Listener Feedback #79

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-222.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-222-lq.mp3

Leo Laporte: This is Security Now!, Episode 222 for November 12, 2009: Your questions, Steve's answers #79.

It's time for Security Now!, the show that covers all things, you know, safety on the Internet and privacy and security. And he's the guy to do it, just the guy you'd want to have on your side in a troubled time, Mr. Steve Gibson from the Gibson Research Corporation, GRC.com. Hi, Steve.

Steve Gibson: Yo, Leo. Great to be with you again.

Leo: Yeah.

Steve: Episode 222.

Leo: Which we're recording on 11/11.

Steve: At 11:00.

Leo: At 11:00. Wow. Wow.

Steve: Lots of alliteration today.

Leo: Yeah. Well, I don't know if - again, this is not going to air until the 12th, but we're recording on the 11th. I don't know if - you don't say Happy Veterans Day, but a tip of the hat to our men and women who are serving and have served in the armed forces. We appreciate your service. And we're having a big parade in Petaluma at 1:00.

Steve: Oh, very cool.

Leo: So if you hear marching bands and cannons firing and all of that, that's the Veterans Day Parade going downtown. It's such an old-fashioned community. I mean, we do stuff like that. I love it.

Steve: Yeah, that's really neat.

Leo: Yeah, I wanted to drive the Mustang down the street, but I'm busy. But our VP Finance, Lisa's husband, Lisa and her husband Mike own a beautiful 1967 cherry-red Camaro.

Steve: Mmm, nice.

Leo: And he and their six-year-old son is driving that in the parade with their club. So that's really neat. So, my friend, it is time once again for a Q&A segment.

Steve: Brought to us by our fantastic listeners. 334 postings when I pulled the mail down, from which we selected just a handful, but good ones, interesting ones. And a really, really nice, well, and sort of - we wrap it up with, I think it was the Biometric Horror Story of the Week.

Leo: [Laughing] You know...

Steve: So, always trying to find something fun.

Leo: I was talking to somebody who works at Disneyland. He is a programmer, actually works for Pixar, but he goes - he has one of those cards that lets you go to Disneyland all the time. I said, "When you go there, do you scan your finger?" He said yeah. I said, "You should scan your knuckle." He didn't know what I was talking about. He's a computer programmer. He ought to know.

Steve: Yeah.

Leo: We're putting out the word, anyway.

Steve: So we have a bunch of security news. We've got a little bit of errata. I have an interesting SpinRite piracy story to tell. And a bunch of great questions from our listeners that we will answer.

Leo: Did you know - I don't know, I should have asked you this off the air, but somebody has made an iPhone application out of our Vitamin D segment?

Steve: I actually do know. I've communicated with the author a number of times for him - he wanted permission, and I said, oh, of course. And I aimed him at the Vitamin D page where I had a...

Leo: It has a link to that, yup.

Steve: I had audio that was just the Vitamin D stuff. And a number of our listeners wrote to say Vitamin D, the Vitamin D podcast is on iTunes.

Leo: We license our stuff using the Creative Commons License for noncommercial attribution share alike. So you're welcome to do mashups like that, provided you're not making money on it and you have the same permissions, use share alike permissions. So I think that's really a great way of sharing the information.

Steve: Yeah, it's very cool.

Leo: Yeah. So if you want to share Steve's great episode on Vitamin D - what is it called? Do you know?

Steve: I have not...

Leo: I have it right here. I just found it. It's called "Vitamin D: Listen and Learn."

Steve: Ah, perfect.

Leo: And you can get it by just searching for Vitamin D on the iTunes store. And this application was made to promote awareness of this misunderstood "vitamin." Information by Steve Gibson of GRC.com and designed by Ultra Software Solutions, USSapps.com. So I think that's really neat. And it does have the audio of the show. What a great way to share all this information. And it's absolutely free, it says, just like the sun. So that's great.

Steve: That's perfect.

Leo: Yeah, that's really great. Thank you, USS, for doing that. You want to do, I'm sure, some errata and news, I would guess? Yes?

Steve: We have all that kind of stuff.

Leo: All right.

Steve: And we had a Patch Tuesday. We're recording this after the second Tuesday of November. And we know Microsoft, they're up to the same old routine, as is pretty much everybody else. And speaking of the iPhone, we have news of the very first iPhone worm.

Leo: Oh, yeah. It's a funny one.

Steve: It's a wacky thing, yes.

Leo: All right. So let's hear what's happening in the world of security.

Steve: Well, we have pretty much all the regular culprits are present today.

Leo: The usual suspects.

Steve: Of course top of the list is Adobe. By the way, remember last week I said I hate Adobe? I still do. And now I discovered that, in addition to installing this unasked-for demo of the speech package, when I rebooted my machine after a security update and then next ran IE, there was now a toolbar that it had added for the speech demo that had appeared in IE. I just...

Leo: We had talked last week about Flash doing that; right?

Steve: Well, I'm sure that I remember, yes.

Leo: I tried it, and I saw there is a checkbox, but you have to notice it, on the web page before you download the Flash.

Steve: [Growling]

Leo: It's not even in the install. It's like before you download.

Steve: And did you see also the offer to install Google, I think it was a Google toolbar

also?

Leo: I don't remember what it was. But...

Steve: But you did see a demo for the speech app.

Leo: No, I saw the toolbar thing.

Steve: Oh, okay.

Leo: I don't know how this - but I just can't believe they would install something without at some point, however fine the print, saying we're going to install this.

Steve: Oh, I'm sure it was there. I mean, again...

Leo: But who sees them?

Steve: I'm sure it was there. I must have missed - maybe there was a way to turn it off, although I remember turning off the no, I don't want the Google toolbar, thank you very much.

Leo: Yes, that's the one I thought I saw.

Steve: And I didn't see the other one, so. And in fact I meant to look through our mailbag because I figured our listeners would have said, oh, yeah, I ran across that, and here's where it was. But I ended up filling up our Q&A sort of near the top of the pull and never got through reading everything, so I didn't confirm it. But in this case we have Shockwave, that has about 450 million users, which has five critical flaws, four of them allowing arbitrary code execution. I don't have Shockwave installed. So users should look in your browser's add-ons, or extensions, rather. For example, in Firefox it's not an extension, it's a plug-in. That's where those sort of like the non-extra UI features, but more of the built-in browser enhancement plug-ins go there. And so, for example, I see Adobe Flash, but not Shockwave, which is different than Flash.

So first of all, if you don't have Adobe Shockwave in, for example, your Firefox plug-ins, don't add it. I mean, it's just more gunk, you know. And I'm never one for installing stuff you don't need. But if you do have it, it is exploitable. And so you definitely want to go to get.adobe.com/shockwave, so that's the URL, get.adobe.com/shockwave. And it will, as of this date, give you v11.5.2.602. And anything earlier than that, which would be 11.5.1.601 and earlier, has these problems. So if your browser does have Shockwave, first of all, you may want to just remove it. If you don't think that you - if you don't know you need it. I'm surviving quite well without.

Leo: Most of the time you don't. Yeah, most people use Flash now. Shockwave is not necessary on the web.

Steve: Yes. So I think basically when they talk about 450 million users, it's like, yeah, well, these people acquired it at some point, and it's still living in their machine, creating vulnerabilities they don't need. So maybe, I mean, what would make the most sense is, if you know you don't need it, get rid of it. And if you do need it, then make sure you've got 11.5.2.602, which you can get from get.adobe.com/shockwave.

Also Sun, the latest version of the Java Runtime Environment, and I mean the latest version, has multiple vulnerabilities and no updates available. They have acknowledged multiple problems. There is enough disclosed for problems to be - for exploits to be created on the 'Net. Unfortunately the only workaround is the workaround we all know all too well, which is disable JavaScript to prevent Java, the Java Runtime Environment components from being exploited until Sun updates themselves. One of these weeks I'll happily report that there is a new version of the Java Runtime Environment. At this point there's notices all over the various security sites, talking about a zero-day problem, that is, problems that have been acknowledged and are being exploited, but for which there's yet no patch, which is the case with the Java Runtime Environment.

Leo: So a week ago Java was patched. And you're saying it has new holes now?

Steve: Yes.

Leo: For crying out loud.

Steve: Yes.

Leo: I can't believe it.

Steve: Yes.

Leo: They were just fixed.

Steve: Yes. Yes.

Leo: Okay. Okay.

Steve: There's some interesting news from the EU, the European Union. There's been a lot of issue over the rights of users to stay connected to the Internet when they've been accused of filesharing. And, you know, we've talked about sort of, I mean, there's really no global consensus yet. We've talked about how now Internet access is being considered a human right. Whatever that exactly means I think is unclear. But the idea

being, well, it's really important. And so given that, the question is, what level of proof does an ISP need to be given in order to disconnect users? France, under strong lobbying pressure from the music and film industry, has been proposing a three strikes and you're out approach, where you are notified twice that you are downloading illegal copyrighted content. And if you're found to be doing that a third time, after two prior notices, you're disconnected for one year. What's happening is the EU is trying to unify all of this various sort of unclear policy under a single agreement. And the good news is it's looking like agreement is being reached and Internet users are going to get some relief and a sort of a strong pro-user result where...

Leo: Good.

Steve: Yes, where none of this will apply, where you can't simply be accused. Remember that we talked about a story recently where just being accused of doing this, where the ISP was offered no proof, there were ISPs that were under pressure from the Motion Picture Association to disconnect users just because they said so. And the ISPs were saying, wait a minute, you know, we need more than that. So the good news is now it's going to require a court order that includes proof that this is actually going on.

Leo: There is, you know, this is a larger issue. This is the ACTA treaty, the anti-counterfeiting treaty that has been negotiated, currently being negotiated in secret. And it's not just France. They want to get this through all over the world. And by the way, it's really sneaky because, if the treaty is ratified, the U.S. Congress has to make this the law in the U.S. as well.

Steve: Ooh.

Leo: Even though it would never, nobody would ever, you know, concede to these really draconian provisions. It's anti-American. The treaty is in secret, but some of it has leaked out. Michael Geist in Canada has leaked it out. And it is in fact calling for that three-strikes provision, with three strikes accused, not three strikes conviction. France modified it. But the other countries are considering this, three strikes accused, and off the Internet for life. And your name being distributed so that other ISPs will not give you service.

Steve: Whoa. This goes way further than...

Leo: Horrible.

Steve: ...than the article. Because I was reading an article from the Associated Press that MSNBC covered.

Leo: It's a much bigger story than this. And if you want to read more, EFF.org, read about the ACTA treaties. And Michael Geist's blog in Canada is an excellent blog. He was the one who found these provisions. Now, it's still being negotiated. But the

point was it was being negotiated in secret. They obviously didn't want anybody to know about this. Now that it's in public, we need to say, hey, that's not okay. That's not acceptable.

Steve: Yeah, this is not okay. Off for life.

Leo: Yeah. Isn't that nice.

Steve: Come on.

Leo: EFF has gotten very involved in this because they really are afraid that - and it wouldn't go through a legislative process in the U.S., that's what's interesting. Because it's a treaty, it would have to be ratified. And of course that's what the movie industry and the record industry want. They don't want discussion over this. They want it to be secret, and they want it to go through without anybody knowing. And then it would be the law.

Steve: Wow.

Leo: You wouldn't have a choice.

Steve: And so accusation without proof, three times, and then you're banned from the Internet for life.

Leo: Isn't that nice.

Steve: That's horrifying.

Leo: They tried to do it in New Zealand. If you look at the list of countries, Korea, they're trying to get this through in, it's just appalling, yeah. So this is something everybody - I'm glad you brought it up because it's something everybody should be aware of. And find out more at EFF.org.

Steve: We'll keep our eye on it too. Microsoft brought us Patch Tuesday this month of November, as they always do every month. We had six security bulletins on Tuesday, November 10, so I'm sure everyone's Windows machines are lighting up with their little yellow shields, saying oops, we've got some updates for you which will require a reboot. Three were critical. Three were rated important. They were pretty much obscure, remote code execution, kernel-level things that involved the Windows kernel and a couple of the Office apps. So, you know, nothing really earth-shattering. But as usual, find a time when you can download them and shut things down and reboot your system. You want to stay current with that.

Both Leopard and Snow Leopard, the Mac OSes, were updated, Leopard to 10.5 and Snow Leopard to 10.6.2. Apple as usual is not being very forthcoming. They said, "This affects the stability, compatibility, and security of users' computers." Yeah. Thanks a lot. No kidding.

Leo: You know what else it does? And I'm sure there's security involved. It also breaks Atom support. So that if you're running a Hackintosh, which is a Macintosh on one of these Netbooks...

Steve: Interesting.

Leo: ...don't update because it won't work anymore.

Steve: Interesting.

Leo: Yeah.

Steve: Yeah. They said that it fixes a number of security issues...

Leo: Yeah.

Steve: ...including arbitrary - yeah. Yeah, exactly. And some usage we don't like issues, arbitrary code execution flaws, some cross-site scripting vulnerabilities. There was a denial-of-service flaw, privilege elevation flaws, unexpected application termination, of course unexpected boot termination in the case of these Atom machines. And also said that attempts to download unsafe content may not always produce warnings. And there was - apparently there was a way for dictionary attacks against SSH logins to not be detected. So you definitely want to update this unless, as you said, Leo, you really can't because you're using the Mac...

Leo: Well, they got you, don't they.

Steve: Yeah, on a non-Mac - on non-Mac hardware.

Leo: Yeah, yeah.

Steve: So, yes, update. It's 157MB, which I updated when I fired up my Mac this morning to fire up Skype and do the podcast with you, Leo. So it takes a while to grab it and update it, but definitely worth keeping current. Meanwhile we have iKee is the name of the first iPhone worm, which is spreading only locally throughout Australia at the moment because it scans for specific Australian 3G wireless networks. So it's not going to go global.

What's interesting is that the way this happened is sort of a gotcha. The most popular jailbreaking system for the iPhone installs an SSH server as part of its jailbreaking process. And unfortunately it has a default password which is set, which because being default, everybody knows what it is. So if the user who jailbreaks their iPhone in order to allow it to run non-Apple iTunes store-based software, what they're getting in the process is an SSH server which is exposed to the Internet with a default password everyone knows. So that's all you need to create a worm, which has been created. The good news is this is benign, relatively. It changes your wallpaper to some '80s singer guy that I've never seen or heard from.

Leo: Oh, you've never been Rickrolled, obviously. It's Rick Astley.

Steve: Okay.

Leo: It's an Internet meme. People were sending links saying, oh, this is the latest greatest thing, and it would be Rick Astley singing "Never gonna give you up, never gonna let you down." Anyway, it's a horrible meme. It's a prank. It's a prank.

Steve: Well, and one other Dutch hacker, actually even before this, there was a Dutch hacker who was using this hole to send offers to iPhone users to close the hole in return for them sending him five euros to his PayPal account. So there was sort of an attempt...

Leo: But you don't have to.

Steve: Well, yeah, exactly, you don't have to. All you have to do...

Leo: Just a suggestion.

Steve: Exactly. I'd be happy to show you how to change your password if you send me five euros, please.

Leo: Well, and that's the fix; right? You just change it from the default.

Steve: Yeah. That's, exactly, change it from the default, change it to something, you know, I mean, while you're at it, make it a gnarly password because you don't want, you know, the next thing that'll happen is - actually what you'd love to do, I mean, I don't know enough about the iPhone, I haven't looked at it, you would like to shut down this SSH service. I mean, that's really the solution, is why would you want an SSH service running in your iPhone?

Leo: A lot of jailbroken iPhones do so you can SSH into your phone.

Steve: Oh, okay. Well, there you go.

Leo: That's one of the reasons they do it. In fact, it's a necessity for, I believe, some of the jailbreaking.

Steve: So in order to set it up like underneath the phone's UI and so forth.

Leo: And then you can - you could turn it off, but a lot of people want to keep that - they hook up their iPhone, and now they can SSH into it. It's a full operating system. So you can mess with it.

Steve: In that case, change the password. Absolutely.

Leo: If you're going to use it, change the password, yeah.

Steve: Absolutely change the password. Now, the biggest scary news of the week is our topic for in-depth coverage next week. Which is the bad hole which has just been found in SSL.

Leo: Oh, crud. That's not good.

Steve: There's a session renegotiation hack which has been discovered in the latest current version, v3 of SSL, which we all know is also TLS. We covered the protocol at some length a while go. I didn't talk about, for example, session renegotiation because it's sort of off the mainstream of how two machines normally hook up. Turns out that it's possible for a man in the middle to attack an SSL session, that is, an existing SSL connection, and insert his own transactions into the stream.

So people are scampering around, this has been known for a couple months, it's been kept under wraps. Under NDA the details were given out. I mean, this is a big, big problem. The various standard bearers, the OpenSSL and the Gnu Project are working to update their specs. There'll be a new RFC. I mean, basically this is a fundamental flaw in the protocol that we're going to cover in depth next week. So I wanted to let everyone know I know about it because there was a bunch of people dropping mail to me saying, oops, have you heard about this? Oh, yes, I know about this, the renegotiation problem. So we will all know about it in detail next week.

John Graham-Cumming, who was our guest last week, prepared a PDF and PowerPoint files of his slide presentation. I knew about them last week, but they were in the PPTX format that required you to have the very latest version of Office. And so I'd asked him if he could do it in the older version of PowerPoint. He went a little further and also created a PDF. So you need no Microsoft PowerPoint viewer of any sort. So I just wanted to give a heads-up to people that at the Security Now! page at GRC - and by the way, I made it quicker to get there, GRC.com/sn. You don't even have to type out securitynow any longer, in response to some people in our newsgroups saying, hey, Steve, how about /sn? It's like, okay. You got that. So GRC.com/sn. And the Episode 221, which is when we had John talking about JavaScript, I put links there to both of those files, which I'm hosting locally, courtesy of John.

Leo: Great. That's nice of you. That's great.

Steve: In an interesting twist on SpinRite, from someone who gave me his name - I don't have any problem sharing it, I doubt that he will because he went legit in the end - Cody Krieger. He said, "Dear Steve. Not sure if this is the right address to send this to." Actually he sent it to our sales address, but Sue forwarded it to me. "But to be honest, I'm feeling a bit lazy after the ordeal I've just gone through. About two months ago I underwent a fairly serious hard drive malfunction. Two went dead. Both had an extreme case of the click of death, and another started to fail. These were all in the same machine." And frankly I'm beginning to wonder maybe if his machine's, like, way over temperature or something.

Anyway, he says, "I immediately whipped out" - what he called "my trusty copy." We learn a little bit later it wasn't quite his. "I immediately whipped out my trusty copy of SpinRite and set it on data recovery mode. After the long wait, I discovered that SpinRite had brought both" - he put in bracketed and asterisks - "drives back into working order. Not feeling very trustworthy of either drive after all that clicking, as I suspect both drives' failure can be partially attributed to firmware or mechanical defects, I immediately imaged both onto their own brand new 1TB drives. Not 10 seconds after this finished, both drives collapsed dead again. This time they failed to even spin up.

"I then ran SpinRite on maintenance mode on the drive that was starting to go bad. That drive has lasted me up until now. Today that drive started clicking angrily at me while my system was sitting idle for the most part. I'm running SpinRite on it right now, and I'm buying another drive to replace it with in anticipation. Having realized how many times SpinRite has saved me (two months ago wasn't the first time), I figured it was time for me to actually buy a copy of the software. I confess I had been using a pirated copy. No single piece of software has ever worked wonders like this has for me, and you deserve every penny of my money for it. Thanks for making such a wonderful piece of software. Keep it up. Cody."

Leo: Wow. So isn't that nice. I love these stories. They're so heartwarming, in a geeky kind of way.

Steve: Well, yeah. I mean, and I frankly, I mean, I have a mature appreciation of the reality of software sales. I never locked my software or copy-protected it.

Leo: Thank you.

Steve: Never done anything like that. That just gets in the way. I ignore, like, if you look at anyone else's hard disk utility software, they tell you you can run it on exactly one drive. Well, that's ridiculous, too. I mean, if a person's going to buy it, I fully expect they're going to want to run it on all the drives they own. And that's what the license is for. Use it on any drives you personally own. So, you know, if someone is going to use a piece of pirated software and accept the risk of doing so, I do know unfortunately that there are damaged, deliberately damaged versions of SpinRite that install bad stuff on people's computers.

One of the things that Greg deals with, my tech support guy, is someone will call up and

say, hey, you know, SpinRite just installed a trojan on my computer. And it's like, I don't think so. What's your serial number? And then the guy, uh, oh, well, this isn't really my copy. And, you know, and then we go from there. So, you know, certainly there's a risk of doing so. But if someone's going to do that, then they're not a customer of mine. So I don't recognize, I don't regard it as lost revenue or piracy. I mean, you know, it'd be nice if the MPAA and the music industry had a similar mature feeling, recognizing that not every piece of song or movie that is being viewed or listened to represents lost revenue. But, you know, it's not the way they operate. So anyway, I appreciate Cody's purchase and his great note talking about how - what a good job SpinRite did for him.

Leo: I'll start with Question #1 from Mike in Baltimore, Maryland. He wonders about changing his SSH port every single day. Hmm, that's interesting. Every time you talk about SSH and port configuration, I remember I want to ask this question: In addition to logging the SYN packet request, would changing the SSH open port number every day to a randomized port provide any additional security? My Linksys router running the Tomato firmware allows me to map an external port to a different internal port. I can map external port 62305 to internal port 22 - the SSH port - for example. So if I wrote a script to randomize that external port number every day, would that provide any security benefit? Or am I wasting my time? Thanks for the great podcasts. As soon as I stop spending so much money I will be buying SpinRite and running it on all my drives. Maybe spend some money on SpinRite first. I currently have no problems with any, but a few are quite noisy and starting to worry me, and I would like some peace of mind. Well, it's an interesting strategy. What do you think?

Steve: Well, I'm not a big fan of that. The idea, I mean, it's sort of clever. My sense is you'd get much better benefit from using a really good, really impossible-to-guess username and password. Certainly moving the port around every day would prevent someone from sitting there, pounding on it, on a fixed port, even if it wasn't 22, for some period of time. On the other hand, if the port disappeared, I guess they would figure you'd closed the service. But they could also just scan your IP and find where you'd moved it to.

The question, though, most put me in mind of I think perhaps, depending upon how Mike uses SSH, the best solution of all, which is not to have the port mapped at all unless he has some reason to believe he will be logging in over SSH from somewhere outside of his home or office, and thus need the port mapping at all. That is to say, not running the service when you don't need the service, if that's feasible, is by far the best solution. Microsoft famously got into so much trouble during all of the early years of Windows, specifically because they were running services by default all the time on every version of Windows, when it was installed, even if users had absolutely no need for doing so. Remember we lobbied, for example, about the Universal Plug and Play service. And I just said, you know, it should not be running. Nobody needs it running. Sure enough, there was a bug found in it, and it became a big problem.

And we just, you know, the whole issue of SSH came up because remember there was a protocol flaw found in a Windows SSH daemon that was - it was a popular free SSH daemon that it turns out skirts around your need to even log in. So there's a perfect example of where you just, if you don't need the service running, it's really better not to just leave it open all the time.

Now, for many people it may not be feasible. The way they operate, their lifestyle may be such that they can't explicitly, for whatever reason, fire up the service when they

know they're going to be remotely accessing home. Maybe they never have notice of when they're going to be doing so. But it may also well be that a person's mode of using remote access does give them the ability not to have this thing open all the time. In which case, turn it off when you're not using. That's by far the best solution. Failing that, moving it around daily is interesting.

On the other hand, if you didn't know where your script had remapped the port, then you'd have to be checking the port mapping before you left the house in order to know what today's port was that had been chosen. So that seems a burden also. Why not just turn it off if you're not using it. And, you know, if your port's jumping around all over, and you don't know what port it's on, it might as well be off anyway because you're going to have no better luck finding it than a hacker would.

Leo: Yeah. I mean, that's security by obscurity; right?

Steve: Yeah, I mean...

Leo: And not even that obscure because a port scanner's going to find it right away.

Steve: There's nothing wrong with some obscurity. That's, I mean, that's just another layer. But you don't want to depend upon obscurity. So when people talk about security through obscurity and say that it's a bad thing, what's bad about it is if you depend upon its obscurity. It's not, you know, there's nothing wrong with adding some obscurity. That just gives you more security. But you don't want the obscurity to be the only security that you have.

Leo: Better to implement port knocking, if you could do that; right?

Steve: Yes. And we've got an interesting question about that, too, in our question bag of today.

Leo: Well, port knocking coming up. Let's get to Question 2, however. Dana Rae Park in Kelseyville, California declares herself to be a "SNite head," she calls it. Through Leo's radio program, I got turned onto Security Now. Wow. That's interesting. Because usually I think people who listen to the radio program are less sophisticated. You know, I don't expect them to be the geeks. In fact, I even say when I talk about the podcasts on the radio show, I say, "Now, if you really want to geek out...." But this is the geekiest of all the shows we do, so that's pretty good. Well done, Dana Rae.

It took two weeks, but I finally downloaded every one of Security Now's 220 episodes. Steve, your reminiscing about old computers took me way back to when I was selling Apple IIs and IBM ATs. Well, I guess Dana is a geek. I've been out of the game for quite a while. Until a couple months ago I was using an ancient Compaq desktop Windows 98SE machine with dialup. Welcome to the '90s.

Steve: Can you still do that? That still works? Is there some modem somewhere you can

call?

Leo: Then a friend of mine kludged together an XP Pro machine for me, and my landlady gave me access to her ATT DSL modem/router via a 100-foot cable. This was a quantum change and challenge for me. I bet. Steve, following your advice I use Firefox with Adblock, NoScript, Tree Style Tab, Flashblock extensions, and the very useful KatMouse - five Steve Gibson-recommended Firefox extensions.

She says, when I access the 2701HG-B Gateway System Summary through my browser, there is a Firewall icon which tells me, "The firewall actively blocks access of unwanted activity from the Internet." Am I behind two firewalls, one on the router and one on XP? The Summary also says "Your system software is current. Check back for future available upgrades." I don't know what the 2701HG-B gateway is. Do routers phone home for updates like XP? Am I safe? Am I practicing safe computing? SNite Heads For a Safer Internet, she signs it. That's great.

Steve: Well, there are a couple things here. First of all, I wanted to remind people about KatMouse, which Dana refers to. I just love it. And I've noticed, I've been using one of my Macs a lot more than usual for some - actually because the best PDP-8 emulator is running over there. And I guess the Mac, Leo, when you float your mouse over a Mac window and use the scroll wheel or ball or whatever, it's smart about automatically scrolling the window that you're over; right?

Leo: Yes. I never noticed that, but you're absolutely right.

Steve: So you don't have to click on it in order to bring focus to it.

Leo: Wow, that's a nice feature. I never noticed. Steve, you found something. I never saw that before. You're absolutely right.

Steve: Well, it's a fantastic feature. And what's what KatMouse does for Windows. And I wanted to bring it up again because I've had so much positive feedback from people who love it as much as I do. Just the idea that you can have multiple windows. And, I mean, I'm a scroll-wheel fanatic. I love my scroll wheel. And in fact we talked about the Logitech mouse which is still my favorite, that has that high-inertia zero-friction scroll wheel where you can just, like, spin the wheel and go whizzing through things. It's just great.

Leo: It's a VX Nano. In fact, I'm buying more of them. I love them.

Steve: I have some now in stock because I like it so much. Oh, actually, though, there's a better one, or one that I like better. And that's the - it's the MX Anywhere. It's also Logitech. I think they call it the MX Anywhere Mouse. What I like about it is that the VX Nano, you have to - it provides you the ability to store the little tiny, tiny little receiver, the USB receiver...

Leo: Yes, right in there.

Steve: ...underneath the mouse. But in order - and doing so turns the power off, which is what's nice. But that also necessitates that you're constantly opening the little battery door and pushing the red button to pop the thing out and so forth. The MX Anywhere Mouse has also internal storage, but the idea is the receiver lives normally just in your laptop, on your laptop. And then there's a cover that you just slide with your thumb to cover up the laser tracking hole.

Leo: That shuts it off?

Steve: Yes.

Leo: That's much better.

Steve: It's much better. The mouse also is a little bit larger and a little heftier feeling.

Leo: It's not a laptop mouse. It's more of a desktop mouse.

Steve: It's more of a desktop mouse. But no, but it's meant for laptops because it's got the same micro-size receiver. And so anyway, I really - I've switched to that one, and I like it more because it's a nice-feeling mouse. But mostly you just, you know, when you're done, you close your laptop, you leave the receiver sitting in the USB hole, the USB socket of your laptop, and then just slide the cover closed on the laser positioner, and it powers it down.

Leo: So my only question on this, is it a leftie-friendly mouse?

Steve: Oh, I'm a left-hander, but I right mouse. I mouse with my right hand.

Leo: So is it symmetric, I guess would be...

Steve: Not exactly symmetric.

Leo: It looks like it's slightly asymmetric, but not so much that it might...

Steve: No, it's not horribly so. And I also like that it's got buttons all over it. It's got, you know, both the - you can do the tilt wheel, and it's got a little button back behind it that works like, for example, the Mac does that Expos thing, is that what they call it, where all the windows short of shrink down and go to their corners so you can quickly choose between windows?

Leo: Yeah. It's nice to have those features.

Steve: Yes, so it's got all the...

Leo: And you can remap those if you want. I mean, you don't have to...

Steve: Actually I have. I've got the mouse - who wants to horizontally scroll? I never...

Leo: I know, I don't understand that one.

Steve: No. So I've got that set up for back and forward the browser, so that I tilt the wheel to the left in order to, like, go back to the prior page, or to the right to go forward. And which frees up the other two buttons on the left-hand side, which I use for top of page/bottom of page in the browser, so I can instantly jump to the top if I just press the upper of those two, or jump to the bottom, which I think is much more useful for me. So anyway, so yeah.

Leo: Well, I'm going to try this because I'm running out of VX Nanos. So I have to get a new one.

Steve: Yeah, try the MX Anywhere. I really like it. I think it's better than the VX Nano.

Leo: I'll get a couple of them.

Steve: And I've got a few used VX Nanos, if anyone wants them. No.

Leo: Don't say that. You'll get some emails.

Steve: And also...

Leo: Now, let's answer her specific question, though; right?

Steve: Yes. She is behind, Dana is behind two firewalls. And I thought that was an interesting point that we've never really made before, and it's something we're probably going to be getting used to more and more in the future, is we're all going to be behind an increasing number of firewalls. You know, over time ISPs are beginning to do some firewalling themselves. Then you've got the firewall in your router. But any router, even if it doesn't specifically say it has firewall features, the nature of NAT routing is such that unsolicited packets coming in don't have anywhere to go. There isn't - by having packets leave the network out onto the Internet, a temporary mapping hole is created to allow packets coming back from that packet's destination have a way to come back in through

and know which one of the computers behind the router to go to. Failing that, unexpected packets just hit the router and drop. They're just ignored because there's nowhere for them to go. So any NAT router is a firewall.

And then, thank goodness, Microsoft has turned on their firewall that's built into their OS. And I think probably every personal operating system now, no matter whether it's Microsoft, Mac, Linux, anything, there's a firewall component there which is blocking unsolicited inbound traffic. So, yes, there's multiple layers there.

And finally she asks about router firmware, which I think is the most important issue and point of her question. And it's something I wanted to bring up because routers do have problems. Manufacturers are finding problems. And I don't think there's ever been a case where I've checked with my router's firmware, and there has not been an update available. Many times they're little non-security-related things. But just the other day we talked about a really ridiculous router that was broadcasting its WiFi key in the SSID that it was using, and also the MAC address.

Leo: Oh, yeah. So stupid.

Steve: And there was a router that had a default password exposed out to the WAN side. So there are definitely instances where manufacturers are being apprised of serious problems that firmware updates are fixing. So I just thought this question was a nice little reminder to me and all of our listeners to go visit your router. We don't do that very often. You know, log into your router's admin page. Now, most routers require you to explicitly check to see if there's a firmware update. It sounds like this 2701HG-B, whatever it is, if it's volunteering that the system software is current, it does sound like it's phoning home. It's pinging or doing something from time to time to see whether the firmware is current or not.

Now, a router typically has no way of affirmatively notifying any machines behind it. So that's why it's necessary for you to go to the router and check in with it and ask it, is there a new version of your firmware? I would just recommend, you know, I don't know how to add a tickler to everyone's life, except, you know, here, this podcast, #222, Dana's question sort of reminds us it's useful to check in with your router from time to time and see whether there's newer firmware for it. Because who knows what they will have fixed?

Leo: Yeah. It's something I always forget. So we'll just have to make it, you know, we'll just remind you.

Steve: Right.

Leo: Keep it in mind. Check it from time to time. I think there's no - as far as I know, there's no router that does it automatically. You kind of have to check.

Steve: The only thing they could do would be to bring up a web intercept page, which would be a little jarring for people.

Leo: Yeah. You don't want to see that.

Steve: So if you were trying to go somewhere, and your router blocked you and said, hey, I've got new firmware, I mean, on one hand that wouldn't be - that'd be a cool feature if you could turn it on, and you chose to turn it on. But it's not the thing you probably want to have on, or certainly not be able not to disable. So...

Leo: You might not want to rely on it, either.

Steve: Yeah.

Leo: Question 3 from Andrew DeFaria in Tempe, Arizona with some SSH tips. Here are some SSH tips you didn't mention for securing SSH. We should - we've said SSH a few times. We should mention the secure way to get a terminal session on, at least as far as I know, on Linux and UNIX-based computers; right? I don't know...

Steve: Well, exactly. It stands for Secure Shell, where "shell" is the term for, like, getting a command prompt window from a UNIX environment. And there are certainly, you know, we talked about the free SSH daemon the other day that had the problem but is available for Windows.

Leo: Oh, yeah.

Steve: And certainly there is SSH for Windows, as well.

Leo: What do you get, though? Do you get a DOS - I guess you get a DOS prompt when you do that on Windows.

Steve: Well, and there...

Leo: I don't know what you would get. Because, I mean, on the Mac, there's a terminal, you know, there's a command line. And on Linux there's a command line. But I guess you'd get DOS.

Steve: Yeah. I would think you would get a command shell. And actually, you know, the Windows command has become increasingly powerful over the years so that it's not so much just running a pokey old copy of DOS that has no awareness of what the system is doing. You can start and stop services. You can reconfigure your Internet connection, I mean, there's all kinds of things you can do now from a contemporary Windows command line.

Leo: And if you have a web server, SSH is often the way you will manage it and control it.

Steve: Well, and yes. SSH then, the big thing that people do is it can be used as a tunneling protocol. We've talked about tunneling where the idea being you use one protocol to route packets for another different protocol inside. And so SSH is often used - it's a little bit of a techie thing to set up. But it works sort of well. The other problem is that SSH is a TCP protocol. And there are all kinds of problems with tunneling TCP in TCP because, if you are in a situation where there's packet loss, then the SSH tunnel will lose packets, but those packets contained TCP traffic that you were tunneling. And so you can end up with a situation where both of the TCP connections are timing out and are doing retransmits, and you get something called a "tunnel stall" as a consequence. So the fact is, SSH is not an ideal tunneling protocol. But it's sort of a poor man's VPN, and it works. It's not what I would recommend, though. But, again, it works.

Leo: Yeah. So continuing with Andrew DeFaria's recommendations, if you want to run SSH: Turn off username/password functionality. Use a preshared SSH key only. And I do that when I have SSH on my servers. So that way it's using public key crypto, actually. It's kind of cool. You generate a key, and you share that key on the server. And when you log in, you don't have to give a password because you offer instead your key. Is that right? Am I understanding it correctly?

Steve: That's exactly right. And it's something I should have mentioned. So when I saw this, I thought, ah. And I ought to give other people credit. Many other listeners who use SSH said, hey, Steve, forget about this whole username/password guessing altogether. Remember we've talked many times about how SSH servers are coming under tremendous attack from just brute-force password guessing. So it's possible to completely eliminate username and password functionality and, as you say, use PKI, public key infrastructure, use a previously set up SSH key which your roaming client, your remote client will have, your local server will have, and that's the way you negotiate the connection. So there, you know, you can let them pound on your username and password all they want. They will never get in because they won't have this preshared key that you've established beforehand. That's absolutely the secure way to go about this. But it's, you know, takes a little bit more configuration.

Leo: Yeah. I have a whole step-by-step thing that I just run. Tell you, it's a real convenience, and it's kind of nice, it's one of those few things where it's more convenient and more secure. So it's kind of a nice thing to have. You know, I just SSH to my server, and I'm in automatically. And nobody, I think nobody else, unless they could get my key, could do that.

Step 2. This one's even more complicated. I put together a Perl script I call "bice" that scans the logs nightly looking for attempts to break in (of course they can't because they'd need a preshared key but try they do nonetheless) - and I see that in my server logs all the time, dozens and dozens of attempts. And they just - it's brute force. They're just trying random passwords - and then emails the upstream provider to tell them to stop. I'm not sure that's such a good idea.

Steve: That's exactly my feeling, Leo. I get, I mean, I've been running Security Now! for

- Security Now!. ShieldsUP!. Been running ShieldsUP! for I don't know how many years, and we've got - we're on the high side of 80-something million uses. And every so often there's some, I won't use any derogatory term, there's a person who has some sort of automated log-reading emailer that emails complaints to Level 3 that I'm trying to break into his system.

Well, I've never done that in my life, and I never would. You know, he's using, he or some user in his network, is using ShieldsUP! over an SSL connection so that we know that we're bypassing any proxies and we're connecting directly, we get the real IP. So it means that, you know, he asked us to scan the ports of his machine, which then logs the scan, and has an automated emailer that sends out complaints to everybody in the world. So, and I among them.

So it's just - this automated email thing I think is just really dumb. And unfortunately it's done enough that all it's doing is causing the security services that are valuable, that belong to ISPs, to stop reading their mail because they get all this automated crap, frankly, that they just think, well, there's no person here. This is just some bot that the guy set up, and oh, isn't that clever, read his logs and send email.

Leo: Yeah.

Steve: No.

Leo: Yeah.

Steve: Bad idea.

Leo: And, you know, there's so many break-ins, I don't really - or attempts to do this, I don't really monitor them because it just tells me something I already know. Most of them come from China.

Steve: IBR, Leo.

Leo: What's that? Internet Background Radiation.

Steve: Internet Background Radiation. That's my acronym for it. Unfortunately, this is the reality. We've got worms, I mean, this Code Red is still out there, living on some machines, scanning around, looking for a vulnerability that we hopefully knocked off, you know, many years ago. If you grab a bunch of IPs and monitor the traffic, there's just garbage on the 'Net now that will never go away. It's Internet Background Radiation. It's not aimed at you. It's just aimed at random IPs and hoping that, you know, I mean, stuff that's just dead, that just, I mean, has no chance of finding any recipient target. It's just out there, you know, packets on the 'Net.

Leo: It's amazing, isn't it.

Steve: Yeah, it really is. I mean, it's sort of like a life form now.

Leo: It is, it's alive.

Steve: Yeah.

Leo: Tip 3, you mention you can't trust your logs if the hacker gets in. Seems to me a nice modification would be to change syslog or other logging facilities to shadow all logs to a location not known to the intruder, perhaps even on another system, then perform comparisons with the original and shadowed log files. Any discrepancies would indicate a break-in to be investigated. This is actually a class of larger programs, intrusion detection programs that do things like this; right? They're always looking for changes in fundamental system files.

Steve: Well, again, there's a right way and a wrong way to do this. And I thought this was an interesting point. I mean, I agree with Andrew. The problem is, if you change syslog or other logging facilities that are running on that machine to also log somewhere else, how is that going to be unknown to an intruder who's intruded into your machine and is able to look at everything you are? So the only way to do this is if you are logging on the wire, as they say. That is, not logging in that machine and having that machine send duplicates of its logs somewhere else because, if it's doing that, then presumably the bad guy can know that and go there in order to defeat them. And in fact we've seen recently a situation where bad guys were able to get in and were able to use the keys that a logging system had in order to then gain access to the system they were logging to. So exactly this kind of thing can be done.

The only way to do it safely is if you have sort of a separate machine which the machine it's monitoring has no awareness of, which is promiscuously monitoring all of the data, and exactly as you said, Leo, like an intrusion detection system. It's looking at all the traffic and logging it itself. In fact, such a machine doesn't even have to have an IP. I've got one myself set up that way at Level 3, a machine which is logging all the traffic, but it's able to be on an Ethernet without an IP. Because remember, Ethernet uses MAC addressing. And the IP layer, which is created by ARP, is just a convenience to allow machines to be numbered by Internet protocol within the network. But it's entirely unnecessary. So I have a machine that you cannot address by IP because it doesn't even have one. And it works just great, monitors all the traffic on the network. And that makes it quite invulnerable.

Leo: Let's move on to our next one, shall we? But anyway, thank you, Andrew, for some good ideas.

Steve: Yeah.

Leo: It's all about thinking about security. That's the main thing.

Steve: Great talking points, absolutely.

Leo: Duane McElvain in Chicago, Illinois wants some clarification on SSL. Oh, boy. You gave me such bad news this morning. Ai-yai-yai. Steve and Leo, on a previous episode someone brought up the idea of using SSL on every web page, every time. This sounded brilliant to me, and I believe your consensus was that, while it's not necessary, it's a great idea. Actually we said that the cost nowadays on modern machines is so low, why not? The drawback is the server overhead could be crippling - I guess he's right. For a high-volume site, that could maybe be a lot of overhead.

Steve: Because, yes, all of the individual users are concentrating their connections to a single server.

Leo: Yeah. Wasn't really thinking about that. Later in Episode 220 you were discussing public key encryption bit lengths and how (roughly paraphrased, please forgive me) 2,048 bits is mathematically a huge jump up from 1,024, but processors these days have no problem with it so that's why some sites potentially use 2,048 bits. Pardon the nitpick, but which is it? Is it overwhelming or not? I just want to make sure I'm understanding you two correctly. Do you think that encouraging (if not requiring) SSL for all websites would work without crippling web server capacity? If not, what's the downside (aside from computational cost)? I'm a big fan and have been since the single-digit episodes when I

discovered your show. SpinRite has saved my bits at least twice. Yay. Keep up the fantastic work, both of you. Regards, Duane McElvain.

Steve: Okay.

Leo: He's confusing issues here, isn't he?

Steve: Well, it's because it's a confusing issue.

Leo: Yes. I'll give you that.

Steve: If we were still back in the days of HTTP 1.0 and SSL version - well, I would say v1, but that one never really went to the world - v2, then there was a concern because browsers were initiating connections for - there was a constant stream of connections, maybe 10 or 12 or 15, if you had a page with lots of, for example, images on it. When the page came with all those image references, the browser would open connections, individual connections to retrieve each one of those images. If you didn't want the so-called mixed content warning, where it says remember - and IE and various browsers like Firefox have different terminology. But it's that warning that says this page contains some secure and some insecure content.

Well, that freaks people out. But normally that means that things like ads or images are not coming across SSL, where the page itself did. And it can be a concern, but it's not necessarily a concern. So the way you solve the problem is, if the page comes across by SSL, then you'd like all of the references to also use SSL, so the whole page is secure. That meant that, back in the HTTP 1.0 days, that the browser would open a flurry of

connections back to the origin server in order to retrieve all those bits and pieces. So each of those connections would require an SSL handshake and would seriously damage the server. So back then the webmasters quickly learned not to leave sites in SSL. Switch the user into SSL when necessary, and quickly get them back out as soon as not because it was going to be much faster.

Well, many things have happened since then. So we went to v3 of SSL, also known as TLS, that we'll be discussing next week for reason of this recently found man-in-the-middle attack problem. One of the things that we learned when we talked about the SSL protocol in detail is that it is possible to cache credentials. That is, that expensive setup process only needs to be done once, the first time a client and a remote web server talk to each other. There are time limits. But they're ample. And what that means is that the client is able to say, hey, I have a fresh credential from a recent connection to you. How would you like to reuse that? There's no reason for the server not to do so, and servers do. What that does is it completely short-circuits the expensive part of the public - the expensive public key part. They do negotiate a new shared secret symmetric key, which is what you want for a new connection. But that's instant. That takes no processing power at all. So first of all, you get credential caching.

The other thing that happened is HTTP evolved. That is, the protocol that SSL would be carrying evolved to 1.1, where browsers stopped opening individual connections for every asset that they were querying from a given server, and by default they limit their connections to two so that a browser will open up to and no more than two connections, and then the protocol was enhanced so the browser could pipeline. The browser could send multiple queries to the remote server. The remote server could return multiple assets all over a single connection. So instead of bringing up a connection, getting one object, and then taking it down, bringing up another connection, getting one object and taking it down, now the browser will bring up some semi-persistent connections and do all of its work through them.

So even if you didn't have credential caching, you no longer have a flurry of connections being brought up. The browser brings up to two connections and then will hold onto those as the user moves around the site, being able to continually move pages and the server's assets, other site assets, buttons and images and so forth, through those connections. So my feeling is both with SSL 3, also known as TLS, with its credential caching, and the fact that all browsers now support this pipelining of multiple assets through a single connection, there just would not be a burden on contemporary servers if the sites used those features and left people in SSL all the time.

Leo: There you go. The definitive word.

Steve: You're right.

Leo: Jason M. in San Diego raises an interesting point: Steve and Leo, thanks for the great podcast. I always look forward to seeing the latest episode pop up on my media player. I have to take slight issue with your justification of - well, here we go, this is right down your alley here - of using public keys of 1,024 bits, specifically that they will expire after usually no more than three years. Now, that's true of the certificate, but not necessarily true for the key. The certificate may expire, but nothing prevents the site operator from generating a new Certificate Signing Request (CSR) against the same public key. I would even be so bold as to wager that very

few sites actually generate new key pairs in conjunction with a new certificate. Likewise I could wager that the Certificate Authorities do little, if anything, to prevent their customers from reusing key pairs or even informing their customers about the issue. As someone deeply involved with security, I'm always impressed that each episode provides new, relevant information. As someone who does technical education and presentations, I'm amazed at how smooth and professional each episode is. Keep up the great work. Well done, Steve.

Steve: Well, Jason raises a great point. So let's elaborate a little bit. What he's talking about is that there's sort of multiple steps for manual production of a key over in the UNIX world. In the Windows world, it's pretty much automated. And he raised a question, and I thought, well, I wonder if all of my keys, for example, for GRC.com over the years have been the same. So I went and looked. And every single one of them is different. Microsoft encapsulates the...

Leo: It does it automatically.

Steve: Exactly. It encapsulates the process through the GUI. You press a couple buttons. And it says, okay, where do you want to write your new CSR, your Certificate Signing Request? So what's happened behind the scenes is that a hopefully very good, cryptographically strong, random number generator has generated a new key pair, a public and private key pair, which you want to be based on very good, high-quality randomness so that it can't be guessed. And remember then that what we're doing is we're keeping our - we're keeping one of those to ourselves. It's secret. The other one we're going to be publishing. So what we want is we want then - we want to provide those to the certificate authority to prove that we're the owner of this key pair, and then the Certificate Authority is going to digitally sign what we call the Certificate Signing Request. We send them the CSR, the Certificate Signing Request, requesting that they sign the certificate. So it's that certificate that then has this typically three-year expiration which is enforced by the Certificate Authority. You can typically buy certificates for one, two, or three years, not longer than that, which is annoying. But that's - it's a tradeoff.

Leo: Well, they have, yeah, they have to have some expiration; right?

Steve: So Jason's point is that, since technically the typical user controls the generation of the CSR, that there's nothing to prevent the user from just having the Certificate Authority resign the same key pair. And he's right. Over in the UNIX world, using for example the OpenSSL package, you have a much more manual process for generating your certificates. And if for some reason you wanted to keep the same public key pair, you could do so. You could simply use the same one to submit, to generate a Certificate Signing Request to a Certificate Authority. They would sign it. I mean, they're doing no policing of whether...

Leo: It's not their job.

Steve: Exactly, it's not their job. And that's the other key point is, yes, they're definitely

not going to tell you, wait a minute, this is the same key pair you gave us three years ago. Well, yeah. You may have a reason for wanting to do that. I can't think of a good one, but you might. So Jason's point I think was very well taken, that is, that certainly anyone who understands security will absolutely change their public key pair. They'll take the opportunity of renewing their certificate to do that. But he's completely right that there's nothing, nowhere in this system are you forced to do so. And what that would mean is that a 1,024-bit public key could have a lifetime greater than three years. It could have a lifetime of 40 years.

Leo: Forever.

Steve: 50 years, if you kept - if you insisted on continually using the same one. You'd be exposing yourself to a larger attack window because there would be much greater time for someone to get to your key. I can't think of why you would. Windows doesn't give you the option. UNIX does. But standard protocol always has you regenerating your public key when you're updating your certificate. So even then, it seems unlikely that it would happen. But his point is a very good one. It could.

Leo: Yeah. Just something to be aware of, again.

Steve: Well, and I liked it from a theoretical standpoint. It strengthens our understanding of what's going on.

Leo: Yes, exactly, yeah. Paul Wilde in Bristol, UK feels that security shouldn't annoy the user: Guys, love the show, very informative. I'm in the UK. My bank, Smile - makes you just happy to be doing business with them - has just introduced two-factor authentication with the addition of a card reader pin-generating add-on. Great, I hear you say, but it's not. It's a big calculator-sized thing you have to use when you want to use functions like bill paying. Great, I hear you say again, but it's not. It means you can only pay bills, et cetera, when you have this stupid, fat, ugly thing with you. A credit card-sized add-on or a key-ring dongle is the way to go. It annoys me so much I'm moving the bank account. Keep up the information overload. Best, Paul Wilde. You know, the best to me is cell phone. And my bank, BofA, uses that. You press a button, it sends you a text on your cell phone with a number, and you use that. And I always have the phone with me.

Steve: Yeah, I think that's pretty good. It's funny, his - first of all, I completely understand. I think his bank is going to get a clue pretty quickly as they start losing accounts when they tell people. And this big thing can't be cheap, either. So I'm wondering, you know, who's paying for this big calculator-sized thing.

I overheard a conversation at Starbucks last week that I made a note, I mean, literally right then I sent myself a text message or email so that I would get it at home, so I would make a note, so I could share it with our Security Now! listeners. A guy explaining to his buddies that the password policy at his company was so obnoxious because they made him change his password periodically. And so immediately my ears perk up. It's like, oh...

Leo: That's good. That's what you're supposed to do.

Steve: That sounds like a good policy. And he says, the problem is they apparently remember the passwords you had before. So you can't change to the same one. You can't fake it out and use the same one and not really change it. Nor can you change to the one that you had before. So you can't ping-pong between two. And at this point everyone started listening with rapt attention. And he says, I figured out that apparently they remember the last five.

Leo: Wow.

Steve: So every time I am told that I have to change my password, and I don't want to because I like my password, I sit there and change it...

Leo: Five times.

Steve: ...five times. And the final time I go back to the password I want. And I just...

Leo: That's just silly.

Steve: I was, well, but here's the point. This is, I mean, there's a big lesson here, of course. That is, if users are absolutely determined not to have security, they're going to arrange somehow not to.

Leo: Well, yeah. Even if it's a real big pain in the butt, apparently.

Steve: They're going to write the password on their forehead. They're going to scratch it onto their LCD screen. They're going to do something. I mean, they're going to stick it underneath their keyboard. They're going to do whatever they can, you know, to get around the best security intentions of whoever's trying to protect them from themselves. I mean, you could imagine this policy where it's like, okay, we'll remember the last five because we really, really, really, really, really want you to change your password and not use any that you've used before. Now, all of us, any of us who are programmers or algorithm designers, we know how to defeat what this guy has done with his, you know, five in a row.

Leo: You make it for 20.

Steve: Well, you make it 20, or you prohibit them from changing the password...

Leo: Right, five times.

Steve: Like once a day or something.

Leo: Yeah. Yeah, yeah.

Steve: Anyway, so I just - I got a big kick out of the idea that this guy was - he was just determined to keep the password he likes. Which if course is really not secure. It's like then never changing your password, and he's figured out how to arrange never to change his password.

Leo: It blows my mind that he goes to such lengths to do it.

Steve: Yeah. And, you know, and so when I read Paul's note, "Security shouldn't annoy the user," I thought, okay, well, and on the other hand, security policy should try to strike a balance between not annoying the user and getting the user to behave. I guess, you know, it would take maybe some security training for this smart-aleck whom I overheard at Starbucks to understand why this was in place. And, I mean, and he could use different passwords for every month of the year or something so that he...

Leo: He's just determined. He says, I don't want to have to remember another password. But, I mean, the amount of time it must take to enter and change it five times...

Steve: Yup. He's going to win. He's going to win.

Leo: That's really what it's all about. This has nothing to do with anything but I'm just mad, and I'm going to win. Jason in Rochester, Minnesota had a brainstorm: Steve, I got an idea while you were talking about port knocking a couple episodes back. Since each port knock conveys one fewer than 16 bits of information - okay.

Steve: Well, okay, yeah. Remember, because ports go from - there's no port zero.

Leo: There are 16 bits a port.

Steve: 16 bits a port, but minus one because there's no port zero.

Leo: Right. 65,000...

Steve: So it's 65,535, not 65,536.

Leo: Okay. One fewer. Four port knocks equals close to, but not exactly, 2^64 combinations (16 billion billion). But if someone happens to be sniffing the traffic

during a valid knocking sequence, they'd know the combination.

Steve: Uh-huh.

Leo: That's kind of a problem. What if something like the old PayPal football were used? The ports knocked would be pseudorandom depending on a preshared key, you'd salt it with the key and the time of the day, let's say. What do you think? Jason in Rochester. I like that.

Steve: Yes. Well, first of all, I'm not a fan of port knocking.

Leo: Oh, really. Okay.

Steve: Yeah. Well, it's so prone to problems. First of all, the Internet is known to deliver packets out of sequence. That is one of the big, you know, big things that the TCP protocol guarantees is that, at the application layer that is above the protocol, everything will be in sequence. So, for example, when you download a file, you're just downloading - you set up a connection, and this binary blob comes through TCP. Well, you want to know that all of the chunks of the file that you receive are in the right order. So the application doesn't have to worry about that at all because TCP uses sequence numbers specifically to make sure that the packets that are being reassembled at the receiving end are in the proper sequence. And in fact, packets often arrive out of order, and then the TCP protocol will hold an out-of-order packet, waiting until a missing packet to fill in the gap comes along.

So here we have a problem with port knocking because what this means is that, if you send packets out to remote ports, you have to wait a long time to make sure that that packet got there before you send another packet. And if you send them too quickly, you risk the chance that they will arrive out of order since routing can often do that on the Internet. And in that case you've got a bad knock. So then you couldn't connect, and you don't know why. So you do it again. And maybe you luck out this time. I mean, it's just - it's error prone. The first patent actually that I already have pending for CryptoLink's technology is an invention which solves this problem completely. And it does it in a single packet. And it's very cool. And we'll talk about it when I can. But aside from the fact that port knocking has these problems, it's nice; and it's, I mean, it's clever.

So what Jason has suggested is, well, first of all he highlighted the other big problem. And that is that, if you have a static knocking sequence, which is the way most systems are configured, then you are protected from somebody randomly trying to access a service. So what the port knocking does is, if you knock in the proper sequence, by sending, for example, in Jason's example, four packets to successively different ports, in the proper sequence, something is monitoring that and will see that and go, oh, that's the secret knock, and then will open a port to allow anybody incoming.

Now, hopefully, if it's clever, it will only allow the same remote IP as the packet's source IP. So you've opened a port that is still filtered only to accept incoming traffic from you, the source of the knocking packets. Many systems don't do that. They open it for everybody, and that's, again, not as much security as you could have. But anyway, that's what the knocking does is open a port that you are then allowed to use to access a

service that's been protected until then.

So the problem is, as he points out, if your connection is being monitored, somebody eavesdropping who, for example, knows you're using port knocking, if you have a static knocking sequence, all they have to do is repeat the knocking sequence, which they captured from the wire, and the port will open for them. And then they're in.

So his notion was use a port-knocking sequence which changes. Now, time of day is a possibility. But there's actually a better approach. And it's - which can function very well. And it's a little more like the way the VeriSign credit card works, where it's a known sequence rather than a known time. That is, there is a counter which is encrypted with a secret key. So if you take a counter, and you run it through - say a 128-bit counter, and you run it through a 128 symmetric block cipher, you're going to get out 128 bits which are pseudorandom. Every time the counter counts up, these bits are going to change to something different. So you then - you take that 128 bits, and say you just took the lower 64 for our example. So those 64 bits give you the - you break it up into four groups of 16, which give you the port you want to knock on. So the receiver would have a synchronized counter so that it's able to anticipate the knocking sequence that comes in.

The problem is that you might have a packet dropped. And that's another problem, is that the Internet not only resequences packets, it has a complete freedom of dropping packets at will if any router's buffers are too full along the way. So many different things could cause this knocking sequence to have a problem. Meaning that these counters would get out of sequence. The beauty of this approach, though, is that if you sent two knocking sequences in a row that got through without any packet loss or reordering, and there's always a problem with port knocking, but if you did, then the recipient would be able to decode the knocking sequences back through the secret key and see that, even if the counters were out of sequence, then what happened was two successive counts. That is, the knocking sequence, when decrypted, turned into a count, and then a count plus one.

Well, the only way that's possible is if the other guy who sent the packets had the same secret key. So you've authenticated yourself independent of time and independent of bad knocks and their counters being desynchronized. It would allow multiple users to authenticate through the same system, if they had counters in different states. So it's a cool way of using two successive knocking sequences to prove that you know the secret key, even if you don't know what time it is, that is, you're not timer based, or if your counters are in different states. So with the limitations of port knocking, it is possible to come up with something that could not be sniffed.

Oh, and the other thing you do is you never accept a knocking sequence that you've seen before. That is, this counter counts up for every knocking sequence. There would never been an instance where you would get an obsolete knocking sequence. So you only allow them to move forward in time, much like the football does moving forward in time, or the counter in the VeriSign credit card does. So it's possible to make it work. It's just tricky. And because of the problems of packets not arriving reliably or arriving out of sequence, the whole knocking thing I don't think is a great solution. And there are better ones.

Leo: That's great to know.

Steve: Yeah.

Leo: Yeah. I wasn't aware of that. Last question, our Biometric Abuse Question of the Week.

Steve: Or story.

Leo: Story of the Week. Michael OConnor of Oswego, Illinois shares his Biometric Abuse Story of the Week:

Dear Steve and Leo, I'm flabbergasted. Like many married couples, my wife and I each have our own checking account in addition to our joint account from which we pay our bills. Every couple of weeks each one of us goes to our respective bank, withdraws some cash from our own account, and deposits it into our joint account so as to have funds immediately available. We've done this for a couple of years. There's never really been a problem until today. Since my wife wasn't feeling well, I told her I would handle the

transaction for her. All she would need to do is write out a check to me drawn on her account. I'd take it to the bank to cash it.

When I pulled up to the drive-thru, where I presented the check, the teller asked if I had an account with them. I said no, but mentioned my wife did. The teller said I would need to come into the branch in order to complete the transaction. I wasn't pressed for time, and although it was a slight inconvenience I was happy to comply. When I got into the bank I approached the teller window, mentioned I needed to cash a check drawn on their bank. The teller asked me if I had ever cashed a check with them before. I informed her that I had not. She said that in order to cash the check, mind you it's drawn on the bank I'm attempting to cash it at, she would need to see ID, but would also need to scan my fingerprint.

I can only imagine the look that I must have shot back. "What?" I asked. "You need a scan of my fingerprint before you will cash this check? I won't do that." I guess I caught her by surprise based on her response, where she assured me that the only way they would cash the check is if I provided them with my fingerprint. I asked to speak with a manager, and the manager informed me of the same thing. It's bank policy, and no exceptions would be made. I asked to speak with her boss, who a little less politely told me the same thing. To be honest, they actually made me feel like I must be hiding something if I wouldn't let them scan my finger. I shared with her my opinion of the bank's policy and quietly departed, feeling emotionally mixed between pissed off and violated.

Over lunch I continued to ponder what had just happened, and I couldn't believe that what the bank was doing wasn't a violation of my privacy. After all, I had provided them with a state-issued ID. And there was no reason to believe I was committing a crime. Therefore there was absolutely no reason for them to need to access my biometric data. I'm not sure how far I'll take this impromptu crusade of mine (EFF, are you listening?). But I thought I'd start by sharing it with you guys. After all, when it comes to computer security and privacy, you guys are one of the first resources I turn to. Thanks for letting me share my tale. Best, Michael OConnor, Oswego, Illinois. It sounds like this might be something since 9/11 would be my guess.

Steve: Well, first of all, I completely agree. We've talked about the whole issue of biometrics. It's a mixed blessing. The blessing is mixed because biometrics are not something you can change. You're able to change your password anytime you want. But your fingerprints are you. And that's what makes them so valuable, for example, in criminal forensics is people tend to leave fingerprints behind, and that uniquely identifies them as being them. The problem is, electronic security is basically repurposing something which I think has substantial value. Now, I can, I mean, when I renewed my driver's license last time, you know, I've been - I don't drive that much, so I have not been on, literally, on the police radar for a long time, and I think I went however many the maximum years is you can go. And then they said, well, look, you know, we've extended you automatically as long as we can. You need to come in now, make sure - you need to let us see you because we haven't seen you for so long, and make sure your eyes still work and so forth.

So I was annoyed that the California DMV wanted my fingerprints. And it's like, okay, well, fine. I mean, that's - I have no choice there. But I would argue, as Michael did, about casual disclosure, like we've talked about the Disneyland card gate, casual disclosure of biometric data. We don't know what the bank is doing. I don't trust the databases in general to keep this information from leaking. We're constantly hearing stories about this data getting loose. And I would hate for my fingerprints to be surfing the Internet without me. That's just - that's creepy.

So anyway, I really do, I really think that, I mean, first of all, as he says, he's got a valid state ID. It's his wife's check that he's cashing, so she's got the same last name as he does. I mean, this really does seem like a rigid policy. And I've got to wonder, I mean, you know, obviously this is not a huge check. If this is what they do every couple weeks for spending money, it's probably a few hundred dollars. It's not, you know, $10,000. So I don't know, it really does seem like a bad policy.

Leo: I think we're seeing it more and more everywhere. But I think he raised a really important issue, which is that you get one and only one.

Steve: Yeah, exactly. And if it gets loose, it's loose. I mean, yeah. And just casually, cavalierly scanning them into some database is like, oh, I don't know. I mean, I would like a firm explanation from the bank. And I don't think Michael got one. All he got was "It's our policy, sir."

Leo: Well, presumably, if she had mailed me the check - you know, it's because he's cashing it at the window. I mean, if he deposited it, they wouldn't require - that's what's bizarre about it.

Steve: Right.

Leo: I mean, that check is something that the bank honors, and transfers money on the basis of, all the time. But it's just because he's asking for cash at the window, I guess.

Steve: Yeah. And, you know, I guess I question it being anything relative to 9/11 because I don't think everybody else is doing this.

Leo: Unless it was a huge check.

Steve: Yeah, exactly.

Leo: I don't know. That's, yeah, very bizarre. Well, Michael, let us know what happens. That's not good. And Steve, we've come to the end of another thrilling, gripping edition of Security Now!.

Steve: Number 222. 223 is going to talk about this recent bad news for SSL, the latest, greatest version of SSL and TLS, and how a hacker has found a way to insert plaintext into a transmission during renegotiation of security credentials during an established connection. That's not good.

Leo: Unh-unh. Wow.

Steve: I mean, basically what it means is all clients and servers will have to be updated. This is not like something you can work around. This is the SSL protocol, or TLS I guess I really ought to start calling it because that's the official name. But it needs to be scrapped. Version 3, or TLS 1.0, we need to go to 2.0 of TLS. That's the solution. And it necessitates SSL being changed everywhere.

Leo: Yeah.

Steve: It'll certainly be - SSL is very tolerant of backward compatibility. So new versions will come out that close this hole, which will still run back versions. But there'll be an impetus for everyone to update their browsers and their - actually it's probably the operating system level. So it'll be another patch that'll come out from Microsoft and the various Linuxes and Macs and so forth that'll fix this.

Leo: My friend, we have come to the end. People, there is much more if you go to GRC.com - 16KB versions of the show for the bandwidth-impaired, Elaine's great transcriptions, all the show notes. And of course GRC's the home of SpinRite, the world's best hard drive maintenance and recovery utility. You must have it if you have a hard drive. And all of Steve's free stuff, too, some great, wonderful free programs like ShieldsUP! and Wizmo and on and on and on. Just go to GRC.com, Gibson Research Corporation, for more information. Steve, once again we have been nominated for a podcast award in the technology section. You won, I think, last year or the year before. So...

Steve: Yeah. We cheated, though.

Leo: What, we asked people to vote for us.

Steve: Yeah. We used our listeners, who just overwhelmingly came out. And we just swamped everybody else.

Leo: We shouldn't do that. So I won't mention that if you wanted to vote for a show on the TWiT Network, that you would go to PodcastAwards.com and, starting November 13th, cast your - I wouldn't say that because, you're right, that wouldn't be fair to the other guys.

Steve: It wasn't even a contest. Our listeners stood up for us, and it was game over.

Leo: Yeah. So whatever, you know, just forget that I mentioned that. We're nominated - a lot of shows are nominated this year round. So...

Steve: You mean there's, like, other people than TWiT?

Leo: Oh, yeah, yeah.

Steve: Awww. In that case, everybody, let's go swamp everybody else.

Leo: Yeah, I didn't mention that part. There is a competition here.

Steve: We don't want you to lose. We want one of the - we want the TWiT shows to win.

Leo: Nah, that's fine, I don't care. Hey, Steve, it's great to talk to you. We'll see you next week on Security Now!.

Steve: Thanks, Leo.

Leo: I don't care if we win.

Steve: It's fun to win.

Leo: You're nominated for Best Technology, along with - I think you're the only - oh, no, Floss Weekly is also in that category. And there are some categories where we're competing against each other. It's silly.

Steve: Yeah, actually there were a lot that I noticed that were like that last time. I mean, where it's like, wait a minute, you know, I mean, I...

Leo: Basically all of them are nominated.

Steve: Yeah.

Leo: All right, my friend. Thank you so much.

Steve: Absolutely. Talk to you next week for a propeller-spinning episode on SSL.

Leo: That'll be very interesting.

Steve: Yeah, absolutely. Have a great week.

Leo: Thanks, sir. Take care.

Steve: Take care. Bye bye.