has to be done in RAM in order to use the cipher in real time, if anything is using the cipher, like if you've got TrueCrypt running or anything else, then there's this block of memory which has to be there accessible to the computer in order to perform the encryption and decryption on the fly back and forth. Well, the act of creating that dramatically increases the redundancy - reduces the entropy, increases the redundancy of information.

So what the researchers found was, if they Freon-sprayed RAM, even if there was some deterioration of the data, because they knew where the data came from, they could find it in memory. They could sort of lock onto it, and they could figure out what the original key was, even from a corroded expansion of that key. So I guess the point is that giving

anyone access to the running contents of your computer's RAM is really a bad thing to do.

Leo: It's all in there.

Steve: It's everything. It's your logons.

Leo: Decrypted keys, things like that?

Steve: Decrypted keys, everything. It's the running state of your machine. It's the mother lode for a bad guy. And FireWire, due to its nature, allows that. Well, so does Thunderbolt. And so on his blog, Robert Graham painted a picture of some presenter is in a conference and goes up to the podium of the future and plugs his MacBook into a Thunderbolt port...

Leo: For the video.

Steve: For the video.

Leo: But somebody's tapped into that.

Steve: Exactly. Well, and because you have read/write access, a sufficiently clever hack would be to download some code that dumps the hard drive. And at 10Gb, it's not going to take that long to suck the hard drive out through the port.

Leo: Wow.

Steve: So anyway, I just - I wanted to put this on people's radar. Now, it's important not to get too concerned about it because not only does Thunderbolt do this, but pretty much all the Apple ports do. FireWire does. ExpressCard does. And even the SDIO slot allows this. So these are all connections into the system's memory. Now, in the case of Thunderbolt, there is chipset support for imposing limits on the range of memory that is accessible over the bus. But as far as we know, the Mac OS isn't yet exercising those features of the chipset.

Leo: They do have - Lion is in development right now. It's close to coming out.

Steve: And I was just going to say, since Apple's giving so much security focus on this next iteration of the Mac OS, maybe, especially after listening to this, they'll do something about it.

Leo: Yes. That would be a good thing to put in Lion. So you could constrain the amount of memory that Thunderbolt could see. Is that how it works?

Steve: Yeah. You would constrain the range. Like you set an upper and a lower limit to where, for example, to the display buffer, so that you're only able to access the display memory which you're wanting to export to a remote monitor and not export the entire contents, 4GB address space of the machine, which it's no one else's business.

Speaking of Facebook, someone tweeted me, and I remembered that I had seen this now a number of times, and so I pursued it because I wanted to get the whole story. And I have verified. We were excited that Facebook offered recently the ability to force HTTPS full SSL connection security as a user-configurable option for individual users' accounts. The bad news is that it was known that some Facebook apps themselves may not support HTTPS. You may be trying to connect to a server that just doesn't have an SSL certificate.

Leo: The apps are served by the creator, not by Facebook. And I would bet you a lot of app creators haven't bothered with an SSL certificate.

Steve: Correct. And so what happens is, if you try to, from in Facebook, go to a Facebook app that doesn't support HTTPS, you get a dialogue box that comes up. And the title on the dialogue box says "Switch to regular connection (HTTP)?" And then in the text in the dialogue, it says, "Sorry, we can't display this content while you're viewing Facebook over a secure connection."

Leo: So at least you'll know.

Steve: Oh, yeah. "To use this app," it says, "you'll need to switch to a regular connection." Now, here's the bad news. If you do that, it turns off the option in your configuration permanently.

Leo: Oh. Not just for that session, but forever.

Steve: Yes. It just puts you back to the Stone Age. Back the way we were last year. So that's unfortunate.

Leo: But people, if you really are concerned about security on Facebook, you shouldn't be using those third-party apps anyway. I know it's tempting and fun. But those things leak your information like a sieve.

Steve: That's a very, very good point, Leo.

Leo: Stay away from them.

Steve: And I would say, given that Facebook now supports this, something like Force HTTP or HTTPS Everywhere, using those add-ons for Firefox, which will keep Facebook back in a secure mode, even if it's not enforcing it itself, you can enforce it at your client end. That's certainly the way you'd want to go. But I wanted to make sure that people knew, since we were celebrating the addition of this feature to Facebook, that they hadn't quite done it the way we wished they had.

Leo: I don't think they have any choice because they'd have to compel all the app developers to go SSL.

Steve: I don't understand, though, why they don't allow an exception for, like, non-Facebook domains, when necessary...

Leo: Right, but keep it on.

Steve: ...but keep the settings set for themselves. To me it sounds like it was a quick hack; that like they, oh, shoot, we need to turn off HTTPS for our app, so we'll turn them off for everybody.

Leo: It's also that convenience versus security thing. They don't want to bother grandma.

Steve: Ah, good point. Because what this would prevent is it would prevent you from being prompted with that dialogue every single time you use a noncompatible app.

Leo: Grandma's saying, I don't care, so they turn it off. That wouldn't be, yeah, I can see why they did it that way, but still.

Steve: So you mentioned earlier the Gmail email lossage. It's not really a security concern, but it just sort of came on my radar. Google is estimating about 150,000 of their Gmail users, which they're saying is 0.02 percent of Gmail users lost all their email. Just gone. Whoops. Don't you hate when that happens?

Leo: Bye bye.

Steve: They're saying that it was a storage software update which was buggy, which they pushed out across their network. And people had said, wait a minute, I thought you were replicating our email in multiple sites all over the place, specifically so that an outage couldn't happen. And they said, well, yes, that's if there's no bugs. But if there's bugs, then the bug replicated the deletion of all of the email in all those accounts. And from their blog - well, the good news was nothing was permanently lost. It was taking them a lot of time to recover, though, because - get this. They were restoring from tape. So...

Leo: Well, hey, at least they had the tape.

Steve: Exactly. I was going to say, the good news is they had that, and so they were able to get - they will eventually have restored everybody's lost and deleted email. And their blog I thought gave us a sense for the size of the window. They said, "It's important to note that emails sent to you between 6:00 PM PST on February 27 and 2:00 PM PST on February 28 was likely not delivered to your mailbox, and the senders would have received a notification that their messages weren't delivered." So 6:00 PM on the 27th to 2:00 PM on the 28th is 20 hours. So there was a big window during which things were not happy for those 150,000 Gmail users. Again, not a security problem, but just something that is in the process of getting put back together. I don't know if any of our listeners might have been affected. I wanted to let them know what had happened and that, if they weren't yet mended fully, they probably would be once the tape was finished spinning.

Leo: I don't see anybody in our chatroom who says it happened to them. I don't think it happened to me. I guess you would only know - it would be hard to know. I guess if you got a lot of email, and you went 20 hours without email, I guess you'd know.

Steve: Yeah, and you might know later if...

Leo: Right, some appears.

Steve: Yes, if some tireless SMTP server is retrying a few times and finally gets the mail to be accepted. Okay. So, LastPass. The good news is I don't regret my recommendation or the analysis that I provided our listeners months ago, nor all of the feedback I have received from people whose lives have been dramatically improved thanks to LastPass. What was discovered by a very clever security researcher, Mike Caldwell, who's in the U.K., was that there was a way that a user's logon session could be hijacked by a malicious site. So what the user would experience would be what we always talk about here, is you go to a site which is malicious. And if you are currently logged into LastPass, as most of us are statically so we have access to all of our other sites' usernames and passwords, if you were logged in, it was possible for a malicious site to execute script, JavaScript, in the context of the LastPass domain.

So this is what's called cross-site scripting, and it's the way that malicious sites are able to get around all of the preventions, all of the barriers deliberately erected in order to keep domains from leaking information to other domains. So what was posted was that it was possible to determine that LastPass user's email address, their password reminder, and their site usage history. Which is a big information disclosure problem. The LastPass folks fixed it within three hours. Mike responsibly disclosed it, let them know. This was fixed in three hours.

It was a very clever hack. I take my hat off to him because many of the things he tried, and he's got a lot of experience doing this, didn't work because LastPass had thought through all of these possible vulnerabilities. Yet he came up with a way, and this is the problem with web-based stuff in general, he came up with a way that he could close a script that was going to be broken with a closed-script tag, and then reopen a new script,

and then inject some script himself.

Now, here's the good news behind all this. At no point was any of the encrypted data, which is what we have LastPass for, vulnerable. And it wasn't because of the architecture of LastPass, which is why I endorsed it. So in his blog posting he went a little too far, and he said that he was certain it would be possible to obtain encrypted and protected site logon username and password data. But it's not. He was wrong on that count. I've responded over on his blog, after analyzing it and talking to the LastPass guys to make sure I had my facts straight. And here's why.

What's so cool about LastPass is that they don't ever get our cryptographic key. That's why I felt so comfortable using it. What Mike found and was able to demonstrate was the fact that, by using cross-site scripting vulnerability to impersonate essentially us, he was able to get the LastPass site to reveal some information, the nonencrypted information that it had about us because LastPass back then, it's been fixed since then, but they didn't realize, there was no way for them to determine it wasn't us making the query, so they were willing to provide to us what they had. But they can't possible provide what they don't have. And what they never have is our cryptographic key. It never leaves us.

When we log into LastPass, our master password, as our users will remember from the podcast we did covering this, script running in our browser takes our username and our password and cryptographically turns that into the symmetric cipher key which we then use, that is, our browser uses to encrypt our login data. And so all that LastPass is doing is saving opaque blobs which they are unable to decrypt. So even with cross-site scripting, another session is being created that has no access to the cipher key that is in our browser, running either in the plug-in or in the regular web UI. So thanks to this architecture that LastPass established, at no point was anything more than the information that Mike showed available. And Mike is a LastPass user then and still. This hasn't put him off of it. He's still using it now.

Leo: Me, too.

Steve: As am I.

Leo: Me, too, yeah. So don't be afraid. Should we consider not putting everything, all our eggs in one basket, as you said?

Steve: Well, the advantage of LastPass...

Leo: That's the whole point.

Steve: ...is that it's one big secure happy basket. I mean, the guys were very embarrassed. They immediately fixed the problem. Mike also suggested that they enforce strict transport security, which we've talked about, STS, with browsers which understand it, like Firefox does, which they immediately implemented. And basically I think it was a good wakeup call because they have been spending their time broadening their reach and making LastPass more pervasive on more devices. And this sort of refocused them on the web side of things, rather than the third-party device side. And as a consequence, in about six hours, Joe has reimplemented, or rather implemented a very strong

technology to much better control this sort of just prophylactically, to preemptively keep other things like this from happening.

And I'll also mention that anyone using NoScript would have always been prevented from this because NoScript has built-in cross-site scripting blockage, and it was effective. Mike, in some follow-up comments in his blog, commented that users of NoScript, naturally we're trusting LastPass. We would not be trusting some other random site. But even if we were, NoScript itself would have blocked this particular exploit against us. So one more reason to use NoScript, as if anyone needed another reason to use it.

Leo: Well, that's a little scary, but I guess a happy ending makes it all okay.

Steve: Well, yes, we would like there never to be a problem. The good news is the architecture that the guys implemented prevented this exploit from going any further. What we really need to have protected was protected because they didn't have it to disclose. And they don't want it to disclose, as I talked about when we talked about it originally. The reason I liked it was that our stuff never left our local control in a nonencrypted form. And the convenience of LastPass is that it is integrated into our browser so that it's able to participate. Well, I mean, it's a difficult thing to do that securely. I mean, it's really difficult because, as we know, browsers are the main focus of today's malware. That's where all this is happening.

Leo: If you didn't use the LastPass plug-in, would this have affected you?

Steve: Yes. The only thing that would have prevented this is if you were completely logged out, meaning not logged into LastPass at all, such that it would have had to ask you for your credentials, your username and password, in order to log into it. So because it was a session hijack, this cross-site scripting vulnerability was essentially hijacking our logged-in status. The fact that our browser was logged into LastPass is what it was taking advantage of.

Leo: And for those of you who are interested, Steve did a thorough analysis of LastPass on Episode 256, 2^8 if it's easier for you to remember. And so you can go back and listen to that, and then add this to the mix because this is a new flaw.

Steve: Well, fixed immediately. I mean...

Leo: Yeah, that's what I like.

Steve: Mike let them know about it. It was instantly fixed. None of this Microsoft wait for next Tuesday or a month and a half or so forth. I mean, this thing was addressed instantly. And I think it ends up further increasing the security because these guys realize they have a huge stake in maintaining everyone's confidence in their service. And I continue to feel, as even does Mike, who's still using it himself, that they got this right.

Leo: Bravo, LastPass.

Steve: I did pick up a little note from a listener who wondered if I was ever going to review the Kindle 3. And I don't want to take much time on that, but I'll just say that I love it for what it is. But recently I've been reading some textbooks, more O'Reilly stuff, programming stuff, and I switched to the DX because code was wrapping in the little screen of the Kindle 3. And then, sort of out of curiosity, I tried reading it over on my iPad. And the iPad just blows it away. The ease of use, the higher contrast that you're always going to get from an active backlit screen, means that things are just sharper. And the various fonts show up better.

I love the Kindle 3 for fiction book reading, for textual book reading, for stuff that isn't graphics, that isn't diagrams, really for paperback sort of books. Nothing beats it because it's great for that. You can hold it in one hand. The iPad really is too heavy to hold in one hand. You really need to prop it somehow. But if you're sitting down somewhere, and you have a lap, you can sort of prop it up in your lap and just flick the screen forward with your finger. So I'm excited about tomorrow's iPad 2 announcement. And really...

Leo: Well, that's why the speculation is it'll be lighter and thinner, exactly for that reason.

Steve: Yes, and I'm really excited about iPad 3, which is supposed to give us that same quad resolution retina display in a pad form factor that we now have in the new iPhone 4.

Leo: iPad 3, that's not till next year; right? I mean...

Steve: Maybe later this year. The news was that they're kind of - the problem is they can't get production levels up. But it means they're trying to produce them.

Leo: That sure would be great. I mean, that's a beautiful screen.

Steve: That's all I want. If I could get that retina screen on a pad-size device, wow. We heard from a listener, a Security Now! listener, Anthony Pitcher, who wrote: SpinRite Saves My Raid Zero. He said, "Hi, Steve. Long-time listener of Security Now!. Enjoy the podcast immensely and appreciate the work yourself and Leo do for the community to inform us all about being more secure. I purchased SpinRite two years ago to support the podcast." Wow, thank you, Anthony. "I never needed to use SpinRite to fix a drive until today. On my Windows 7 x64 Raid Zero installation, Windows began to have some weird behavior, random freezes and the like. I also noted that the Intel matrix storage manager began saying a drive was being disconnected from port 1 and reconnected, disconnected and reconnected. However, Windows was still limping along. Thanks to Leo and his continuous mention of backing up, I have a local and offsite backup at my office."

Leo: Good man.

Steve: "So I wasn't concerned if Windows just fell over. However, I was more interested to know if there was something wrong. I ran SpinRite on Level 2, booted from a USB floppy drive." A USB floppy drive, that's what it says.

Leo: A USB floppy drive. Well, yeah, that's probably the only kind there is these days.

Steve: Yeah, "...booted from a USB floppy drive, and it began chugging along. I have two 500GB Seagate drives, so the scan took one hour and 30 minutes on the first drive, which reported no problems. The second drive was about 80 percent done, and I was thinking by this time SpinRite would probably find no problems there, either. Then the DynaStat monitor kicked in. It ran for about two hours, and then completed the rest of the drive without hesitation. Windows booted normally and no longer had the lagged experience I was having before. And of course the Intel Matrix Storage Manager software no longer complained about a drive being disconnected and reconnected.

"Thank you, Steve, for such a superior product with so many boot options available. Your software saved me from having to reinstall Windows and set all the settings I've gotten used to again from scratch. I know this story isn't saving lives or someone's job, just a fellow fan who really enjoys your programs and hopes to be one of the first customers of CryptoLink. Take care. Anthony Pitcher." Thank you, Anthony.

Leo: What is the status with CryptoLink? I don't mean to beat you up on this.

Steve: Put me on the spot?

Leo: Yeah. Just an update.

Steve: I'm very nervous still about...

Leo: Oh, because of COICA or whatever it's called.

Steve: Yeah, about what the FBI is going to try to do. As I mentioned last week, it seems like what they're going to ask for is - and they seem to be backing down a little bit from some of their earlier statements, which I think is good - they seem to be wanting sites and services that could comply with a wiretap order to be forced to. So, for example, Google and Facebook, they're endpoint services that have decrypted information because they're at the other end of their customers' SSL connections. So when the federal government goes to Facebook and says, we need all of the communications of this person, Facebook says, well, our technology isn't set up to do that. It's not that they couldn't. It's that they haven't had a reason to before, so they don't have the code in there to do that.

Leo: And I bet you they intentionally don't add that. I mean, they don't want that responsibility.

Steve: Right. And so what the government would like would be legislation which makes it mandatory for something like Facebook or Google to implement the technology that would allow them to respond to a wiretap order like that. That's what I'm thinking. Now, the problem is, in the past they've mentioned Skype by name. Skype is non-U.S. based, probably makes the U.S. government a little nervous. And as we know, it's point-to-point crypto, just like a VPN, just like CryptoLink would be. So what's unclear is what they're going to ask for along those lines.

I'm going to keep myself busy. I actually have a project I haven't talked about yet. I've got a couple of things I have to wrap up first, some very cool technology that's been running on the server for years, we've talked about it before, third-party cookie stuff, that I need to make - that I just need to finish documenting. All the technology is in place. But I have a plan for something that I'm going to do relatively quickly that I think a lot of listeners will find extremely interesting. So I'll have more to talk about soon.

Leo: Okay. Fair enough. All right, Steve. I've got questions. I presume, since you gave me the questions, you've got answers.

Steve: And we got some good dialogue from our customers - from our customers. From our listeners, too.

Leo: Good. Hey, they're our customers. This is Question #1 from Charles G., Pittsburgh. I'm presuming Pittsburgh, Pennsylvania. He wonders about a two-factor authentication, Intel style. He says: Am I missing something? This is something that you must have talked about when I was gone. Am I missing something? So you'll have to fill me in on what this Intel thing is. But if the six-digit number can be generated on demand for authentication - I guess I have one of those VeriSign cards that does that - what's to prevent malware from being able to do the same thing? Yes, it stops crooks from using other machines. But if your machine is compromised, isn't it worse than having a separate dongle? Oh, I get it. Intel's building this into the machine.

Steve: Yes. It's in the Sandy Bridge chipset.

Leo: Interesting.

Steve: They built in exactly the same technology, this one-time password authentication. And I have to say I love our listeners because this was one of the two most popular topics when I dumped the mailbag for this week's podcast. The other one was still Bitcoin. Everyone just wants to keep talking about Bitcoin. That just really catalyzed everyone's imagination. But so many of our listeners said, whoa, whoa, whoa, whoa, wait a minute. If this second-factor hardware is built into the computer, then what's to prevent malware from accessing it?

Leo: That's a good point.

Steve: Oh, it's a great point. And it is absolutely the point. So what we know we can say

is that what this is doing is authenticating the machine. It does not have the advantage of a freestanding credit card or football, but neither does it have the cost. That is, it's just going to be there. So I'm also very interested in how Intel will use this, that is, how it will surface, how it will be protected, what measures there will be to prevent malware from accessing it. But the point is that that isn't the threat model.

What Intel has hoped to do - and this is inexpensive, it's not like this is some big huge project or anything. I mean, this is a trivial little algorithm. It probably took a minuscule portion of one of their chips to do this. So it was sort of one of these things that, well, we'll just throw it in because we can and because it's simple and another bullet point on our checklist of features that we've got for our chipset. So I didn't mean to make it a big deal. What it does allow, though, is for that hardware to be uniquely identified. So while it's true that malware running in the machine - I would be the last person to say there's no way malware could get access to it. But the point is there's no way that someone outside of the machine could know what's happening inside the machine.

So I'm sure it was intended to prevent, for example, keystroke logging, as an example, from being able to be captured. Or somebody doing a man in the middle, sniffing your traffic in an open WiFi, who sees you logging in and then tries to log in again, that's a perfect example of what this would defeat. So, yes, it wouldn't defeat malware running in your machine, but it provides authentication for the machine itself to entities that don't have access to the machine. And there's a lot of those.

So I completely agree that it doesn't have the same strength as something that software can't access because by definition some software has to be able to access it. But that isn't what it's trying to prevent. And I'll also mention that there have been exploits, even against one-time passwords because, if malware is in our machine, it can intercept us entering our one-time password and ride on that session. So it's still important that we keep our machines clean. And really the one-time password concept is much more meant for external protection than it is for protection against things that have already crawled into our machine and set up housekeeping there.

Leo: It's just the same story as with the Thunderbolt issue. If somebody has physical access to your machine, you're in bad shape. I mean, there's all sorts of things they can do. But presuming they don't have physical access, this is a great solution.

Steve: Yeah, and it's free. It'll just be there. It'll be, like, hey, why not use it? It's part of it.

Leo: It's like TPM; right? I mean, it's just another kind of thing in the chip. Mike Norris - not Chuck Norris, Mike Norris, his brother in Louisville, Kentucky, I don't know if it's his brother - wants to poke a hole. It's good Chuck's not doing that because he'd just kick it right in there. Steve, I've installed Bitcoin, and it's cranking away. I have a question about the comment to set the TCP port forwarding to 8333 so that you can create more connections. I'm having trouble doing this. What is the procedure to set this up safely? I'm running Windows 7. Mike. He wants to know the port forward; right?

Steve: Yes, exactly. Now, one thing scary about Windows is where we think we've got a firewall which is protecting us, when you look at all the exceptions that have been made

through the firewall for incoming traffic, it's just Swiss cheese. It's why it's still really necessary to be behind a NAT router if you want anything like protection. In fact, I know that Mike is behind a NAT router, otherwise he wouldn't be having a problem. You can, in all versions of Windows that have had the personal firewall, and I did it in Windows 7 just to look at what the process was, you can bring up the advanced firewall configuration screen, and then you will be terrified as you scroll down through all the applications that have said to Windows, oh, I'd like to receive incoming traffic, please.

Leo: No problem, no problem.

Steve: Yeah. And there are, sure enough, two entries in that list for Bitcoin. I think they're alphabetically sorted. And so they were at the top, as a matter of fact. And one is to allow any TCP connection from any IP on the outside to any IP on the inside, from any port on the outside to any port on the inside. And the second is to allow any UDP traffic, similarly, from anywhere to anywhere, on any port to any port. So it's just wide open. Basically Bitcoin negotiated with Windows 7 and said, I need to listen to everybody. And Windows 7 said, oh, not a problem.

Leo: Is that UPnP that's doing that?

Steve: No. That would be used if your computer were talking to your NAT router.

Leo: Oh, okay.

Steve: So this is just done locally in the machine.

Leo: It's done in Windows software. I get it, I get it.

Steve: Right. So what it means is that we really, yes, kind of there's a firewall, like maybe there's some traffic that it would block. I'm not really sure.

Leo: Well, isn't there a way to turn this behavior off? Can we just say don't do that automatically?

Steve: Oh, yeah. You can override it. There's buttons and switches and settings and everything. But that's really not the problem. The good news is, we're behind a NAT router, so that's Mike's problem, is that Bitcoin has taken responsibility, negotiating with the Windows firewall, to open everything up so it can hear anything happening. The problem is it's still deaf because the NAT router is not allowing stuff through. So the amazing site for helping people with port forwarding is just called portforward.com. And it is - I didn't even know there were this many routers in the world. You go to portforward.com and start scrolling. I don't think it's possible not to find your model, make and model of router on their list. And they will show you how to do this.

Essentially, you need to change some configuration settings in your router so that port

8333, which is the port Bitcoin uses to run its whole peer-to-peer network, all of the traffic is running over port 8333. You want to allow that to be sent into the IP address of the Windows 7 machine where you're running Bitcoin, and then that will allow it to participate in the network. Although I should also say I'm running Bitcoin successfully behind NAT without port forwarding. The connection count is dramatically limited. I think I have eight connections which my Bitcoin machine has been able to make. Out of curiosity I ran it on a different machine and did allow incoming traffic, and it had set up 60, six zero, connections.

Leo: Oh, that's a big difference, yeah.

Steve: So a huge improvement.

Leo: And more is better; right?

Steve: Well, I don't know. Because it's the eight-connection machine that won the Bitcoin prize, and I got 50 bitcoins.

Leo: Well, that's just luck. That's not...

Steve: Yeah, so I'm not really clear on why more connectivity is necessary. They seem to be promoting it, so I would say do it. I don't see any reason not to.

Leo: Do you get more data sets if you have more connections? Not clear. Unclear.

Steve: It's not clear. I got eight connections, and that allowed me to win coinage. So I'm on the network, I think I've got something like 3,000 confirmations now have come in that my computer did solve the puzzle correctly. So we're in good shape. But portforward.com for anyone's port-forwarding needs. They really - they've got it well covered.

Leo: Bravo. I have that Bitcoin server, I've just got to turn it back on. I turned it off because I thought, well, what the heck. What am I doing to do with it? Maybe Bryan L. Gay in Atlanta, Georgia has a comment. He says: Bitcoin Fail. Well, I installed the Bitcoin client on the only Windows machine I have, one built for gaming. So, yeah, that should be fast; right? And the rest of my machines are either servers or work machines and laptops. Unfortunately, Bitcoin chose the wrong port to try to operate on. 8333? Really? This is VMware's port. Ooh, I didn't know that. I run a VMware server on all my machines. So Bitcoin won't even attempt to run on any of them, citing its inability to bind to the port, assuming it must already be running. Now I'm looking for a way to change the port and get off 8333. You know, there are only 65,000 total ports, and there's going to be some collisions from time to time.

Steve: Yeah, I was disappointed to see that there isn't an option for changing the port from Bitcoin. But on the other hand, it probably can't be done. That is, it's probably not

possible, or just maybe their software's not flexible enough, for one machine in their big peer-to-peer network of clients that are all running on port 8333, for some random guy to be on 62942 or whatever. So there is a collision with VMware server. The good news is that Bitcoin fully supports running over a socks proxy. And of course proxying was the topic of last week's podcast.

Leo: Bitcoin, by the way, I should mention, the topic of two weeks hence. So if you go back in time...

Steve: Yes.

Leo: Or three weeks hence, you can get all the information we're talking about.

Steve: On Bitcoin, yes. And Bitcoin does explicitly support TOR. In their FAQ they talk about someone who wants additional anonymity, for example, of the IP that they're connecting from, can use TOR. So you could run TOR, the TOR system in that machine, have Bitcoin connect to TOR, which it does by not using port 8333, and then TOR would tunnel out to the TOR network in order to get out. Or you could set up a local socks proxy, set the local proxy to run - if you use a socks proxy, then you're not going to be running over port 8333 locally. You'd be tunneling that out through the socks proxy and thereby avoid the collision. And there are lots of FAQ pages over on Bitcoin.org to explain how all that's done.

Leo: Okay. People really got - their imaginations were captured by Bitcoin.

Steve: Oh, I'm telling you, this last week it hasn't died down any. Half of the questions people submitted were interesting things about Bitcoin. It just captured people's imaginations.

Leo: Yeah, it did. Our banker is sitting here. Ron's from Exchange Bank, and he's going, what? What's wrong with the old American greenback, I ask you? Andrew in Northern Ireland wonders whether the server will protect us. Steve, I recently started working for a small company with a single server and around 15 client machines. Having listened to Security Now! for several years, I was a bit startled to see that most of the client machines run XP SP2 with few or no updates applied. I think this is actually pretty common.

When it comes to my home machines, I have a mild case of OCD regarding keeping everything up to date with the latest patches and fixes, so seeing this got me asking my employer some questions about their security practices. They have been told that because they are behind a server, a Windows-based server, I don't know what version, they don't need to update the client machines at all. Maybe they don't get email. Maybe they just don't give them email. That could be. Maybe they're not allowed to surf.

Now, this doesn't seem right to me, and I'm sure there must be some example of how this could provide a security hole, but I can't think of any good ones. If all traffic

to and from the Internet goes through the server, does that automatically protect the client machines? If a virus or a trojan were to be installed locally, maybe via say a USB pen brought from home or a downloaded malicious PDF file, what damage could it do if it can't get through the server to the Internet? Of course, this assumes the server is constantly up to date with Microsoft patches, virus definitions, et cetera. Of course I doubt this also takes place. Am I wrong? Will the almighty server protect us all? Or am I right to advise updating some machines? Love the show. Regards, Andrew.

I guess you could extend this to say, what if they're running, say, an Astaro Security Gateway or some sort of security gateway. Do they have to update the machines?

Steve: Yeah, they really do. First of all, it's not clear from what Andrew said what this Windows server is. I mean, Microsoft does have an ISA server, which is a firewall border protection server that can do filtering and things. But we assume that these client machines have access to the Internet. It's hard to imagine that they wouldn't. Maybe they're running some client server application software to do whatever their business does? I mean, when he talked about they're behind a Windows server, it's like, okay, well, I'm not really sure what that means.

But if these machines have access to the Internet, they are absolutely vulnerable. I mean, all of the exploits that we've talked about for the last several years that have involved browser client vulnerabilities, I can't see how any of these things that Microsoft has been fixing constantly, that we're sitting here waiting for Microsoft to fix these things. If these machines haven't been updated since SP2, then they've got years of missing updates for exploits that are going on, known problems with, for example, PDF files, which someone could email to them and open. And these are - sometimes antivirus software catches the stuff. Other times it never catches up before Microsoft patches it, or there aren't any fixes for these problems through various AV tools.

So I'd have to say, I mean, I don't want to get Andrew in trouble with his boss or challenging whoever their security nonprovider is, but this seems crazy to me. So, yeah, I just can't see a safe way of running machines that are out of date, that have contact with the Internet, because that's where the problem comes from.

Leo: Well, I mean, yeah, they're going to get infected. Period. Right? So it's just a question of - and they don't have to communicate to the outside world to be dangerous.

Steve: I was just going to say. And, as he points out, if something did come in on a USB stick, the most recent viruses, malware, and trojans are using local LAN connectivity to spread throughout an organization.

Leo: So you can assume everything behind your server is now infected. I think this is an example of what CNN - remember CNN got bit by one of those worms, Conficker or something. And it's exactly what happened. Somebody brought it in from the outside. It infected that computer and then spread through the network to get all the computers. And I'm sure CNN had a server between it and the outside world, and they probably had routers, too. Doesn't matter. Still get infected.

Let's talk about proxies. We talked about that last week. Steven Meyer in Switzerland has a great comment about proxy dangers: I'm new to your podcast, really like it. Welcome, Steve Meyer, good to have you. Thanks for the high quality of content and sound. Sound's important to us. We want to sound good. When you were talking about proxies you forgot to mention about the sniffing risk of a proxy. Any password sent through the proxy can be listened to; and, if the proxy is malicious, it could impersonate you even when using SSL. Thanks for the nice podcast. Steven. Now, we've talked about this many, many times, maybe not on that particular episode, but certainly we have talked about this.

Steve: We have, and he is so right that I really, I got so carried away with the technology of proxies, which is really what I was trying to cover last week, that I don't think I did justice to the huge risk of using them. And many people wrote to say, uh, Steve, you forgot to mention that that's really unsafe. And it's like, oh, you're right, I did forget to mention that. And it really is unsafe because he's right, and many of our other listeners who wrote in to remind me are right, and I should have spent more time on that because, think about it.

I mean, it absolutely is the case that, when you are surfing through a proxy, even if you're using an SSL session between you and the proxy, you're then decrypting that at the proxy end. The proxy sees everything going on. So I guess what I should have and I didn't say is treat it the same way you would surfing in a library, on a library computer, which I was covering earlier. I mean, it is just - it is that bad. It is something you would use sort of in the context that I was painting it, for if you can't get out any other way - although I did use Facebook as an example, and that's a bad one because you have to log into your Facebook site. So any proxy could see you do that, and game over at that point.

So I'm glad Steven brought it up, and I really should have spent more time on it last week. So I wanted to do so, to absolutely make it clear that you have to - the proxy has to be trustworthy, and probably none of them are. So what that means is you have to treat it as an absolutely untrusted channel and only use it as a means for having access you wouldn't otherwise be able to have, but put nothing through it that is important to you because it's absolutely, I mean, we don't know that they're sniffing things. I think that there are trustworthy, I know that there are trustworthy proxies. I forgot to mention famous Anonymizer.com last week, a really good - they're not free, they're commercial, but I would trust them. And there are other commercial ones. So they're not all shady. But you just don't know who they are when you're using HideMyAss.com.

Leo: Yeah, who are they? That's a good point. Question 6, Jesse in Minneapolis wonders about C language character arrays. What other show, ladies and gentlemen, what other show would you hear the broad range of topics that you hear on this show? I'm in my second semester of the CSCI program at the University of Minnesota. We are currently covering the String class in Java. My professor mentioned that, in C, all strings are handled as a character array ending in zero, a null terminator. That's how you know the string ends. Java probably is like Pascal where it begins with a character count, a string length. Is that right? I don't know. My question is, if that's the case, doesn't that make any application in C vulnerable to buffer overflow attacks?

Steve: Bingo.

Leo: I can answer that one. Yes.

Steve: You were laughing by the time you got to the end of it.

Leo: As a matter of fact. Well, we've talked about using, for instance, there are a couple library routines in C for string copying. One is strcpy, and one is strncpy. And all good programmers now know you use string ncpy.

Steve: Exactly. The way to phrase this, Jesse, is because C uses null terminator strings, the way you copy a string is you copy the characters from one place to the other until you hit the null terminator. So it's hitting the end of the string, that zero, the null character, is what stops the copy operation. But that is the malware author's absolute dream because this means that they can get code on the stack or in your system to copy whatever data they want, anywhere they want it to go, and it will just keep copying away until it hits a null character.

So it is absolutely the case that, unfortunately, it's very convenient to use null terminator strings, but they are incredibly dangerous. And exactly as Leo says, a strcpy "n" variant allows you to specify the target buffer size so that the copy will terminate at either the end of the source string being copied or the end of the target buffer being reached, whichever one occurs first. And that makes things way safer. As we know, it doesn't solve all the problems. But it certainly makes things harder to exploit.

Leo: That's probably, at least it was for a long time, the single most common exploit was exploiting these string overflows.

Steve: Yeah, actually, cross-site scripting has surpassed...

Leo: Buffer overflows, really?

Steve: Yeah.

Leo: Welcome to the new world of the web. I'm surprised, though, I would imagine, Jesse, that they're going to cover that in your class. They sure ought to because nowadays when you're teaching computer science and programming, you've got to always teach about good security methodology.

Steve: I think that's what's so nice is that we're to the point now where security can no longer be an afterthought. It's in the consciousness of everyone, both users and authors. It's now a factor. So we're maybe beginning to leave a little bit of the Wild West days. I don't think we see evidence of it yet out on the frontier. But I think we will in another decade.

Leo: Yeah. Kristofer Thurston, Plano, Texas, wants to tell us about another kind of proxy: I've been a network administrator in public and private education (K-12) for over 10 years, and proxy use/abuse has been a nemesis for me most of the time. Of course, if you think about it, especially in a high school, where you're using some sort of filtering, that kids are just going to use a proxy server to get around that, if they're at all sophisticated. Just like in China and Tunisia. Don't get me wrong, I'm a full believer in the right to privacy, the Internet, and speech. But in my line of work, the distraction of the Internet can drastically reduce student performance while simultaneously increasing the level of aggravation in our teachers.

In order to make the Internet a resource instead of a distraction and also because of CIPA, which is the Children's Internet Protection Act, we're forced to filter traffic during school hours. We're currently using a filter that leverages URL filtering in combination with deep packet inspection to prevent access to some of the less illustrious Internet content. The DPI portion uses matching rules based on Snort packet signatures. That's pretty sophisticated. This solution does a fantastic job of eliminating proxy type traffic as well as instant messaging, and as a result is a great supplement to the URL filter.

However, as a result, our students have had to use even more devious means - I have an opinion on this, which I'll share later. But what they've found are two products called UltraSurf and Freegate. These proxy services work by creating a local proxy server in the student's machine, and pointing their browser to this local proxy server. The proxy server then negotiates an SSL connection to the network of servers on the public Internet which then proxy the web requests. Of course, as soon as it's SSL, you're dead because you can't inspect it.

The distributed network, like BitTorrent, prevents blocking of specific IP addresses. This is effective because SSL encryption completely masks the packet payload, so that deep packet inspection is no longer useful. Also, because more and more sites are "going dark" and switching to SSL, like Facebook and Gmail, DPI is further more ineffective. As a result, we are in the process of using a filter similar to those you have discussed previously, that require the installation of a CA certificate on all clients so that SSL traffic can be decrypted inside the box, authorized man-in-the-middle. When we do this, we will, of course, do so with full disclosure and warn against using the school network for truly secure purposes like banking. In other words, the school is now saying you're going to go through our SSL, and we're going to look at everything you're doing.

Sorry for the length, but I thought you'd like to know about these two products specifically. By the way, from what I understand, UltraSurf was created by the CIA to subvert government Internet filters in China, and Freegate's a derivative of that.

Steve: So I actually mentioned UltraSurf and Freegate last week. They were the impetus, as I said, for me thinking we had never talked about proxies. And I didn't focus on them because I did poke at one, and I got a response back that my IP was not in China, and their service was only limited to - for people in China. However, they both allow - they go further and do something I didn't do because it really wasn't germane to the podcast. But clearly Kristofer's kids, high school kids are doing this. And that is they have the option of installing Firefox plug-ins. And that's where they establish their local proxy server that uses SSL out to an agile IP network out on the public Internet in order to create the connection.

So I just wanted absolutely to bring this to people's attention. Again, I don't intend to be promoting the bypassing of established proxies and filters and things. But our listeners are savvy and responsible and may have a need. And both UltraSurf and Freegate, with the caution that the CIA seems to be involved somehow, do look like ways of bypassing the use of what would otherwise be frighteningly unsafe proxies by switching to something that was deliberately designed for free speech and communication. And what were you going to say about...

Leo: Well, I'm on the board of trustees at my kid's school. I'm kind of their tech advisor in high school. And we go back and forth on whether to filter or not filter. And I also talk to educators a lot. And of course there may be legal requirements for them to filter; and, if that's the case, of course you have to do it. But barring those, philosophically, my point of view is, look, every kid's got a cell phone now, and their own 3G network. You can't stop it.

Steve: And they go home after school, where they've got...

Leo: You can't stop it.

Steve: Exactly.

Leo: So better to use this as a teaching moment and not block, but teach the kids the right way to use the Internet, teach the kids how not to be distracted. They're going to face this at some point anyway. Might as well use - that's what you're doing now, isn't it, in school? You're teaching them ways to cope with life. And so I think it's an artificial constraint to say, well, no Internet for you. First of all, it's not going to work. Second of all, then they're not learning key skills about how to deal with the Internet, how to deal with distraction, how to appropriately use the Internet.

Steve: They're still distracted trying to get around the filters.

Leo: They're spending more time doing that. And every kid who has an iPhone or any smart phone can surf and go anywhere they want anyway on their 3G network, and you can't stop them, neener neener neener.

Steve: Very good point, Leo. They've got their own connection out to the cellular network.

Leo: It's hopeless. So your teachers have to learn how to deal with it. They say, "Shh, put your phone away and shut your laptop." Your teachers also have to teach kids how to deal with the Internet. I mean, this is a very - what more important thing can a kid learn nowadays, frankly? Anyway. I'm sorry. I'll get off my soapbox. Question 8 from Jack Daniel. He's the man at Astaro, another one of our great sponsors. He's got a little opinion on proxies. He says:

As always, thanks to you, Leo, and occasionally Tom for the great shows. Tom did a great job filling in. The web re-writing, browser-as-client kind of proxies you mentioned in last week's episode have a few problems. Jack is all about security, so this is his thing. First is the inherent domain obfuscation. This breaks what little cross-domain protection we have left as all content looks like it's coming to the browser from the same domain.

Steve: True.

Leo: So it can't discriminate.

Steve: True.

Leo: Second, and a bigger issue for many, these sites are free. And some, perhaps most, are supported by unsavory practices like serving spyware and malware. You don't know. How could you know?

Steve: Right.

Leo: This type of proxy is a big problem for schools, both as they try to keep the students focused on school work instead of Facebook or worse, and because of the malware issues they bring. Astaro systems have tools to address these issues, but I don't want to do an advertisement, he says. But good security practices are of course part of this. I haven't looked lately, but I would also worry about those that support HTTPS sites. Are they proxying that traffic by performing man-in-the-middle proxies? I would look closely at the certificates. Finally, if they're just SSL/TLS wrapping the HTTPS, the tunnel is prone to the infamous TCP over TCP tunnel collapse once retransmissions begin. But complaining about poor performance on a free proxy is probably pointless.

Also you've talked a lot about two-factor authentication over the years. Have you ever looked at WiKID Systems? They have two two-factor authentication systems, one open source and one commercial, and they do some pretty cool things and support a myriad of devices. I don't know this one, wikidsystems.com.

Steve: I looked and only saw the commercial side, so I want to find out what the open source thing is because they look like they've got a broad range of support, and I would love to find a free source for good two-factor authentication.

Leo: You bet. Finally, updates on free events: HacKid has a few more events on the horizon, no dates yet, but hackid.org/wiki lists several in the planning stages, including one in the Bay area near us. Well, we'll check that out. I'd love to cover that. He says: Another event I've been involved in and sponsored by Astaro is Security BSides. These are a series of free InfoSec events held around the world, sometimes adjacent to large events, sometimes standalone. The focus is high-quality

content in a relaxed and conversational format. Great for our audience. I think Leo will be dropping by to see us in Austin next week. I will. Registration is full for that one, for South by Southwest. We'll be down there covering it. Unfortunately that one's full. We may have space for a few walk-ins, but there are many more coming up all around the world. And you could find out more about that at securitybsides.org. That's the wiki.

No need to mention my name, says Jack. I'm sure folks are sick of hearing from me. I just wanted to drop you a line with a few things I thought would be interesting. Well, of course we love hearing from you, Jack. Thank you.

Steve: So, great points from somebody who has been over on the front lines of this proxying stuff. Certainly the points he makes about the fact that domains are collapsed into one is very good because it is only by keeping the domains separate that our scripts are able to be controlled. So losing that really does dramatically weaken script security as another negative to that kind of proxying. And that was the last one of the multiple type of proxies I talked about where you go to a site, and you enter the URL you want to visit into that site, and what you'll see then is all the URLs and all the links of the page that come back refer to the proxy server, not to yours. What that means is that scripts are confused and believing that they are hosted by the proxy server, not by their actual origin server. And that can be a problem.

The SSL thing is not such a problem. I think that Jack was probably assuming there was more SSL going on than there is. For example, in the case of, we talked about it last week, HideMyAss.com, you create an SSL connection between you and them only, and then they take that apart, and then you have a non-SSL connection out to the remote site. So you probably aren't trying to tunnel SSL within SSL. And so I don't think that's a problem. And then the other information I just wanted to share with our listeners.

Leo: Cool. Want a YubiKey story? Brett Moffett in Adelaide, South Australia uses his YubiKey to log onto PayPal: Long-time listener, first time emailer. Ever since hearing about YubiKey on Security Now! I've been a convert. The power of a one-time password and the ability to store a very long random password all in a very small device is fantastic. I just wish more sites would support it. I do, too. Well, it looks like instead of waiting for sites to accept YubiKey, Yubico has brought YubiKey to them. I noticed in a recent Yubico newsletter there is now an option of buying a YubiKey with Symantec's VIP installed in the first memory slot of the key. This can then be associated, because Symantec is supported by sites like PayPal, and you can use your YubiKey to access these sites. The second slot can be programmed to use Yubico's OTP, OATH, or even a static password. Unfortunately, it can't be retro fitted to existing YubiKeys, but buying it with it built in costs no more than a regular key.

Keep up the great work with Security Now!. I can't wait for your VPN solution if it ever gets off the ground. Isn't VIP VeriSign? Is it Symantec?

Steve: Symantec bought it.

Leo: Oh, I didn't know that.

Steve: Yeah, they bought the VIP division from VeriSign. So it's now Symantec. And I had coffee last Thursday morning with Stina Ehrensvrd, our founder and actually inventor of the YubiKey concept. And she mentioned this to me, which I wasn't aware of, so I was glad to see Brett bringing it up and reminding me to let everyone know that among all of the other types of tokens which the Symantec VIP service offers, YubiKey is now one of them, if you get one of their Symantec VIP YubiKeys. So exactly like the credit card that we've talked about with eInk, or the little LCD-based football, or the app in our phone - which frankly, that really solves the problem for me...

Leo: That's the way I want it.

Steve: Exactly. So essentially this is the same incremental algorithm that the static credit card approach uses where it just sequentially generates the next key. The YubiKey has that built in. And so that's one of the authentication options that it offers. So you stick it into a USB slot, and when prompted to, just tap the button. And I guess, if you had some application where you needed to authenticate often, then that could be pretty handy. You'd just sort of leave it in the USB slot. And every time you're being asked to reauthenticate, you just tap the little dot on the YubiKey ,and it would send in the next code to say, yep, it's still me, I'm still here.

Leo: I'm still here.

Steve: And that would be easier than getting your phone out again and bringing up the app and keying in what the code is now being displayed. So I can see a positive application for that.

Leo: You bet. I didn't realize the YubiKey was so sophisticated. I mean, it's really a little computer there.

Steve: They're taking it in a lot of directions. And in fact, I'm under embargo on a very cool, completely new thing that they've done, probably till the end of March. Stina said they wouldn't be able to talk about it yet. It doesn't have huge end-user significance. It's more something for the enterprise. But a very cool piece of technology that I'll be able to talk about in about a month.

Leo: Great. Our final question comes from an Eagle Scout. Actually, I don't know if it's a question. It's more like information. Lance writes: Steve, you've talked the last couple of weeks about Bitcoin. After your podcast I decided to check it out. After running two computers with Bitcoin for a day I saw nothing. Well, it takes time. Be patient. So I decided to look into what others were doing to compete with these GPU Bitcoin farms. I found that pooled mining is a great way to combat this. I joined the mining effort at mining.bitcoin.cz, which is I think the Czech Republic. And after two days I had generated [fanfare] one bitcoin.

However, my computer had, just as yours, cranked out the hot air, and I could hear the liquid constantly being pushed through. In other words, it was working. So I decided it wasn't worth wearing out my two computers for 50 cents a day. Now both

computers that I was using were quad cores, one at 3.4GHz and one at 2.8GHz. So I was on the higher end of CPUs. I can't imagine how long it would take an older PC, even in pooled mining, to generate one bitcoin. I still wanted to get in on these bitcoins, but it was pretty apparent that I was either going to wear out my PCs doing it, or I had to invest money into a GPU or just buying coins. But after searching around I found actually most people using bitcoins do not farm for them. They use sites that accept bitcoins to sell items and make bitcoins from those sales. In other words, it's really become an economy. So they have sites...

Steve: Yes, there is really an economy.

Leo: There's sites like eBay and Amazon that you can bid or pay in bitcoins. For those looking to get into the Bitcoin game I would highly suggest trying to sell items on these trading sites that accept bitcoins instead of using eBay - then you can build up, accumulate some bitcoin - as I personally found it a lot easier than farming bitcoins. And this way you will still make bitcoins, but you won't have to wear out your computer, waste bandwidth, and run up your electric bill doing it. You know, I've got to set up Bitcoin donations. Thanks, Steve, Leo and Tom. I just, I feel like I like American money better. For now. Thanks, Leo, Steve, and Tom for the excellent podcasts. You guys are my secret weapon at work to keeping my job and getting raises. Lance, Eagle Scout.

Steve: So I just wanted to, again, there's been so much interest in Bitcoin that I think people liked it because it was a little wacky and off topic from what we normally do, yet it still has serious crypto side because of the architecture that was developed, and the fact that...

Leo: It's kind of hacker-y, too. It's kind of out of the mainstream kind of anarchy.

Steve: Anarchists, yes. And so this farming, I did mention farming in the podcast, but I wanted, thanks to Lance, to bring it up again because the idea is that it sort of flips the model around. Rather than one machine cranking away in isolation all by itself in the corner, pumping out heat, and not very often pumping out bitcoins, like they're estimating you would win one puzzle once a year and get 50, I just lucked into it and won one after less than a week. But I haven't done anything since. I've still got those machines running, and nothing. The alternative is to change the model around and join a mining pool where any machines that you can allocate to it, you're all working together. And if any one of those machines in the pool solves the puzzle, then you all share equally in the bitcoinage.

Leo: Doesn't seem like that would give you any bigger advantage, though, because you have to split it.

Steve: Yeah, but he got one bitcoin in two days. So it's like instead of getting 50 at once, you get one at a time, but at least...

Leo: It's like creating a pool for the lottery, though. I mean, I...

Steve: It's very much the same, yes. But at least you're not looking at 0.00.

Leo: Right. I guess that's frustrating for people. It's kind of a weird system that way. You just, you know.

Steve: The whole thing's wacky. But...

Leo: So there's no expectation that you can generate any particular amount of bitcoin over any particular amount of time. It's just random.

Steve: Right, it's purely statistics. And so because I solved the puzzle, someone else who'd been working for a year, who came in second, didn't solve it.

Leo: Poor guy.

Steve: But I was working for less than a week and got 50 coins. It's like, uh, sorry about that. I'm not unplugging mine, though.

Leo: Well, and what have you got? I think people would do better just buying a copy of SpinRite from Steve.

Steve: It's just pure curiosity. It's just...

Leo: I had the same reaction when I first heard about it, ran the server for a while, same thing. It's very interesting.

Steve: On these cold days you could use a little extra heat.

Leo: Yeah, why not. God knows I've got enough computers running, idling, sitting there. Steve is at GRC.com. That's his website. He actually tweets, too, if you want to follow him on Twitter. It's @SGgrc. And his corporate account is @GibsonResearch. GRC.com is the place to go for 16KB versions of the show, transcripts, show notes. And there are 290 shows on there, so you can go back in time and look at them all. He's got them all there, GRC.com. While you're there, pick up a copy of SpinRite. Every hard drive needs SpinRite. And a lot of free stuff there, too, some great free security apps and more, GRC.com.

And Steve, we'll be back at our regular time next week. Thank you for allowing us to shift you with MacBreak Weekly because tomorrow of course the iPad

announcement. But normally we do record Wednesdays at 2:00 p.m. Eastern, 11:00 a.m. Pacific at live.twit.tv.

Steve: And so your trip next week doesn't interfere with our podcast?

Leo: I don't leave for Austin till Friday afternoon.

Steve: Cool.

Leo: Yup.

Steve: Okay, my friend.

Leo: See you next week.

Steve: Talk to you then. Thanks.