SECURITY NOW!

Transcript of Episode #15

VPN Secure Tunneling Solutions

Description: Leo and I discuss the use of SSL and SSH encrypted tunneling for providing privacy and security whenever an insecure local network is being used - such as at an open WiFi hotspot or when using a hotel's network. These solutions are not transparent and tend to be configuration intensive. They also require the use of a "server" of some sort at the user's home or office. This makes these approaches less suitable for casual users, but offers a solution for the more technically inclined road warriors.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-015.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-015-lq.mp3

Leo Laporte: This is Security Now! Episode 15 for Thanksgiving Day, November 24, 2005: Part 2 of our continuing series on Virtual Private Networks. Steve Gibson. And a very good afternoon, and a Happy Thanksgiving to you.

Steve Gibson: Yeah, thank you.

Leo: We're actually doing this the day before Thanksgiving. Are you going to do Thanksgiving? Do you do Thanksgiving?

Steve: Oh, yeah. I've got plans down here with some friends...

Leo: Great, great.

Steve: ...to hang out and relax a little bit.

Leo: My mom's in town. We're going to have a big Thanksgiving. And I'm supposed to be making stuffing right now. But no. I'm making a podcast. Security Now! is more important than stuffing.

Last week we talked about VPNs, Virtual Private Networks. And we did the theory of VPNs. In fact, I got a number of emails saying, quick, get to the practice. What do you do with them?

Steve: Yeah.

Leo: So shall we recap last week?

Steve: Yeah. I also want to mention a little bit on the continuing saga of Sony. The news is now the Attorney General of Texas has filed suit against Sony.

Leo: Good.

Steve: The EFF, who were considering filing suit - that is, the Electronic Frontier Foundation - have filed suit.

Leo: Whew.

Steve: And there are reportedly six class-action suits also now pending.

Leo: Wow. And what I thought very interesting, the Recording Industry Association of America said no, no - that was, you know, Cary Sherman, who's the CEO of the RIAA...

Steve: Yeah, defended this.

Leo: Yeah, defended it. He said no, no, no, the only issue was there was a security flaw. But they were just doing copy protection. No big deal, we plan to keep doing this. You know, wasn't perfect. But he said Sony did a good job of making up for it with the recall. And he said they're doing the right thing. And, you know, no problem. What's the problem here?

Steve: On TWiT on Sunday I mischaracterized some email that I received from Amazon that I wanted to just fix because I'd read a bunch of BBS postings and various blogs and things after receiving this email. And I made a comment that they were saying that Amazon sent me email saying that the CD was defective.

Leo: Yeah.

Steve: Actually, when I went back, I thought, oh, I really need to make this clear, that I got email from Amazon because I purchased a CD specifically with the Sony DRM rootkit stuff on it in order to experiment with it myself. They followed up - and I was very impressed, actually, that Amazon did this. Their email reads: "The Sony CDs listed above contain XCP digital rights management DRM software. Due to security concerns raised about the use of CDs containing the software on PCs, Sony has recalled these CDs and has asked Amazon.com to remove all unsold CDs with XCP software from our store. Since you purchased this CD from Amazon.com, you may return it to us for a full refund, regardless of whether the CD is opened or unopened. Just visit..." blah blah blah. "Thanks for your understanding. We hope to see you again at Amazon.com."

Leo: That's actually a very good explanation, I think.

Steve: Yeah, it was very clear and very thorough.

Leo: Yeah.

Steve: And, you know, and anybody who isn't in the loop on this would have received email like this if they had purchased one of these CDs and, you know, in that fashion be brought up to speed, at least to some degree. It's like, oh, wow, you know, now what do I do?

Leo: Yeah, I think that's fair to both sides and an accurate description of it. And I hope other retailers maybe kind of take a page from Amazon's book because of course the recall is global.

Steve: Well, of course Amazon has the advantage of being able to look at their database of everyone who purchased the CDs in question and send them retroactive email. You know, if you buy something from Good

Guys or Best Buy or Fry's or wherever, there's no way for other retails, you know, physical retailers to get a hold of you.

Leo: So if you've got one, bring it back.

Steve: Yeah.

Leo: And they're supposed to take it back.

Steve: The other thing I wanted to mention before we continue with VPNs is I've gotten a lot of great feedback from people who are loving the little passwords page that I created. I've been so bogged down with VPN research for this podcast that I haven't been able to finish the page. Although it's fully functional, I'm adding application notes to the bottom. So I just wanted to remind people about, if you just put in GRC.com/passwords, you get a page which generates really high-security passwords for whatever purpose you have. [Perfect Passwords: GRC's Ultra High Security Password Generator, at www.GRC.com/pass or www.GRC.com/password or www.GRC.com/passwords.]

Leo: All right. And that's free and available for anybody who wants to use it. And it's that 63- or 64-character password that Steve recommends for people who are using WPA on a wireless network.

Steve: Right. Well, and even, for example, VPN Shared Keys.

Leo: Oh, you could use it for that, too, sure.

Steve: Or, yeah, I mean, just sort of any...

Leo: Anything you need a 64...

Steve: ...any purpose.

Leo: Or take what you need and leave the other letters, as well.

Steve: Well, yeah. In fact, the passwords are what's called "maximum entropy" passwords, meaning that at any point, the next character is equally probable to be anything within the alphabet, which means that shorter chunks of those passwords are just as good, that is, in terms of like just as random as the whole thing. So if you've got something that, like, only allows you - for example, I have a router that only accepts a 49-character pre-shared key for VPN setup. And it's like, okay, you know, you give it as much as it'll take.

Leo: A number of people sent me notes with other password generators. There are lots of programs out there. In fact, OSX includes in its Keychain application a random password generator. And I'm sure they implemented it well, although the problem is one doesn't know.

Steve: Right.

Leo: And Steve's taken great care to make sure this is really random.

Steve: Well, you know, it only displays itself over a secured connection. And the page is postdated 1999 so that no proxy will hold onto it, thinking that it might still be good, and serve it to somebody else. So I did what's possible in terms of, you know, preventing anyone else from ever seeing the passwords that the page provides.

Leo: If you are creating a password in OSX, open the Keychain Access application, press the "+" button, and you'll see next to the password there's a little key. You can press that, and that gives you the password assistant. And Steve, you'll like this. They give you a variety of different choices: memorable, letters and numbers, numbers only, random, and FIPS 181 compliant. I presume that that's some sort of high-quality...

Steve: Standard.

Leo: ...standard.

Steve: Yeah.

Leo: And they have a little slider for length. And they even give you a little bar that shows you the quality. So I think that this is probably a pretty, I mean, pretty good one. Although it only goes up to 31 letters, so you'd have to double it up if you want to be 64. Isn't that strange.

Steve: Yes, do it twice.

Leo: I don't know where they got 31 from, but that's the maximum number.

So last week, Steve, we established that for home use you want to turn on WPA encryption, and that's going to give you a safe connection with your wireless network at home. And we also raised this specter of this problem: When you go out, and you're using a hotel or a hotspot that doesn't have any encryption, it is of necessity open, that everything you do is now suddenly back in the open and insecure. And we talked a little bit about VPNs as one way to secure this. Let's just go back a little bit and explain the issue and why VPNs are helpful.

Steve: Great. First I'll mention, relative to WPA, there have been a lot of questions, great questions that people have sent, which we'll be addressing in next week's divisible-by-four Security Now! Podcast No. 16, which will be our Q&A episode.

Leo: Right, right, okay.

Steve: Okay.

Leo: So hold your questions till next week. Actually, send them, but we'll answer them next week.

Steve: Right.

Leo: But now we're on the road. Our WPA is not helping us at all.

Steve: Yes, exactly.

Leo: Because I'm using the Holiday Inn, and it's got an open access port. And now what do I do?

Steve: Exactly. So the idea is, in a network where you have control, we've really covered clearly the issues of securing your own environment. The question now is, what happens when you're out on the road? You're in a hotel. We know from empirical evidence that about half of the hotels use a hub rather than a switch - because hubs are a lot cheaper, and the hotel doesn't care - in order to network all of the rooms. A hub means that somebody in any room in the hotel running a packet sniffer is going to see all the traffic from all the hotel guests using their broadband solution.

Leo: So actually I misspoke. It's not just wireless. It's wired connections in the hotel, too.

Steve: Yes.

Leo: Every connection.

Steve: Yes. And the problem, of course, is that typical email protocols are in the clear, not only for the log-in, but for the text of the email. So when you go to a hotel - and, I mean, most people, what they're doing is they're checking their email. They're wanting to make sure, you know, that they're staying current with any email that they've had come in. It's over POP or IMAP, you know, port 110, or they're sending out through SMTP on port 25. Anybody sniffing that will see their username, their log-in, their password, and the contents of their mail. And of course the same is true if you're in an open WiFi hotspot, where you're taking advantage of somebody's kindness to use their open wireless. Again, as we know, that means there's no encryption.

Leo: And lest you think, well, this is only a temporary thing, once they get your email password and log-in, they can get it from now on until you change it. So, you know, you could go home, and they're still reading your email until you change your password.

Steve: And, I mean, some people might think, well, that's not a big deal. But, I mean, you can be - any cookies that are sent back and forth would allow someone, for example, to impersonate your automatic log-on onto Amazon or eBay.

Leo: Ooh, good point.

Steve: Which is cookie based.

Leo: Right.

Steve: So they don't even need your username and password because, as we know, when you come back to Amazon or back to eBay or one of these sites that uses persistent cookies to identify you, they just say, "Hi there, Steve." You know, "Click here if you're not Steve." They do that by using a cookie, which is in the clear, not over a secure connection. So this allows somebody else to impersonate you. I guess my point is, this is really a serious security issue, using non-encrypted communications in a public setting where there's one way or another for your traffic to be sniffed, either in a network with a hub or wireless, which is inherently sniffing.

Leo: Okay. So we now see the problem. And we all are exposed to this. I mean, I do this all the time. How do I secure my traffic? What are the options open to me?

Steve: Certainly there are some protocols, like we talk about HTTP as opposed to HTTPS. With HTTPS, the "S" stands for "Secure." That creates what's called an SSL tunnel. When you connect to the server, there's a secure negotiation which occurs. And essentially, any traffic that then travels over that connection is encrypted. It only carries the normal web traffic. But, for example, it's the way ecommerce is able to accept your personal credentials, credit card numbers and so forth, without that being a security risk.

Leo: Okay. So that's good.

Steve: Yeah, exactly. And, for example, there are some email systems where you're able to use a similar SSL connection for email. The problem is, those are protocol-specific. They're only for specific applications that you're running which are sort of preset to operate in a secure fashion. So in general, if we step back from those examples for a second, what they have in common is this establishment of a secure tunnel. We started talking about this last week, the idea being that a connection is established between two points. Then any traffic which you're sending back and forth is encrypted and then placed in a packet which is sent to the other end. It is expecting this, so it takes the wrapper off of that and then decrypts the traffic in order to get what it was that you originally sent. Essentially, it means that anybody sniffing your traffic, and even someone who is able to intercept your traffic, the so-called man in the middle, would be thwarted and would not be able to in any way gain access to the content of what you were sending. I mean, it's the perfect solution for this kind of mobile use of an otherwise insecure environment in a secure fashion.

Leo: I wish we could do it all the time.

Steve: Well, that's really the goal. And we're going to, over the next few weeks, we're going to talk about strategies and approaches. And I've sort of set myself a holy grail. I know where I want to get. But there's a lot of history here that I want to talk about because there are sort of halfway measures and other solutions that are useful, too.

Leo: All right. I don't want to get you out of order here. So we want to get that holy grail set up at some point. Can we talk about that down the road, and talk instead now about the easy stuff?

Steve: I think we should. One of the oldest...

Leo: I want to protect people now, you know?

Steve: Yes.

Leo: Yes.

Steve: Or in the meantime, at least make sure people understand what dangers they're facing if they're not protected.

Leo: Right.

Steve: One of the oldest approaches is something called "SSH tunneling." SSH stands for Secure Shell. It comes from the UNIX world, as does so much of this technology, the idea being that when you are remotely administering a UNIX machine - in the old days you would use Telnet, which is just - basically it's sort of a terminal window where you're typing text into the window. As you're typing it, the text is sent to the computer, and it's just like you were sitting in front of it at the console, entering those characters. And then the things it would display on the screen instead it sends back to you, and it's displayed on your remote screen.

Leo: Yeah. Well, I have that capability on my Mac. I mean, that's something I could set up, SSH tunneling. Although I've never really got it working exactly.

Steve: Well, actually that's Telnet that I was talking about.

Leo: Oh, Telnet, okay. Well, Telnet's the insecure SSH.

Steve: Exactly.

Leo: I see, okay.

Steve: And it was the lack of security of Telnet...

Leo: I get it. I'm sorry, I didn't mean to interrupt you.

Steve: It's okay. It's the lack of security of Telnet which induced people to say, okay, we've got a problem here. If I log on to my UNIX machine remotely, my root username and password, for example, are going over the 'Net in the clear.

Leo: Right.

Steve: So anybody that was sniffing the traffic would be able to get my log-on credentials as God of that remote machine. So they said, okay, we need to come up with a solution. What they came up with is actually an extremely robust solution called Secure Shell, or SSH. The idea is it's very much like what we just discussed with SSL, the Secure Sockets Layer, which is used, for example, with ecommerce, or anytime you're doing HTTPS over your browser. The Secure Shell, SSH, creates an endpoint-to-endpoint tunnel. And at that point your log-in, anything after that's connected is completely encrypted. It is very, very securely encrypted.

Leo: Now, I use that from now - I've been using that for the last couple of years to get into my website, once I realized Telnet was insecure. And that works really well.

Steve: Yes.

Leo: And it's very much like Telnet. You get a terminal session and everything.

Steve: Well, in fact it's identical. You would not know that you were running over a secure connection in any way. Basically it's transparent. And in fact, transparency is one of the keys of this whole notion of tunneling, the idea being that you could be, you know, in a Telnet, in an insecure Telnet session. You're typing, and you're seeing what you've typed. The same thing happens with an SSH session except that, unbeknownst to you visibly, everything that you type looks just like noise, completely scrambled, you know, dust going through the Internet to the far end.

Leo: So that's SSH log-in versus Telnet. Now, what is SSH tunneling?

Steve: Well, now, that's an interesting sort of, I have to say, kluge. Because that's really what it is. Now say that somebody wanted a secure connection for traffic which would otherwise not be secure, for example, email. Normal POP account to port 110 is username and password, and all the contents back and forth of your email is in the clear. Say that you were traveling around, and you wanted some way to protect that, even though that protocol itself is not secure. What you can do is you use what's called an "SSH tunnel." And now we're going to be a little techie here. It's not something for the faint of heart. But it is a very well-proven and effective approach. The idea is, your SSH client running on your computer is able to support what's called "tunneling." You instruct it to listen for connections on an unused port of your own machine, like maybe port 1111, for example. I mean, I'm just making that up. It could be anything from 1 to 65535, you know, any one of the local ports. So you tell this sort of little server running on your machine to listen for anything connecting on your own local port, 1111; and, when that occurs, to accept the connection and send any traffic after encryption down that SSH connection to a receiving server at the other end, to an SSH server to which you've already connected.

So essentially you've got two connections. You've got the secure connection across the Internet from your SSH client to a remote SSH server. And then you've got a second connection that is not secure, but it's just within your own computer. You tell your email system, rather than connecting to a remote server on port 110, which is the normal POP port, remotely, instead you tell it to connect to your own computer, often called the Local Host, and sometimes referred to with an IP address 127.001. You tell it to connect to this Local Host port, 1111. So when you check your email now, your email client connects to a port in your own machine. It has an insecure dialogue to that port. But everything that it sends into that port gets encrypted and sent across the Internet. And everything returning encrypted gets decrypted and returned. So what this does is essentially it creates a secure connection where there was no protocol security beforehand.

Leo: That's why it's called a tunnel. I mean, you're essentially establishing a new highway from your computer to another computer. And you just have - but the problem with this is you have to tell each client, okay, don't use 110 anymore, you've got to use 101. And then when the tunnel's not there, 101's not going to work, so you have to reconfigure it.

Steve: Yeah. I mean, it really is - it's something that people who are spinning the propellers on top of their heads are able to get to work. But exactly as you said, if you were doing this with a laptop, when you were traveling you'd have to establish this SSH connection to an SSH server. The other thing is, you know, you'd have to have an SSH server somewhere. If someone were a Linux or a UNIX guy, they could be running an SSH server at home, and then create an opening through a router or their firewall to allow SSH traffic to come in.

Leo: The good news is that's not so hard. On Mac, for instance, you can do it very easily with the system preferences, under the sharing preference pane. You just turn on the SSH server. The problem is you have to then open up a hole in your - you have to forward the port through your router for that to work.

Steve: Right. And know that essentially...

Leo: And incoming traffic's going to be a problem. I don't know how you do incoming traffic at all.

Steve: Well, in fact, all the traffic coming from the outside would then be routed through that port.

Leo: You have to port forward, right.

Steve: Now, that's normally port 22. One of the first things you would do for security is you would want to move that to some other port number because SSH is often attacked because it's a well-known port.

Leo: And while it is fairly secure, there have been holes in it in the past. So you can't...

Steve: Of course. And that's the problem that you run into when you're running any server on your own network.

Leo: Right. So just for Apple users, just so they know, in the sharing preference pane you turn on Remote Log-in. And that will turn on an SSH server for you. And then you'll have to port forward to make it work. But by default it uses port 22, so I guess that may be not the best advice anyway.

Steve: So, yes. You would want to tell the server to run on a different port if you were getting into this that way.

Leo: Right.

Steve: But returning to the point you made, when you're out roaming around, and you've established this SSH tunnel, and you've got this SSH tunneling server on your roaming computer, exactly as you said, you'd have to configure your email client to use a different port than it normally would in order for this SSH server to intercept it. Not only port, though, but also IP. You're telling it don't go out to my normal POP server, which is located, you know, somewhere on my ISP.

Leo: Use Local Host.

Steve: Instead, yes, connect to my own machine, this own machine right here, which has a special SSH server, which is essentially redirecting that port traffic down the tunnel. Furthermore, it would only work, you know, everything we just described solves the problem for one single protocol. That is, you know, just getting your email from your ISP to your computer. If you wanted to send email, now you need to do the same thing for, essentially, for SMTP - run another tunnel on another port, tell your email client again to send mail to the Local Host address on this port, and that would need to get intercepted and rerouted similarly.

Leo: The way people usually do this is with a series of scripts. The SSH command line is long and abstruse. But once you've got it working, you just run the script, and it turns them all on.

Steve: Well, it would create all of your local port services.

Leo: Right, reconfigure...

Steve: But you'd still then need to reconfigure all your clients.

Leo: Although I suppose if you just, you know, you always used it, it would be okay.

Steve: That's a good point. If you always ran in that mode, then you would just leave it that way all the time.

Leo: All the time, yeah. I've been tempted to do this because I, in fact, do run SSH all the time. And I have SSH open so I can log into my computer as a terminal. And I've been tempted to create some tunnels just to see if this works. But so it's free. Not easy, but it is free, and it does work. And it's secure; right?

Steve: Yes. It's extremely secure. SSH is able to operate with very strong certificates, using public and private key cryptography. So, I mean, it's as secure as anything can be. It's a well-thought-out, solid protocol that will definitely allow you to move traffic between two endpoints that nobody can sniff and have access to.

Leo: So that's SSH tunneling.

Steve: That's SSH tunneling.

Leo: Are there other ways we can do this?

Steve: Yes.

Leo: Okay. Shall we talk about those, or do you want to save them for later?

Steve: Let's talk about SSL solutions.

Leo: That might be the easiest way to do this. And you're already doing SSL.

Steve: Well, yes. SSL is - now, that's back to Secure Sockets, much as web browsers that are running over a secure connection are tunneling their HTTP traffic through this secure tunnel. The real difference with SSL, I think, is that it's got such a strong browser orientation that many solutions are browser based, which hugely simplifies configuration problems. For example, does GoToMyPC run over an SSL connection?

Leo: It does.

Steve: Yes.

Leo: As does Anonymizer. In fact, any solution that is trying to sell itself to consumers is likely to use SSL for the simple reason that that port, 443, is almost always open.

Steve: Well, yes, it...

Leo: You don't have to have any router configuration.

Steve: Well, exactly. Anybody who's going to be using the web needs to be able to switch over to a secure web connection, and that'll be over port 443 and SSL.

Leo: So it's the same thing you're using when you go to your bank or you buy a book on Amazon or any - that's the same SSL. It's secure. It's HTTPS.

Steve: Right, Secure Socket Layer.

Leo: Okay.

Steve: Now, the thing that's interesting about this is that many of these solutions will download an application, like a Java applet, which your local computer knows how to run. And essentially they'll provide the client side for you...

Leo: Oh.

Steve: ...automatically.

Leo: Now, that's interesting. Because you already have an SSL client, your browser. What you need is an SSL server on your home system.

Steve: Well, you need a - when you're using some sort of remote system, for example, you go to a library, you log on to GoToMyPC. What GoToMyPC does is send you an applet which runs on that browser locally to be the client side of the solution.

Leo: Right.

Steve: So...

Leo: Oh, that's so - let me clarify, then. I don't need a client to use SSL with my bank. That's built into the browser.

Steve: Well, actually the browser is a simple text client, is sort of the way to think of it.

Leo: So if I want more capabilities, then I may need to download additional software on the client side.

Steve: Or the browser will automatically run Java.

Leo: Java will do it, yeah, yeah.

Steve: Exactly. So the idea would be that the service would provide you with an applet in Java which any browser knows how to run.

Leo: So I put this on my home PC. I log into it with my browser. I will have to open ports to do that, yes?

Steve: Actually, no, you're not putting anything on your home PC.

Leo: Oh, I'm not.

Steve: Your browser logs into the service, and the browser downloads it, just the same way like you download Flash or something.

Leo: I know what the confusion is. I'm still going for this holy grail of me running the server. You're saying I can't run the SSL server. I have to use a third-party service to do this.

Steve: Well, it's a good thing you mentioned that because I found one.

Leo: Oh. So normally I would go through somebody like PublicVPN.com or Boingo. There are a lot of places that do this that basically offer an SSL server that I can use.

Steve: Right.

Leo: But we want to do this cheap or free, so we don't want to pay a monthly fee. This is that holy grail you're going for. We would like to do it so we run it on our own home system and surf through our own home connection.

Steve: Correct. Now, certainly there are people for whom GoToMyPC or any...

Leo: Anonymizer...

Steve: Exactly. Anonymizer, any of these public VPNing services would make sense because they don't even have to have a home.

Leo: Right.

Steve: I mean, they're using it wherever they are, out roaming around.

Leo: Well, in fact, that's better in some ways. You don't have to leave a home PC on. You don't have to make sure your connection's always up, blah blah. And you may have a slow connection at home. You may not want to use your home connection.

Steve: Correct.

Leo: Right.

Steve: For do-it-yourself people, there's something called SSL-Explorer, which is a very interesting-looking solution. It's hosted by SourceForge. If you just put SSL-Explorer into Google, it'll take any of our listeners directly there.

Leo: I'll put that on the show notes, too.

Steve: Great.

Leo: It's a SourceForge project.

Steve: Right.

Leo: So it's free and open source, okay.

Steve: What's interesting about it is that essentially they do what we've been talking about, except that instead of the server end being something that you have to subscribe to, and which inherently means you're paying, you know, I don't know, $10, $15, $20 a month, whatever, for the service, you run the server out of your home. So the idea is, some machine, whether it's a Linux box or a spare Windows machine or maybe your main Windows machine, it's sitting at home and is the server for you to access remotely. So when you're out and about, essentially you connect to your home's IP address. And of course there are now - all the routers are supporting dynamic DNS, which essentially gives you a normal domain name which will track any changes to your home router IP, so you don't have to worry about the IP changing and you being caught by surprise.

Leo: Oh, I didn't know that. I've been still recommending DynDNS.org and some of these other sites. So your new routers do this.

Steve: Yes.

Leo: Oh, I didn't know that. How...

Steve: Well, in fact, the way they do it is with DynDNS.

Leo: Oh.

Steve: But they're DynDNS clients.

Leo: They run the client on the router instead of on your system.

Steve: Exactly.

Leo: I see.

Steve: And so the router has the responsibility for it. And when it notices that its address has changed, it will contact DynDNS and say, hey, here's the new IP address for this domain.

Leo: So what's wrong with this? This sounds like the holy grail right there. This just sounds like the way to do it.

Steve: Well, it's close. It's not - I'm still not happy because there's another step that I want to go to.

Leo: Why, Steve? Why aren't you happy?

Steve: Because it does mean that it's - again, it's a little more techie. You need to have a computer dedicated to this or, as you said, on and running all the time on your home network.

Leo: Right.

Steve: You need to be able to run a client-side Java applet in a browser. It might be that a library has their browsers locked down, for example, so that you're not able to run applets of any kind on the browser. Although, you know, again, if it was for your own use, for you to use in a laptop as you were traveling around, then you're going to be in good shape. The key is that the connections to it are over SSL. It's extremely easy to configure. There's good documentation about this SSL-Explorer. And for somebody who's interested in experimenting with this kind of solution, I think it's, I mean, I've spent the afternoon reading the docs and taking a look at it. I'm very impressed with what you're able to do. You're able to access your files remotely. You're able to run your desktop using the standard remote desktop protocol, which is very efficient. So you're able to, like, essentially operate your home computer as if you were there, very much like GoToMyPC does.

Leo: That's cool. So we would run SSL-Explorer. It's currently Windows and Linux only, Mac version as soon as the new Java is available for the Mac. And you can surf the 'Net. You can do everything you want. But you have to leave this machine on running this software.

Steve: Yes. And at the client end it's completely browser agnostic. It just uses Java. So, for example, you could go to a library, you could use a hotel's Internet service down off the lobby, you could use your own laptop and your own browser. Essentially, it allows you to access your home network in a completely secure fashion and do all kinds of things - transfer files, map drives, access your regular desktop on the computer. And it's absolutely free. So that's really why I like it, as opposed to using some sort of a subscription service.

Leo: So there you have the easy, or not easy, but there are two ways to do it. Neither of them is the perfect way. But you could do it with SSH tunneling. That's really geeky and hard to set up. On the other hand, pretty transparent once it's working. SSL might be a better way to do it. And if you don't mind paying one of the SSL services like HotSpotVPN.com or PublicVPN.com or GoToMyPC, or Anonymizer might be a good choice. And if you want to save money and do it yourself, SSL-Explorer. But none answers Steve's holy grail, the holy grail of computing. And what is that?

Steve: Well, the holy grail - and this is where we're going to spend some time in a couple weeks, so we'll just leave this as a bit of a teaser - the holy grail is no software installed on the client machine for any version of Windows and Linux and probably Mac, although we'll need to see whether Mac supports one of the protocols required. The idea is you have nothing but a residential VPN router installed in your home. We've already talked about NAT routers and how great they are for security. There's an additional feature that is now offered by the same residential routers called VPN. That is a Virtual Private Network endpoint, the idea being that Windows already has VPN technology. Windows 2000 and XP has it. And they have a free download for 98, ME, and NT4. So the idea would be you use the built-in software in Windows to connect to your VPN router, which means that that's then a secure connection. And it's all protocols. It's actually, you know, it'll run through POP and SMTP and web browsing. Anything you do is running through this Virtual Private Network tunnel to your router at home. Once there, you have access to your home network, if you've left any machines on. But even if everything's turned off at home except the router, sitting there glowing in the corner by itself, then you're able to use that router to get out to the Internet.

Leo: That's the holy grail. I agree with you. I like this. However, it's not that easy.

Steve: It's a configuration nightmare.

Leo: And we're going to hold that discussion for another episode. Because that alone could take us half an hour.

Next week we'll answer your questions. If you have questions, go to GRC.com/securitynow.htm. Or suggestions. We'll just respond - our viewer mailbag next week; right?

Steve: Well, you know, I have hundreds already.

Leo: So don't send anymore.

Steve: I mean, oh.

Leo: We've got plenty. But that's what we're going to - barring a major security issue in the news, that's what we're going to do every fourth episode.

Steve: Yup.

Leo: That's what's coming up next in Episode 16. And then in Episode 17 we will, we hope, be able to explain how to set this holy grail up, this VPN router. Costs around 100 bucks, easy to set up, and you're good to go. You're golden.

Steve: Many people may already have them. If not, they could probably upgrade their router's firmware...

Leo: Really.

Steve: ...to get the additional functionality. Or maybe sell their old one on eBay and buy a new shiny one. The beauty is, you pay then nobody any money every month.

Leo: Right, it's yours. And you run it. And meanwhile, I'm going to try to get SSH tunneling up, just to show it can be done.

Steve: Cool.

Leo: Maybe I'll have something to - then I've spent nothing to do that. And we'll see if that works. Good luck.

All right, Steve. If people want to know more, they should go to the website. You'll have full notes here at GRC.com/securitynow.htm. That's also where you'll find the written transcripts and the 16KB version of the show. The 64-bit, full-fidelity version is available via podcast, of course, and you can use your favorite podcast client to get Security Now!. We're on all the different networks and channels and so forth. Or just go to ThisWeekinTech.com and you can find a direct link. That goes through AOL. And we do thank the folks at AOL Radio who put us on their podcast channel and provide the bandwidth for Security Now! at AOLmusic.com. Thank you, guys.

Before we go, Steve, I just want to mention one thing. And I know anybody who listens to Security Now! knows enough not to open email attachments. That' s kind of the fundamental basics. But just because you will be the go-to person for your friends and family, there is a email scam going around, email purporting to be from the FBI or the CIA or the U.S. Postal Service. I'll read you the FBI one: "Dear Sir or Madam: We've logged your IP address in more than 30 illegal websites. Important: Please answer our questions. This list of questions is attached. FBI." And the CIA one says "We can help disinfect." So if you get an email from the FBI or the CIA with an attachment, don't be dumb enough to run it. Of course it's a phony. Both the CIA and the FBI have announcements on their web page saying, we don't do that. We don't send out emails. Don't open attachments. You'll get the SOBER virus. It's going to send itself out to other people and open your system to trojan access, including probably spam servers. It'll Zombie your computer. So you know better, gang, because you listen to Security Now!. But tell your friends and family, if they get an email from the FBI or the CIA, just delete the email.

Steve: Yup.

Leo: Delete the email.

Steve: I did receive one of those, by the way.

Leo: Did you?

Steve: Yeah.

Leo: We're all getting them. According to MessageLabs, I think there were - MessageLabs has intercepted over two million of these. So what that tells me is that people are opening it.

Steve: Yup.

Leo: Which is kind of hard to believe in this day and age that somebody would actually buy that. But, you know, it's scary, and you go, oh, the FBI, oh.

Steve: It's like the reason we still get spam is that it pays. People will buy things from spam ads.

Leo: You're right. If nobody ever bought anything.

Steve: Right.

Leo: All right, Steve. Thank you very much. Happy Thanksgiving and have a good Turkey Day.

Steve: Thank you, Leo. Always a pleasure.

Leo: We'll talk again next week on Security Now!. Bye bye.