Listener Feedback #26

Description: Steve and Leo discuss questions asked by listeners of their previous episodes. They tie up loose ends, explore a wide range of topics that are too small to fill their own episode, clarify any confusion from previous installments, and present real world "application notes" for any of the security technologies and issues they have previously discussed.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-114.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-114-lq.mp3

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

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

This is Security Now! with Steve Gibson, Episode 114 for October 18, 2007: Your questions, Steve's answers.

Time for Security Now!, ladies and gentlemen, our Podcast of the Year in Science and Technology...

Steve Gibson: [Trumpeting]

Leo: ...with your host, Steve Gibson. Mr. Fanfare, Steve Gibson. Hey, Steve.

Steve: Hey, Leo. Somebody saw the, I guess a streaming copy of me accepting the award at the Expo.

Leo: Oh, I'll have to find it.

Steve: And they wrote a note saying, hey, it was really great to see you accepting the award. And, you know, I did, you know, my acceptance was tantamount to simply thanking our listeners for making it happen, since they did.

Leo: They made it happen.

Steve: Yeah.

Leo: Well, we're glad that you won. And I always - I'm going to, from now on, for the next 52 shows we're going to acknowledge that.

Steve: [Trumpeting]

Leo: Use it all we can. Today it's a Q&A episode, which means we've got some great questions from you, the listeners, and some...

Steve: Really, really good questions this week.

Leo: Oh, good.

Steve: I think this is going to be an exceptionally good Q&A. I had some time to do some research into some great questions that were asked. And so this is, I think - I'm excited about this week's episode.

Leo: Excellent. Let's get to any errata from previous weeks before we get going here.

Steve: I have one big piece of errata. A surprising number of people wrote to ask, not only in the feedback form, and I wanted to remind people about GRC.com/feedback, where they're able to communicate to us their wishes and desires and show ideas and questions and comments and great ideas. In fact, we have - the last Q&A, number 12, is this week's winner of the Brilliant Idea Award.

Leo: We're going to do one of those every week now, huh?

Steve: Well, whenever there is one, you know. We don't always have brilliant ones. But this one was just like, oh, yes, of course.

Leo: Of course.

Steve: But so I just wanted to remind people of GRC.com/feedback. The thing that we were reminded of, I was reminded of, not only here but even in our newsgroups, people were saying, hey, Steve, you promised to track down that question that a listener had, I think it was in Episode 112, which would have been of course two weeks ago, where someone said that in trying to download the virtual debit card software from PayPal, the link was broken for them. And they later tracked it down to the fact that they were using a hosts file to block their computer's access to DoubleClick.

Leo: Oh, yeah. And why was DoubleClick involved in the transaction at all?

Steve: Exactly. Well, and so people have said, hey, you said you were going to check back on that and let us know. Okay, well, I did. And it's as bad as it could possibly be.

Leo: Oh, no. Oh, no. Oh, no. Now, who did you go to to find this out?

Steve: I just went to the website and did a little skullduggery. On our show notes page for this week's episode, for Episode 114, which is going to be chockful of links and goodies of all kinds, I've got a picture showing where I'm hovering my cursor over the download button. And you can see the URL which Firefox shows in the status line at the bottom of the screen, that this is actually a DoubleClick.net URL. So, and I want to explain the implications of this in detail. So we're going to do a whole show about it. Number 117 is going to be PayPal and DoubleClick.

Leo: I suppose we should try to get somebody from PayPal to come on and explain themselves.

Steve: We can try. And, you know, I've tried to use PayPal's tech support or product support in the past, and it's virtually nonexistent.

Leo: Well, we can approach them as the press.

Steve: We'll do whatever we can. But, yeah.

Leo: I'll make a note of that. I'll see if we can get somebody from PayPal. Because I want to give them a fair chance to respond.

Steve: We certainly should hear their side. Because I want to explain what it means when you redirect a web browser through another site because of the - I'm going to describe exactly the way information flows and, I mean, really, how frankly unconscionable this is. I'm noticing it also, for example, when I click on links sometimes in Google, I get an almost blank page with a little link up in the upper left-hand corner that says "Click on this if you're seeing this." And it's because I'm running with browser security cranked up, and a redirect through - I don't remember what the link is. It's like Mediaplex or something. It's another tracking service, and I'm being tracked when I'm clicking on just a search link, not even an ads link or Google ads or anything, it's just a search link. And it's like, oh, well, yeah, we just wanted to keep track of you, Steve. So anyway, I want to really talk about this in detail because it's something which is certainly a privacy concern, and...

Leo: Yeah, no kidding.

Steve: ...really, really troublesome when a company that you have an account relationship with, like PayPal, is apparently allowing - well, anyway, we'll go into it in Episode 117. And we'll make a concerted effort to see if we can get someone from PayPal to defend themselves.

Leo: Yeah. Wow, that's really a surprise. I want to mention before - there's another news story that I think is quite interesting, just broke today. You probably haven't had time to see it. But Apple has announced that they're going to allow third-party applications on the

iPhone.

Steve: Oh, good.

Leo: Hallelujah. In February. But here's the reason I'm mentioning it on Security Now!. Steve Jobs has a post on Apple.com/hotnews. He says it's going to take till February because they're going to release an SDK, which means that anybody will be allowed to develop, anyway. I don't know how Apple will certify, and I guess that's what he's going to address here. Because he says, we're trying to do two diametrically opposed things at once: provide an advanced and open platform to developers; while at the same time protect iPhone users from viruses, malware, privacy attacks, et cetera. This is no easy task. He says, some claim that viruses and malware are not a problem on mobile phones. This is simply not true. There have been serious viruses on other mobile phones already, including some that silently spread from phone to phone over the cell network. As our phones become more powerful, these malicious programs will become more dangerous. And he says the iPhone is a highly visible target, and so he wants to make sure. I think that that's, you know, in retrospect, I've been disappointed with how slowly Apple has put third-party apps on the iPhone. I mean, it's something we expect.

Steve: Well, now, there were already some hacks that were out.

Leo: Yeah. But you saw Apple's response, which was to delete the - to make it impossible to put third-party apps on, delete any third-party apps you installed. And if you'd gone farther and unlocked your phone, they bricked it. This says basically, you know, there will be a legitimate channel for putting third-party apps on the phone.

Steve: I'm not an iPhone user because I'm over on the Verizon/Sprint side with a CDMA link because I like to use the EVDO for wireless broadband, which is about four times, no, maybe even more, more like five times faster than the Edge network that AT&T and Cingular offer.

Leo: But also I suspect because you like your third-party apps.

Steve: Well, I do have an iPod Touch because I just had to have - I had to play around with that cool touch screen interface and the latest iPod. I'm a media guy, so I like to create video clips and carry them around. The problem is, there's no possibility for me to enter my WPA key...

Leo: Oh, yeah. Good point.

Steve: ...through, I mean, I would like to also be able to use it a cute little Wi-Fi web browser on...

Leo: You can on the iPhone. You can use WPA.

Steve: Well, no, I mean, the iPod Touch has it. But I can't enter my key. Have you seen my key?

Leo: Oh, that's right, because there's no keyboard on the iTouch.

Steve: Well, no, there's a keyboard.

Leo: There is?

Steve: Oh, yeah.

Leo: It's just too long for you to enter accurately? Oh, yeah, because they show stars the whole time.

Steve: Exactly. They show stars, and they want you to put it in twice.

Leo: Yeah, no, you're right.

Steve: And there's no...

Leo: And there's no cut-and-paste.

Steve: Exactly.

Leo: Yeah. That's a good point.

Steve: So I'm hoping that, if some third-party apps come out, there'll be something like a Notepad or something that'll give me some copy-and-paste functionality on the iPod Touch. Then I'd be able somehow to, like, dock it and move the key from iTunes into Notepad into the iPhone, then do a copy-and-paste to get it into my WPA because, you know, I use one of my own passwords from GRC.com/passwords. And there's no way I could type that in.

Leo: Well, that's frankly why I use simpler, shorter passwords, to be honest with you.

Steve: Well, I'm just pissed off. It's like, that's wrong that you can't - there's no way to, like - there's just no way to shoehorn a long string into that iPod Touch at the moment.

Leo: But surely you approve of his notion. And by the way, when he said third-party apps on the iPhone, iTouch, too. Does include the iTouch.

Steve: Yay. Okay, good.

Leo: But surely you approve of his point of view that we want to make sure it's secure. And I think that in a way that's admirable that they're going to do it before, upfront, as opposed to the way everybody else does it now, which is as an afterthought, security.

Steve: Well, I mean, he would have to - he. They, Apple, would have to come up with something like a Java environment where the apps are running in a completely sandboxed environment. I mean, unfortunately he's talking about a medium that viruses and malware would love to exploit. I mean, connectivity is the way malware and viruses thrive. And, yeah, what he's talking about doing is extremely difficult. So it's a good thing to say, and I'm glad they have this goal. But I think people who install third-party apps are going to have to - let's hope that they design it in such a way that there's all kind of hoops you have to jump through to manually install the app so that things like worms are not able to slip in through some back door.

Leo: You know what he points out in the same release, he said, look, the way Nokia is doing it is you can only install an app if it's digitally signed and approved with a real certification. I don't know if it's approved by Nokia, but digitally signed, demonstrably, something like that. And that makes - that would be a doable - let me see. Let me read what he says.

Steve: You just go to the Hong Kong Post Office and get...

Leo: Well, traced back to a known developer is what he's saying.

Steve: Oh, okay. So that would mean that developers would have to register with Apple in order to have the right to sign code that would then be obviously revocable if they didn't do the right thing and so forth. So people would grumble that that was creating a bar to entry. But, yes, I'm sorry, it's going to have to be - something's going to have to be done in order to keep this from just getting out of control.

Leo: My suspicion is he's setting it up that they'll sell it in the iTunes store only, and you'll have to go through Apple's certification process to get it in the iTunes store.

Steve: Oh, okay. I'll buy Notepad that way, yeah.

Leo: I think in a way that's a good solution.

Steve: Anything in order to get my WPA key in my iPod, into my iPod Touch.

Leo: Duly noted. We need Steve's WPA. Before we go on, do you have any SpinRite email, any...

Steve: I do, I have a fun short story from a guy named Stewart Leathwood, who said, "You're going to love this!" And he said, "I work as a network support professional for a small government organization." He says, "Because of our size, every penny is precious. We purchased SpinRite to repair some PC hard drives, which it did famously. But the story I want

to relate to you is more unusual." In fact, the reason I chose this is we were recently, I guess it was like in - it was two weeks ago I was answering a question about RAID and SpinRite. And he says, "We have 1.4 terabyte Snap Server network-attached storage device we use for one-site backups. Someone unplugged it, and the RAID array became corrupt and the disks unreadable. So I decided, what the heck, let SpinRite take a crack at them. I installed the IDE drives that needed repair in a PC and ran SpinRite on the corrupt ones. It took hours, and one took days to process." He says, "It fixed each one in turn, and I was able to rebuild the RAID array perfectly. Thanks for saving our bacon. SpinRite more than paid for itself."

Leo: Wow, that's...

Steve: So SpinRite on a RAID array, yup.

Leo: That just proves you can do it.

Steve: Yup, exactly.

Leo: So, let's see. I have in my hand, Mr. Gibson, 12 fantastic questions. Are you ready to answer them?

Steve: They are good ones this week.

Leo: All right. You picked them, so you ought to know. We should say that. I don't pick these questions. I'm not trying to stump Steve. These are - and nor is Steve trying to softball himself. These are questions that are interesting, that seem to be commonly asked.

Steve: Well, and they are, and many of them send me off to go do some research. I mean, there are questions I don't have answers to. So I go, oh, that's really interesting, I had never heard about that, and off I go.

Leo: That's why I love doing the podcasts, and I love doing the radio show. This is the best way I've found to keep up to date on what's going on. Mike Vona of Carolina, Rhode Island. I don't think there's a Carolina in Rhode Island, but...

Steve: That's where he said he was.

Leo: I believe you. Maybe there is. I grew up there. Gee, that's a small state. It's not like there's hidden towns. But anyway, wonders about UPnP and MS Home Server. That's the new Microsoft Home Server. I've been listening since Episode 1, and I, like many, can't get enough, says Mike. I recently started listening to Windows Weekly. I'm enjoying that, as well. My question stems from a recent episode of Windows Weekly, Episode 31. But he wants to ask you because UPnP comes up so often. In this episode Paul and I were talking about Microsoft's Home Server, and that it now uses UPnP. In fact, to work it needs UPnP turned on in the router. In fact, I've seen, Steve, just parenthetically, a number of articles saying is Windows Home Server really secure? Are we going to let, you know, this seems a

little risky.

Steve: Oh, yeah.

Leo: Mike says, I'll cave in to this. I know Microsoft has had their way about them, that it either is their way or the hard way. But couldn't you enable UPnP while the server is configuring what it needs, and turn it back off? Once something in the router is set via UPnP, is it not permanent, even if the router restarts and power cycles? Okay, this is actually a general question, not just having to do with WHS here.

Steve: Right, although, well, okay. There's a couple things. First of all, one of the most distressing things about Universal Plug and Play for router control is that most user interfaces do not show any of the configuration which has been done through the Universal Plug and Play interface. To give our listeners a little bit of background - although I harp on this all the time because it just scares me to death, and I'll just bet you we're waiting to see some real abuse of this pop up. The idea is that the router presents a server to the internal network. And the way Universal Plug and Play protocol is designed, it's possible for any computer on the local area network to send a broadcast out saying, hey there, I know about Universal Plug and Play. Who wants to play? And all the devices that are supporting Universal Plug and Play get the IP address of the machine that sent out this broadcast question. And they say, hey, I know about Universal Plug and Play. Hey, I know about Universal Plug and Play. And so essentially it allows the network to discover itself. It's handy because one of these days Sears is going to be offering a Universal Plug and Play refrigerator, and you plug it into your network or just raise the antenna in the back. And it'll be part of your network. And so the concept is that it's a self-configuring network. My concern is, when your neighbor is able to hack into your refrigerator...

Leo: Put some malware on it.

Steve: ...that can be a problem because Universal Plug and Play in the first incantation - which is all we have so far, although I have heard that there is a second generation on the way that apparently adds security. But in the first incantation there is none. So there is no way to prevent malware from having access to, in this case, literally the entire configuration of your router. It can do anything it wants to with your router with Universal Plug and Play turned on. And because the configuration changes made, like creating static port mappings back from the outside into your network, do not show up in the user interface, you could be a responsible administrator of your router, look to see on those administrative web pages that the router publishes what's going on, and see nothing, even while there is static port mappings that have been created by the Universal Plug and Play interface coming back into your network. So it's a concern.

Now, the good news is, and the reason I wanted to specifically address this question relative to the Microsoft Home Server, is in fact it only needs, that is, Microsoft's Home Server only needs three ports mapped through the router, and in fact may be able to get away with only two. That is, it is a web server, so it needs port 80 and port 443 mapped through. It also needs what they call a "Remote Desktop Proxy" port. Now, Remote Desktop, as we've talked about it in the past, normally runs over port 3389. But in this case the Home Server runs a proxy for that, meaning that it's receiving these requests for the outside and then is relaying them internally to within your network to other machines, for example, or the Home Server itself, that may have, in fact has to have Remote Desktop running on port 3389. But the proxy runs on 4125.

So you do not need Universal Plug and Play enabled at all. And again, you know, for all the reasons I've just finished talking about, I strongly recommend that people don't enable

Universal Plug and Play. However, you can manually administer that port mapping. Simply send ports 80, 443, and 4125 to the IP of your Home Server running on your network, and you're good to go, with the strong caveat that here once again we are exposing Microsoft servers, that is, Microsoft services, to the public Internet. And I don't think there's ever been a Microsoft service that didn't have horrible security vulnerabilities, at least initially. I mean, even their web servers, their commercial web servers, IIS had horrible problems with them. And all of the services that Microsoft has traditionally exposed through Windows have had problems, which is why only after we got the firewall running in Windows Service Pack 2 of XP did all these problems quiet down because now they were - those services were hidden behind a firewall.

Well, what we have to do, and if we're going to expose this Home Server to make it available on the Internet, is break that rule and make those services available. So it's something to be aware of. But at least you don't have to have Universal Plug and Play enabled if you do want to use Windows Home Server.

Leo: And did you - I'm sorry if I missed it, but did you answer the question, with UPnP, if you enable it during configuration, can you then disable it and have all those ports now be open that UPnP is opening? Or does disabling it turn off all the ports that it's configured?

Steve: Disabling, well, again, this does vary with routers because unfortunately one of the things that Mark Thompson, my buddy over at AnalogX, found out - remember he had 60 routers that he was messing around with recently...

Leo: One of which renamed itself to Nadine or something, but that's...

Steve: Venus.

Leo: Venus. That's another story.

Steve: Yes. He has discovered a huge variation in the behavior of Universal Plug and Play implementations.

Leo: I see, I see.

Steve: Part of it is that it's not a mature, well, there hasn't been time to mature the implementations. it's a new spec. A lot of routers rushed into having, oh, yes, we support Universal Plug and Play, too. So there's a huge variation in behavior. So it's impossible to say this is what Universal Plug and Play does. And specifically things like, what if I disabled it once some mappings have been made, do they remain, that's nowhere in the spec. Nothing talks about how the router should even behave in that case.

Leo: So it's unknown.

Steve: Exactly, it's undefined. So you could try it and see how it works. What will typically happen, however, is you disable the interface, and those mappings will remain until you reboot or recycle the power on the router, in which case then they are lost because they're not typically written into the firmware in the same way that your static configuration changes are.

Leo: Although I should point out that some routers, like my D-Link, you can save the configuration. And probably that would be the way to do is save it, and then when you reboot you can reload the configuration.

Steve: Can you save the Universal Plug and Play configuration?

Leo: Well, I mean, it says you could save - I presume it would save port forwarding that you'd done and that kind of thing. That's a good question. I don't know.

Steve: Yeah, I think probably not.

Leo: Oh, it wouldn't, okay.

Steve: Because it is regarded as sort of a software, sort of a dynamic, you know, the application opens the ports while it needs it. One would hope, for example, that when you shut down Windows Home Server, it would know to turn that forwarding off.

Leo: Oh, yeah, right.

Steve: So, uh-huh, so that it closes the router down again. It's just a bad thing.

Leo: Yeah, yeah. Well, and this is the problem. What's happening is people are being asked to do things that sysadmins did only a few years ago. You're running a network. All of a sudden you're doing very high-end network configuration stuff. And a sysadmin would lock it down and do it all by hand. But in order to make this stuff user-friendly so that any nitwit can have a home server, they're doing things like plug and play, Universal Plug and Play. But then, of course, that's not the safe way to do it. So I see the issue. They want to make it accessible. But there are some things maybe that shouldn't be accessible.

Steve: Well, and it would be one thing if Microsoft had a perfect security record. Dot dot dot.

Leo: I have to say they're getting better. We haven't had any major issues with Vista in a year.

Steve: Because the firewall is always on. I mean, no one even knows if Vista's services are vulnerable because there's no point in poking at them because they're hidden behind a firewall. I mean, it really is the firewall...

Leo: Boy, that was simple.

Steve: The moment that got turned on by default, everything changed.

Leo: It was so simple. Something we should have done years ago.

Steve: I asked them to, but...

Leo: Justin Van Viersen from Perth in Western Australia wonders about supernodes in Skype: I'm presuming that, when you forward a port through your NAT router to Skype, you are opening up Skype to become a supernode. Is this correct?

Steve: Okay. What Skype supernodes are is it's the way Skype deals with its inability with some NAT routers to establish a direct connection. For example, I have a very bizarro, nonstandard NAT facility at my end of our connection, Leo. So I've had to establish a static port forwarding through my NAT router, which is exactly what we were just talking about in the case of Windows Home Network Server, in order for you and I to create a direct link between our machines so that we get this really nice, high-quality Skype connection. Without that, what can happen is that we know that NAT routers are able to initiate connections outbound. That's the whole point. In order for Skype to work, we need two people, both behind a NAT router, need some way to link through their NAT routers. When that's not possible, Skype finds some other Skype user somewhere out on the Internet and says, okay, we're going to make you a relay, that is, a so-called supernode. In which case, both of these users behind NAT routers that are unable to establish a direct connection, they both call out and establish connections to this third party, to this so-called "supernode." And it acts as the relay for their traffic. Well, many people consider this objectionable, and I certainly don't blame them. It was one of the very controversial and annoying things about using Skype.

Leo: Well, Skype is peer-to-peer technology, and that's how it does it. I mean...

Steve: Exactly. The good news is, because of concerns, ever since v3.0 the Windows version of Skype has a registry editable setting that disables its use as a supernode. So it is possible to tell your Windows version of Skype, under no circumstances are you allowed to be a traffic relay.

Leo: Now, let me ask you a question because it's my understanding, as you said, supernodes are really used to provide NAT traversal. In order to be a supernode you have to be on an open IP, a public IP. You can't be behind a router; right?

Steve: That's probably the case. And I think that's why Justin's question was...

Leo: It doesn't really come up. You don't even have to do this registry modification if you have a router; right?

Steve: If you're behind a router, then, oh, I'm trying to remember whether...

Leo: It's my understanding - and you know what, Skype is not very clear about this. But it's my understanding that you have to be on an open IP address, unprotected IP address. You can't be behind...

Steve: I don't - I'm not sure that's the case because many NAT routers do behave themselves. And as long as the - and, you know, we've talked about NAT traversal before. There is some behavior in a NAT router where its natting is predictable. And so it could certainly be the case that even somebody, even a supernode behind or a version of Skype behind a well-behaving, easily traversable NAT router, it could accept incoming connections from people behind two non-NAT-compatible routers.

Leo: But remember, they're looking - what Skype, the Skype network looks for is - it's a small set of supernodes. They're not - not everybody becomes a supernode. And according to the research I've read, generally it's somebody who has a lot of bandwidth, although it doesn't take a lot of bandwidth to be a supernode, surprisingly, because you're really - you're not transmitting the data; right? You're just providing a handshake.

Steve: No, you're relaying all of the actual conversation data. Because that's what you have to do. The Skype server itself is able to perform NAT traversal handshaking in order to negotiate a connection between well-behaving NAT routers. You're actually the supernode. And, see, that's why the connection quality drops so badly is suddenly your jitter goes up, your latency goes up, because all of your actual conversation traffic is being relayed through the supernode, and it can be a lot of bandwidth.

Leo: But remember, they're using a lot of supernodes to distribute this.

Steve: Yes.

Leo: So you're not taking a whole call necessarily.

Steve: The other thing anecdotally I've heard is that, if you disable the setting for starting Skype up at boot, that also disqualifies you as a supernode. Which I...

Leo: Because they want you to be on Skype all the time for it to work.

Steve: Well, exactly. They want to assume - they want to know that your Skype is going to be generally available for supernode status. So again, I don't know this for sure, but anecdotally the reports of people that have looked at this closely say that, if you just turn off the autostart in Skype, that that'll do the job, too. But certainly if you're under Windows and you're using a Skype from v3.0 on, it's possible to disable that. I've got the information on the show notes for this episode. So anybody who - and I would recommend, unless you enjoy the idea of having other conversations relayed through your bandwidth, you may very well want to disable, especially our listeners who are privacy and security aware, may want to disable their handling of other people's conversations.

Leo: I have to send you a link to a study of Skype. See, apparently Skype doesn't really document this. They don't want anybody to know their secret sauce. A study done by a guy at Google, couple of guys at Google and a guy at Cornell, basically an experimental study of how Skype works. And they talk a little bit about bandwidth consumption as a supernode and so forth. It's a little dated. It's from 2005. But I think it probably gives you a good idea. See, nobody knows, so you just kind of have to test it. And they did a long-term test of how Skype worked.

Steve: Well, and one of the differences between Google and Skype is that Google will itself relay its users' traffic, and there is no notion of, with Google Talk, of a third-party supernode being asked to carry other people's traffic. Google's server does that itself.

Leo: Well, these guys point out the Skype founders came from Kazaa. That was their previous product. They understand peer-to-peer. And Skype was really designed from the ground up to be a peer-to-peer Voice over Internet.

Steve: Yeah, and it's free.

Leo: That's why it works. That's why it works. So we now - what we do, and your recommendation is, use a dedicated port. Will that prohibit us going through a supernode? Will that just give us a direct connection?

Steve: Well, and see, that's exactly what Justin was asking. He says, when you forward a port through a NAT router, the way you do that is you say I'm going to forward this port from the outside in. And then you tell Skype to listen on that specific port. And so that would clearly enable it to be a supernode because it would mean that anybody could connect to it.

Leo: It's now public, right.

Steve: Yes. And since you've configured your Skype client to operate in this fashion it's certainly able to disclose its configuration to Skype Central and open you up for being a supernode. So again, I don't know that that's the case.

Leo: So maybe I should turn that off.

Steve: Well, the other thing, Leo, is I'm betting - well, first of all, I don't run Skype at startup.

Leo: Nor do I. I turn it off unless we're doing this.

Steve: Yes. And so if it's true, and this was a - there was a study done by the guys at University of Waterloo in Canada that said - the CS department did some real looking at this. And they said when Skype is not running on startup, it does not adopt supernode status. So that's an easy thing to do. And again, in Windows we've got a registry tweak that will disable it for sure. But certainly it's the case, and this is why Justin asked the question, that if you did do static port mapping, anybody could connect to you incoming, and it would sure seem like it would qualify you otherwise for supernode status.

Leo: Here's a quote from 2003 from Nicholas Zennstrom, one of the principals behind Skype and also of Kazaa. And he said, without being too technical, each Skype client is always connected to a supernode. That surprises me. Any Skype client could become a supernode. The supernode is acting as a hub. Supernodes are always on routable, open IP addresses. When a call is set up, the established TCP connection with the supernode is used to signal that a call is coming. Dependent on the firewall status of the client, the data stream is set up either as UDP, if the firewall allows, or in worst cases as outgoing TCP,

which is almost always allowed. And we've talked about why UDP is better than TCP for this kind of stuff. If both clients are only allowed to do outgoing TCP calls, they're routed through another peer. I don't know if that says anything or not. They don't talk a lot about how it works, that's the funny part.

Steve: Well, and right off the bat, the first statement the guy made is wrong because you and I, I mean, I've looked at our traffic. I've run a packet sniffer on our connection. And there is...

Leo: There's no supernode there.

Steve: There is absolutely nothing.

Leo: There might have been for signaling, though. He's saying, remember, every VoIP call has two parts. There's the data carriage, but there's also the ring, the signal. And they might use a supernode for signaling.

Steve: Except that Skype Central is the one that manages presence, as it's called, does the presence management for all their...

Leo: We knew that because, when Skype was down, so were we.

Steve: Exactly. It's not like lots of little supernodes kept the job going, so.

Leo: Maybe he was being disingenuous. Finn, located somewhere in Sweden, asks about Amazon's S3 service and Jungle Disk. S3's amazing. That's Leo saying that. I finally decided to check out some offsite backup solutions for documents, pictures, et cetera. Amazon's S3 seemed like a good, well-known option for long-term storage. I chose Jungle Disk as a front-end GUI and backup software. Security-wise it all seems pretty good. The authentication to Amazon uses SSL. And when backing up, all files are encrypted on the fly with AES 256-bit encryption. But the question one has to ask, is this really safe? In an end-to-end connection manner I frankly don't care as long as the files are stored okay. But the issue for me is trust. What's your take on trusting a company with your precious documents in the long run?

Steve: Well, first of all, Leo, tell us about Amazon S3.

Leo: It's a really neat service, a storage service that Amazon inaugurated a couple of years ago. It's just pennies a gigabyte. It's very affordable. You can find out about it by going to Amazon.com/s3. S3 stands for Simple Storage Service. It was originally, I think, intended for developers. But anybody can use it with the right interface. I use it for storing - exactly how Finn is using it, as a backup solution. 15 cents per gigabyte per month. Data transfer is 10 cents per gigabyte. But it goes down the more you use it, obviously. There are a lot of people developing for it. And, you know, a lot of developers use it. For instance, I'll give you an example, Pownce. Leah Culver, who developed the great Pownce system, which is kind of a Twitter-style microblog, uses S3 for all her database storage. It's reliable. It's fast. Because it's Amazon they've got a lot of servers running. It's asynchronous, and it's

relatively cheap for that kind of storage. So, and then Jungle Disk is just a GUI interface for it.

Steve: Well, actually Jungle Disk has something in addition to that that I like. First of all, everything you just said I completely agree with. We've got an episode scheduled before the end of the year on this issue, on Amazon S3 and Jungle Disk.

Leo: Oh, good. Oh, great.

Steve: Because I want to really understand, and I don't yet from looking at the data that I've seen, if it qualifies as what I call TNO. You know, that's one of my favorite new acronyms. TNO stands for Trust No One. There are different reports from Amazon's tech support about whether or not they can perform lost key recovery. So the question is, I mean, I absolutely love the idea of very inexpensive off-site storage. And in fact, if I determine - and one of the reasons I want to do the research and share it with our listeners is, if I determine that this is true TNO, then I will absolutely sign up and subscribe. I mean, the pricing seems very reasonable. I would love this as a means for, like, moving all of my source code and project work and, you know, basically the crown jewels off-site. The other thing you get is you get, because it's centralized and it's Internet based, then if I'm roaming around with my laptop, I have access to that archive which is centrally located and is distributed and backed up and secure and all the good things you want in a robust off-site backup. But I want to absolutely know that no one can be sniffing on my data. Not only do I need a secure connection, which I'm sure is part of this, I know that already, that I verified. But I want to know that Amazon under no circumstances has the keys to decrypt the data that they're storing for me. This needs to be absolutely opaque from their standpoint so that there's no way for them to perform key recovery. I need to take responsibility for making sure I don't lose my keys. And if I do, I've got to be up the creek.

Leo: Well, here's my question. Is Amazon doing the encryption, or Jungle Disk? If Jungle Disk, which is not from Amazon, it's a third-party client, encrypts the data first, then uploads it, and that's my impression of how Jungle Disk works, using an AWS, it says - oh, I see. You can use an AWS secret key, which Amazon would know, or using your custom encryption key.

Steve: Yeah. And we need to understand what that's about. There's some dialogue in their forums where they talk about, I mean, which is encouraging, where they talk about if you were to change your key, then you wouldn't be able to read any data that you had stored before.

Leo: Just like any encryption key; right?

Steve: Well, but they have this notion, which I really like, of previous keys. So apparently you're able to update your key, and you move the current key into an old key repository. And if it sees that it's unable to decrypt the data stored with your current key, it checks your previous keys and sees whether any of them are able to read the data. So it's really positive looking. But the one thing I wanted to mention about Jungle Disk, just our listeners know - although, again, I haven't yet vetted them, I need to do this, and I'm going to - is that it creates - and I should also mention it's only $20 at the moment, and it's available in all three major platforms: Windows, Mac, and Linux. They even provide you with open source, their own open source that allows you to browse and download your software, I mean, they're really opening the kimono and saying, look, this is how we're doing this. We're hiding nothing.

But the coolest feature from my standpoint is it looks like a local drive. And what that means is I'm able to still use something like my own favorite backup program, it's called FileBack PC. And what I like about it is that it's incredibly sophisticated in terms of the rules I give it, in terms of, like, versioning. I can tell it that I want 25 back copies of source files, that is, anything .ASM, you know, for Assembly language. But I want no more than five of them kept per hour, and no more than ten a day. And this thing manages all that for me. And so, if literally, if it turns out that Amazon's S3 service, coupled with this very reasonably priced Jungle Disk, works and is secure, then I could literally tell FileBack PC that the Q drive is where it's supposed to maintain its archive, and all of this would be done offsite for me transparently. So...

Leo: This is done using WebDAV. And frankly there are many WebDAV interfaces to S3. So if you wanted to roll your own using TrueCrypt and WebDAV, it wouldn't be such a difficult thing to do, either. So if you don't trust Jungle Disk, do it yourself. I mean, Jungle Disk isn't open source, so you can't be sure what they're doing with that key, I guess.

Steve: They really seem like good guys.

Leo: They do seem - everybody's using it now. But again, Jungle Disk isn't required for S3. S3 is just a service. Jungle Disk is a commercial interface to that service. But you can roll your own, absolutely.

Steve: Exactly. And in fact S3 is sort of a UI-less service. It's a backend data storage facility. Although apparently there is some key management that they do. I just need to understand what it all is. And we're going to do a show to describe it in detail so we'll know how secure it can be.

Leo: You have to get - to use S3 you have to get an Amazon Web Services key. And that's a simple application process. And if you just use that, obviously they can recover that. So that's like a password, basically. So, yeah, I think, you know, I'm looking at a page, Elastic8.com, where they have free tools for Amazon S3. And it's a couple of pages' worth. There's a ton of good choices. It's just that Jungle Disk is so easy. That's why...

Steve: Well, I mean, I just love the idea that it looks like a virtual drive. There may be other solutions that do. But again, the guy that asked the question, Finn in Sweden, he aimed me in that direction. It's like, hey, you know, this would be very cool because it looks like a virtual drive, which means my existing backup tool could be told, not only keep a copy on my central RAID array and keep a copy remotely on our Level 3 servers where I do now, but also stick it off in Amazon. As long as it's really encrypted and they can't get to it, I'm really comfortable doing that.

Leo: Brad Fitzpatrick wrote, and I use this actually for our web server, a Perl script that encrypts and then backs up automatically as a cron job to S3. And that's what I use for my backup. And that's really a great, I mean, that's - if you're running a web server, that's the kind of thing you want. Just kind of happens every day automatically.

Steve: Nice.

Leo: Yeah, very cool. You do have to make sure you delete some of the older backups, or you really fill it up fast.

Chris Noble in Wellington, New Zealand wonders about a specific authentication scheme. Are you ready? In Episode 113 - yes?

Steve: Yes.

Leo: You are ready?

Steve: I'm ready.

Leo: I don't want to go too far unless you're ready.

Steve: I was born ready, Leo.

Leo: I heard that. In Episode 113 you mentioned the risks of having a keylogger on one of your staff members' laptops - we were talking about remote authentication and how to do that right - thus making a password login via the keyboard potentially risky. What about a solution like that used by F5 Networks for their FirePass remote access solution? It's a little graphical virtual keyboard on a login page. You don't use your keyboard. You click the letters on the screen to enter a username and password. The key thing is - no pun intended - the keyboard randomly jumps around the screen after each mouse click. Oh, this sounds so annoying.

Steve: So you're chasing it around.

Leo: Oh, Jesus, it's annoying. So that not even a mouse logger can log the pixel position of the click on the screen for use again later. This sounds like a good way to do it. In a way, this is providing a one-time unique entry process - it's a good point - for something the user knows, as opposed to the same entry process via password text in a form, a field in a form. I'm interested in your comments on this form of logging in since it doesn't use a keyboard. It is immune to keyloggers; right? And it's also used by the company I work for to gain remote access. How interesting. Boy, that's secure.

Steve: Well, it's an interesting idea, and I wanted to bring it up because this relates to the topic we're right in the middle of discussing which we opened last week and we're going to finish with next week, and that is the simple but straightforward and very secure approach I developed for allowing GRC employees to have secure roaming access to GRC. And some of the questions at the end of this show are specifically about that, sort of as a lead-in to next week.

But the problem here - first of all, this is better than, for example, an onscreen keyboard that did not jump around. But the test is having somebody know absolutely everything about what you're doing. That is, it does sound like the password that the person is clicking on is the same every time. And that's the weakness of this system. Even though the keyboard's jumping around, the presumption is, I mean, the high bar that we set is an attacker would have full information of what you're doing. And that would mean they would be able to see the screen and figure out what it was you were clicking on, and then reverse-engineer the password so

that they'd be able to click on the same password, even though it would be occurring in different locations on the screen.

So, yes, this is better, certainly, than having a static onscreen keyboard. And obviously someone is concerned about the mouse click positions being recorded. That's what this is designed to bypass. But it still suffers from the weakness that the same password text is being entered each time. And that weakness is something that can be avoided. And that's the topic for next week's Security Now! episode.

Leo: Ah ha. Iain Cheyne in London was shocked - shocked, I tell you - to learn of the importance of the Firefox Master Password. Did we talk about that?

Steve: I don't think so.

Leo: He said, I had no idea - I use it. Now I'm worried. I use it. I had no idea about this potential problem with Firefox. You should tell people about it. He now quotes a page at Ghacks.net. Make sure you set a master password, it says, in Firefox. I presume you have read this page and have something to say about it, Mr. Gibson.

Steve: Indeed I have, my friend.

Leo: So let me just mention, the Master Password at Firefox I use because Firefox will remember passwords for you. And you can say, on that page where you turn that on, use a Master Password. And unless you enter the Master Password, nobody can then log into those pages without it.

Steve: Actually it's a little bit worse than that, and that was what Iain wanted to bring up, and the reason I wanted to share it with our listeners, is to make sure that people are aware that, if you use Firefox's remember-this-password or username and password for the site feature, if you use that, anyone can come to your copy of Firefox and click on Tools and then Options, and then click on Show Passwords, and Show Passwords again. You are then prompted with an "Are you sure?" dialogue box. If you click Yes, it then shows you, in the clear, the website URL, your username and password, right there in the dialogue box for anyone to look at.

Leo: Well, there you go.

Steve: And of course he makes the point that, if you reveal all your passwords to someone, or, that is, inadvertently do so, they could look at, oh, look, it's always the same. Shame on you. Or it's like, oh, look, here's how they're coming up with passwords. I mean, they could reverse engineer your password scheme if you had taken our advice from, I don't know, Episode 2 of Security Now!, whenever it was we were way back then talking about a personal password policy or procedure or whatever, protocol, in order to figure this out. So I thought it was worth bringing up. What Iain was saying was that it is, for the privacy of your passwords, you really need to set that Master Password on Firefox, or anyone who comes along can see what your passwords are.

Leo: And in fact I'm looking at it right now, and I see - because I turn it on, I had to enter it twice before I could see the passwords. And that's good. In fact, it's a little bit of a pain.

You might be tempted to turn it off because every time you launch Firefox you have to enter that password. But I think that's a good thing. So that's good to know. All right. Good thing to mention. I should mention that...

Steve: Yeah, and in fact I think that it is exactly for that reason, the fact that you launch Firefox, you've given Firefox the ability to log you into websites. You definitely want Firefox to harass you for the password to prove that it's you sitting in front of it.

Leo: Because I do. I mean, I let it log into everything, all my secure sites, my banking, everything.

Steve: But it doesn't know it's you.

Leo: Not unless I tell it. Nathan in Huntsville, Alabama wonders about online password storage: I've been very interested in a password management solution for a while now. I even started building my own server-side program so that I could access my passwords everywhere. Oh, boy. But instead of reinventing the wheel, I thought I'd look around the tubes, see what I can find. I found something pretty interesting. It's called Passlet, Passlet.com. As far as I can tell, all the data between the browser and server is encrypted, as well as the data being stored. As a user I'm encrypting that data on my end, then sending it to them. The only problem I could see would be a hack into the server, a change of Java code. I'm just wondering what your thoughts on this might be. Passlet, is it safe?

Steve: Well, it's an interesting idea. It looks to me like these guys have done a good job. They have an FAQ where they understand the risks. One concern is that they're using JavaScript running in the browser in order to perform the encryption. It's like, well, his concern was could someone hack their server to change the JavaScript so that the code you're running essentially on your client is no longer doing what they intended it to be doing. And so, yes, that's definitely a vulnerability. So but otherwise they're encrypting on your browser, on your client, so that all of their storage is encrypted. And it's encrypted through the connection, although it also insists on being an SSL-secured connection. So that looks pretty strong. These guys look like they're doing a good job.

Now, one thing that they're not doing is they're not protecting, that I could see, from the possibility of a keystroke logger on people's machines. And for what it's worth, Leo, I actually, in the email that I read from our listeners, I hear a surprising number of confessions from people who have found keystroke loggers on their machines. It seems to me this really is a problem.

Leo: Well, it's one of the things, one of the three or four things spyware often does, or adware or whatever malware often does.

Steve: Malware, yes.

Leo: Hmm.

Steve: So there are a number of web-based services that are doing this kind of work. I, and we're going to talk about it next week, have a solution for the keystroke logger problem. And in

fact I've already written to the Passlet guys and said, hey, guys, I've got this solved, I'm making all the code public, and the whole system is public. Maybe you want to take a look at it as a way of further strengthening the login to your facility. Because, of course, anyone who then logs in has access to all of the keys to the kingdom. And you want to make sure that that, you know, any time we consolidate all of our information in one place, just like using Firefox to store our web page login, we're consolidating all that in one place, we really have to make sure that the entry to that is as secure as it can be.

Leo: You know, I should ask you as a follow-on, I use a thing from Zarate.org called SuperGenPass. And what it is, it's a JavaScript bookmark that sits on your browser toolbar, and it does a hash of a master password which you give it. And you can either give it to it and have it store it, or for more security you can type it in each time. And then it will hash together your master password with the site's domain name to create a unique password for that site. And it's a good password. It's long, and it's got all sorts of a goobledy gobbledy gook in it. I wouldn't try to remember it.

Steve: We know that's what a good password is.

Leo: But the beauty of it is you don't have to remember it or even store it anywhere because you can generate it again as long as you remember your master password.

Steve: I think that's a very good solution, Leo.

Leo: I use that, not for my banking, but I use that for semi- you know, sites like The New York Times or whatever. And that seems to work pretty well. Google "SuperGenPass," one word, and you can find that easily.

Nathan in - oh, I already did that one. That's the Passlet one. Let's go on to Kerry Merritt, who wonders about VME encryption: Have you guys heard of VME, victory marvelous encryption? I just made that up. I would just like your opinion on it. Their site is Meganet.com. They claim a million-bit key using a matrix created by a secret file, and then the program can be set to use a configuration file to change the type of matrix that is created. If you could, would you give us a quick thumbs-up or thumbs-down on it? I have it, and the configuration tool they have offered all kinds of stuff like a million dollars to anyone who breaks it. I always get worried when somebody offers a million dollars to anybody who breaks something.

Steve: Well, Meganet. So I go over to www.meganet.com. And they've got seals and certificates from the various government agencies and all kinds of FIPS qualifications and certifications and all this. And I go, okay, what? And one of them was a mention about a patent. So I thought, oh, that sounds interesting. So I click on that link and go over to a patent site. Filed almost exactly 10 years ago, on October 24th in '97, so in a couple weeks it'll be 10 years ago. And the patent was granted on April 17th of 2001. Would have been maybe more fitting if it was granted on April 1st. But it's an honest-to-God patent. It's Patent No. 6219421. And the patent, the abstract at the beginning of the patent, this is worth sharing with our listeners, says a data security method and apparatus that provides an exceptional degree of security at low computational cost. The data security arrangement differs from known data security measures in several fundamental aspects. Most notably, the content of the message is not sent with the encrypted data. Rather, the encrypted data consists of pointers to locations within a virtual matrix...

Leo: Oh, how weird.

Steve: ...in a large, arbitrarily large, continuously changing array of values. The encryption technique is therefore referred to as Virtual Matrix Encryption.

Leo: That's the VME.

Steve: There you go. Furthermore, the data security arrangement uses a very large key of one million bits or more, which creates a level of security much higher than any other existing method. The key is not transferred, but is instead created from a file of any size that is available on both the computer used to send a secure message and a computer used to receive a secure message. The term "virtual key cryptographic" is used herein to refer to techniques in which a key is recreated at a remote location from an electronic file without any transmission of the key itself. The file may be a system file, a file downloaded from the Internet, et cetera. A smaller transaction-specific key, e.g., a 2,048-bit key, is sent end to end and is used in conjunction with the very large key to avoid a security hazard in instances where the same file is used repeatedly to create the very large key. So, you know, decoding this, it sounds like you refer to some third-party file somewhere. From that file you somehow create this virtual matrix. And then the act of encrypting is finding strings in this virtual matrix. And then you only send pointers to the strings, rather than pieces of the file, and that's how this thing works.

Leo: Like a code book, but it's generated anew each time, kind of.

Steve: Yes. Now, the problem is, we don't need new encryption. And that's sort of where I immediately fall back to is, wait a minute, this problem has been solved. We have solved the encryption problem in an absolutely robust way. But, and I just sort of had this weird sort of queasy feeling about, you know, this doesn't sound like it really makes that much sense. I mean, it sounds like there's - if you use real cryptography, this thing would not stand up at all. Well, so I did a little bit of web searching, and not much, in fact. I put "virtual matrix encryption" into Google. And the first link that came up was by our friend Bruce Schneier, who is actually a cryptographer, who is the author of Blowfish, a still state-of-the-art, robust, never has been a weakness found in it, cipher. He wrote in February 18th of 2003 the following, which is just a perfect synopsis of sort of this notion of let's just have noncryptographers invent virtual matrix encryption.

So on February 18th, 2003 he says, "Back in 1999" - so four years before this - "I wrote an essay about cryptographic snake oil and the common warning signs. Meganet's Virtual Matrix Encryption (VME) was a shining example. It's now four years later, and they're still around, peddling the same pseudomathematical nonsense, albeit with a more professional-looking website. I get at least one query a month about these guys. And recently they convinced a reporter to write an article that echoes their nonsensical claims. It's time to doghouse these bozos once and for all.

"First, an aside. If you're a new reader, or someone who doesn't know about cryptography, this is going to seem harsh. You might think, how does he know that this is nonsense? If it's so bad, why can't he break it? That's actually backwards. In the world of cryptography, we assume something is broken until we have evidence to the contrary. And I mean evidence, not proof. Everything Meganet writes clearly indicates that they haven't the faintest idea about how modern cryptography works. It's as if you went to a doctor who talked about bloodletting and humors and magical healing properties of pyramids. Sure, it's possible that he's right. But you're going to switch doctors. Two essays of mine, one on snake oil and the other on amateur cipher designers, will help put this into context." And he goes on. So that's our answer to Kerry

Merritt, who...

Leo: He's not saying it doesn't work. He's just saying it's not needed. There's good public key cryptography out there now that works.

Steve: Well, and there's public key, there's symmetric key, I mean, my position is this problem has been solved. The last thing - we have Rijndael. We have RSA. All of this stuff is now in the public domain. Their patents have expired, as opposed to just having been issued in 2001. And they solved the problem in a way that absolutely makes sense. So it's very likely that this technology can be cracked because it's not based on sound crypto science and technology. It's based on some guy saying, oh, let's see, I'm going to use a million-bit key. Whoa, that's a lot more bits than 128. So it must be more secure. No, not necessarily.

Leo: Right, right, yeah. All right, so just it's unnecessary.

Steve: Yeah. But...

Leo: Bruce makes a good point, that it's not like you need it. And why should you trust an amateur cryptographer? What's not known is how reliable this alternate system is, really.

Steve: And I love crypto. And the last thing I'm going to do is go and invent my own bit-scrambling routine. It's like, no, thank you, that's been - that problem has been solved.

Leo: Right. Although I have to say Neal Stephenson did invent a very clever crypto scheme using a deck of cards in "Cryptonomicon." And I think Bruce may have had something to do with it, or was involved in it, or passed judgment on it.

Steve: I think I remember that he was consulted, yeah.

Leo: So, I mean, it's fun to do it. It's a great hobby. But if you're going to trust your business data to it, it might be good to use well-known and established technologies.

Steve: I think even - I would phrase that more strongly, Leo.

Leo: Okay. Ryan Sullivan asks a, quote, "random question" from New York: I'm trying to write a PIN number-generating program. I'm having a slight issue when it comes to generating a completely random number. I know that software cannot be completely random. The issue really is that, when I run the program, the first number to be generated is always the same. He's new to programming, I think. The second is always the same. While the numbers are random, they come up in the same random order every time. It's called "pseudorandom." I'm currently using the Random function in the programming language.

Steve: Actually I have to interrupt. I would say I would call that "not random."

Leo: Yeah, that would be exactly not random. I've spent a few hours trying different combinations of adding or multiplying two or more random numbers together. I cannot seem to figure out how to at least have the numbers come up in a different random order each time. He needs to read his programming book a little better. Is there some mathematical equation or algorithm that can give me somewhat random outcome, or at least more random than I'm currently getting? Everybody who learns to program does this once. I'm not laughing...

Steve: Oh, sure. Absolutely that's the case. Now, the fact is the pseudorandom number generators which are available in most programming languages, ones like Ryan is probably seeing where it's giving him the same sequence of pseudorandom numbers, they generally use a technique which is extremely poor, which is called "linear congruential pseudorandom number generation." Essentially you multiply the current number by some constant, and then you add a different constant to get you the next one. Essentially, this generates a sequence of numbers which is really low in entropy because, again, you're multiplying by something and you're adding something. It's just not going to give you...

Leo: Really? That's how they're generating? That's not random at all.

Steve: That's all they do. Now, the good news is there's a function in most of these languages that I'm sure you're aware of, Leo, called "randomize," which you can call once at the beginning of your program. And it will generally pick a random seed, well, I mean, that's what it does. It prevents this problem of your program being entirely deterministic in the sequence of "random" numbers it generates. However, as it happens, I just finished writing an extremely good source of extremely random numbers as part of the solution to this roaming authentication problem, which I'll be talking about next week. And I'm making all of the code freely available on GRC. So, Ryan, if you listen to this, if you hear this, and you're willing to wait one week, you can just get this code from me, call it from your program, and believe me, you will never get the same number twice.

Leo: Usually there's a seed to the random number generator that your programming language comes with. You seed it first. And if you choose something kind of random to do that...

Steve: Well, for example, you could ask for a high-resolution time of day from your system.

Leo: Yeah, ticks, the ticks.

Steve: Exactly. And use that to start things off.

Leo: But you'll get a random first number, but the problem is the sequence will not be random after that.

Steve: Exactly. You will start with, at a different point in, basically, in a relatively short loop that this multiply and add, this so-called linear congruential algorithm gives. Most languages just do that because they're really not cryptographically random. But again, I've got some free code I'm making available because I had to solve the same problem myself just recently. And it generates extremely good random numbers. In fact, we had the randomness of the Perfect

Passwords Page tested by some crypto guys, and it was way up at the top, at the high end of the scale in a competition with a bunch of other random, high-quality random number generators.

Leo: Where did you get that algorithm? How did you come up with that? Was it out of a book or...

Steve: No, I just made it up.

Leo: You just said, this should work.

Steve: Well, no. The beauty is, if you have a good cipher, essentially a good cipher like Rijndael, which is what I used, I used Rijndael encryption with a 256-bit key, which is the longest key it's able to give. By definition, if you put anything in one end, like just a counter, 1-2-3-4-5, into one side of a good cipher, what comes out is absolutely pseudorandom. I mean, it is highly random, so that you are unable to predict what's going to come out next, even if what goes in is absolutely as predictable as a counter.

Leo: Isn't that interesting.

Steve: And that's what I did, and it works.

Leo: So, Ryan, sounds like he's probably learning to program, and that's something that - don't feel bad. Everybody does the first time.

Steve: Hey, I've been there, too, absolutely.

Leo: I have no idea how you'd write a real random number generator. I'm really interested to see how we do this next time. This'll be the next episode?

Steve: Yup, next week.

Leo: Can't wait. An anonymous listener worries about replying upon browser cookie security. Let me explain.

Steve: Oh, relying upon.

Leo: Relying. Okay, that makes more sense. All right. I have a question pertaining to the last episode of Security Now!. That was 113, I believe. Any cookie set on a system by legitimate.webserverA.com, let's say, is supposed to be displayed back to that company and only to that company. In other words, any illegitimate webserverX.com has no way to read that cookie. Well, what happens if at some point the browser mechanism is somewhat faulty, maybe a bug strikes, and the cookie does get read by some otherwise unauthorized website? In other words, theoretically a browser is not going to offer a cookie back to a

non-matching domain. What happens, though, if some site by, say, using JavaScript is somehow able to read the browser cookie anyways? Is that something to worry about?

Steve: Well, it's a good question. And of course it's a function of what the cookie stores. This was triggered by my discussion last week of this idea of a master enabling permanent cookie and then a second cookie which would be a session-only cookie that would be given to a roaming user after they had securely authenticated themselves to the server. But in general cookies have no meaning to other web servers. That is, most cookies are, although they don't have to be, are what's called an "opaque token." There's some random-looking gibberish, probably something that's been encrypted, which is stored as an ASCII text string in the cookie as the cookie's data, so that this opaque token has meaning only to the issuing site when it receives it back. So if some malicious website did have access to the cookie - and again, this is something which certainly browsers are struggling to avoid because cookies have become such an intrinsic part of web security these days, you certainly hope that a browser is not going to indiscriminately disclose cookies to non-authorized domains. But even if it did happen that it was disclosed, if you actually look at the content of most of your cookies, you'll see that they just look like...

Leo: Gobbledy-gook.

Steve: ...absolute gibberish that would have no meaning whatsoever to a site that didn't know how to decrypt it or interpret what data was stored in the cookie.

Leo: So it's a pretty unlikely combination of flaws. First the browser has to be broken, and the cookie has to be something that makes sense and has to turn out to be something useful instead of the last date of your visit.

Steve: Well, yes. And, for example, a web server that receives it, it's just going to ignore it. It's going to say, wait a minute, this is not a cookie I understand. It's not like it's a smart web server or a malicious web server that knows it's going to be leaking cookies from some broken browser that's going to ever have been to some other website to have acquired a cookie that would be of interest to it. So again, it's, like, way out there.

Leo: It'd be pretty unlikely. Now on to our next question, also about cookies. I've been listening to your Security Now! Episode 113. I had a couple of questions about the permanent cookie you're talking about. First of all, is this cookie placed inside the to-be-trusted computer's web browser, or is it placed somewhere else on the computer? If it's placed inside the browser, how do you make sure it doesn't get deleted by accident or by the user, say when he's cleaning the browser's cache or other browsing traces? And if it's not placed in the browser, but somewhere on the computer, where would that be, and how would the server be able to locate and see that cookie? Is this entire cookie system designed to work only with Internet Explorer, or is it going to work with Opera and Firefox or Safari? Tell me, Steve. He's really worried.

Steve: Well, the idea was that I wanted to use this permanent cookie to potentially enable a machine to connect to GRC when it's not at one of the known secure IP networks. That is, the server at GRC knows my own network, and it knows the IP of Sue's NAT router and the IP of Greg's NAT router. And those NAT routers tend, even though they're using DHCP, tend to hold those IPs static unless they unplug their NAT router or reboot it or leave it offline for some period of time. So that's been the way we've authenticated so far. If their IP changes, then they

simply let me know. They send me a piece of email that has their new IP in their email headers, and I add that, or I change the old IP in the registry, and they're back up and running and up to speed. So the idea is, only if their laptop is at one of those IPs is it able to receive this master cookie.

So to the first question, if that master cookie were deleted, they clean the browser's cache, delete their cookies, well, all they have to do is reconnect their laptop back to their home network, which would give it the public IP that's been authorized. And the moment they try to use our management interface, the server will see that their browser has not given them the master cookie back, either saying this machine is either authorized or not authorized. And so they get an intercept screen saying, wait a minute, please let me know whether this machine should be authorized for roaming authentication whenever it's not at one of these authorized IPs. And it is just a standard browser cookie. It's not stored elsewhere in the machine. It's just there in the browser. It's meant as one of several sort of interlocking requirements for them to be prompted to solve this puzzle or this secret, something which is not going to be easily or even possibly cracked by somebody with full information about their logon. And that's what we're going to talk about next week.

Leo: Okay. Victor, lurking somewhere in the U.K., had a question about roaming - Roman [sic] authentication. That was our last episode.

Steve: Right.

Leo: I just love that name. You talked a lot about your new roaming authentication, but there are a couple of things that occurred to me. If a laptop already has malware, can't this malware just steal your, quote, "secret cookie"? I mean, after you've successfully logged into the system, a thief can use your session cookie and spoof an IP address it's coming from. And if this were all done on a hotel's private Ethernet, he doesn't even need to spoof the IP address because it all looks like it's coming from the same place.

Steve: Well, he's certainly correct that if - let's see. If we were logged in, if a thief were on the same network, then they would be - then we would see that they were on the same IP. They would have to have malware on the laptop which was able to recognize what was going on and get both the static cookie, which authorizes the transaction to create a session cookie, and the session cookie, which would be initiated only when the laptop's user authenticated uniquely each time. And they'd have to then have a way of returning both of those cookies with any transactions and know the nonpublic and the, well, the nonpublic URL for this backend management system, which is also heavily encrypted. And they'd have to be able to do all that over, even though we were using SSL connections for the transaction.

So if the user - if this attacker were not also on the same Ethernet, there is no way to spoof the IP. I've mentioned, I think it was last week, that TCP connections are spoof-proof because you have to have this three-way packet transferal in order to establish a connection. And the session cookie does have encrypted into it, that is, the contents of the session cookie, the authorized IP that the user was located at when they solved this puzzle that we'll be talking about next week. So it's not possible to spoof it. So if you had malware, if you captured both cookies, and you were on the same network so that you saw the same public IP, and you knew the nonpublic cryptographically strong URLs we're using, then yes, you could pretend to be one of our users. But it would take all of that in order to make it happen. Which essentially means you have to be one of our users.

Leo: Yeah. That's interesting. There is a way to do it. I'm actually kind of surprised.

Steve: Yeah. I mean, there is this practice now that there's been some discussion of called "sidejacking," the idea being that some sites do not maintain a secure connection after the user logs on. For example, Google, we've talked about Gmail, Google Mail. It will make you use a secure connection while you're providing your username and password for logon. But unless you use a secure URL when you go to mail.google.com, it will drop you back to a nonsecure connection. The problem is that it continues to maintain state using a cookie. And that cookie is going through the clear. So, for example, in a scenario, in an open Wi-Fi situation, for example, where you are not using secure browser connections, the practice of sidejacking can occur where, even if they did not get your username and password because that had been secured by an SSL connection, once you've established that, if the cookie is not flagged as secure, and you drop back to a non-SSL connection - and in fact in this case the cookie could not be marked as secure because the server still needs access to the cookie when you're over a non-SSL connection - somebody could watch one of your Gmail transactions, every one of which would contain the cookie, and simply masquerade as you. They would grab your cookie, start using your cookie, and essentially commandeer your session. And that's sidejacking, which is a problem for cookie-based state management in transactions which are supposed to be secure.

Leo: It's kind of like a man in the middle, though. Why do they call it sidejacking?

Steve: Well, because you're not in the middle, you're on the side. You're not actually needing to filter and intercept the user's traffic. You're able to sniff it and just watch it go by and see, oh, look, this guy's doing Gmail on port 80, nonsecure. There goes the cookie for his Gmail session. I'm going to grab it.

Leo: You don't intercept it, you just watch it go by and make a copy, basically.

Steve: Exactly.

Leo: Yeah. So a man in the middle, you'd intercept it and send something else to the server.

Steve: And so, for example, this is the reason why the GRC session management, the management interface, absolutely never falls away from being over an SSL connection. It will not accept any of this happening unless we have a secure connection with our server. And that's maintained all the time.

Leo: Right. Brian Polak of St. Louis, Missouri receives the Security Now! Brilliant Idea Award for the week. I hope we can do this every time.

Steve: Yeah, this was a - he makes a great point. It's simple, and it is just - it's so neat, I absolutely wanted to share it with our listeners.

Leo: Thanks for alerting me to the PayPal security key, writes Brian. It makes me feel a lot better about using PayPal. And in answer to the previous questions you've said to just combine the token value with your password on the first login page. I don't think that's a good idea, he says. As we know, if you do not provide it with the password, PayPal prompts you for it in the next screen - which generally is how I do it just because I'm lazy. I use this feature as an extra check to make sure I'm not being phished. If I ever try to log

on and am not asked for the token value, I know right away something is potentially wrong. Oh, that's a good point.

Steve: Isn't that? That is the brilliant idea of the week.

Leo: If it does happen to be a phishing site, I know I've not given them enough to log in, and I can go to PayPal and change my password if necessary. The minor inconvenience this adds to my login is well worth it for the security benefit it provides. As you pointed out, you're safe from phishing in the sense that that number is only good for 30 seconds, so the chances are that even if they get that login, they can't use it. Nevertheless, this will let you know you're not on a valid site.

Steve: Well, yeah. I like it from - just from a standpoint of feedback. I have been putting the PayPal password and my six-digit, one-time output from my token in all at once. And but I'm also very aware of the problem of phishing. I'm always right-clicking on the page and checking the properties and checking the certificate chain because this is such a potential problem, although as you say, Leo, much mitigated once you've got a hardware token protecting your login. Still, I just like the idea. I think he makes a very good point that no phishing site would know to prompt your account for your token. They would just ask for your username and password and hope that you're going to give it to them. If they don't prompt you for that, then that raises an immediate flag. And so I just thought that was a very great observation.

Leo: Bank of America has implemented something finally. They still use that lame SiteKey. But I think I'm now starting to feel a little bit better about the overall protection because what they do is they put a permanent cookie on your machine, kind of like what you do. And if they don't see that permanent cookie they say, ah, you're logging in from a different machine, or the cookie was deleted or whatever. We're going to need some more information. And normally they ask you a secret question. But, and I've turned this feature on, you can also have them send a key to your cell phone. You have to validate the cell phone with them ahead of time in an authenticated session so that they know it's really you. But from now on, if somebody tries to log on to my bank account from a machine that's never logged into that account before and doesn't have that permanent cookie, it'll say, okay, we're going to send the security number to your cell phone, enter it here. So it's very much like that token system, only it's using the cell phone, something I almost always have with me. You can set up multiple phones, so if you have multiple numbers, or my spouse, for instance, Jennifer has a separate number, so she can use it, too. I think this is a good solution.

Steve: Yes. I'm really glad that we're seeing this kind of more robust authentication evolving. You know I'm just an authentication fanatic at this point. I mean, I really - it's why we've been talking about OpenID, we've talked about the security tokens, and it's why next week I'm going to share with our listeners the solution I came up with for allowing myself and my employees to securely log in to GRC with so much security that even somebody who had absolutely full knowledge of that transaction, everything that the user sees and everything that they type in in response, cannot do it again.

Leo: Woohoo. That's next episode of Security Now! Yay. Don't forget that you can get 16KB versions of this for the bandwidth impaired, transcripts for those who like to read along with Steve, and show notes. And in this case there is a number of links you might want to check out at GRC.com/securitynow.

While you're at GRC, don't forget to check out Steve's many free security programs for your download and use, including the ShieldsUP! test that everybody should run when they get a new router or firewall online. And of course the fantastic SpinRite, my favorite disk recovery and maintenance utility, a must-have. If you've got a hard drive, you should have SpinRite. It's all at GRC.com. Steve, next week the conclusion of our Roman authentication.

Steve: Right-o. We'll have a different title for the episode, which we will reveal next week.

Leo: Maybe Persian authentication? I don't know. Thank you, Steve. We'll see you next time.

Steve: Thanks, Leo.