SECURITY NOW!

Transcript of Episode #200

Listener Feedback #68

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

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-200.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-200-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 200 for June 11, 2009: Your questions, Steve's answers. This show is brought to you by listeners like you and your contributions. We couldn't do it without you. Thanks so much.

It's time for Security Now!, the show that covers all things secure online, privacy, all that stuff. Here he is, our security expert, Steve Gibson of GRC.com, creator of SpinRite, discoverer of spyware, and the host of Episode 200. Steve.

Steve Gibson: Yes, our double golden anniversary episode.

Leo: Holy-moly. Who woulda thunk it?

Steve: 200, yeah.

Leo: That's pretty great. I'm really happy.

Steve: Well, and from all the feedback that I get - this is a Q&A episode, so I ran through the mailbag. And we've got of course our regular 12 questions and a ton of security news this week. But it looks like from the feedback we get we're going strong. And everyone just says please never stop. Please please please, never never stop. So...

Leo: Well, unless Steve stops, I don't have any plans to stop. I'm really enjoying it.

Steve: Based on the security news this week, it doesn't look like we're going to ever run out of content.

Leo: I love that. Well, so we have one commercial from our new sponsor, GoToAssist. That's coming up a little bit later, so we'll get to that. How about after the security news?

Steve: Sounds great.

Leo: Let's find out what's going on. I understand Microsoft did its biggest Patch Tuesday in history, I think.

Steve: 31 vulnerabilities.

Leo: Holy cow. Holy cow.

Steve: I mean, yeah. Even there's a vulnerability in Search, if you can believe it. And in Works, that no one even has used for quite a while. It's just...

Leo: How do you even find a vulnerability in a program nobody uses?

Steve: And the print spooler has three vulnerabilities.

Leo: What?

Steve: It's like, oh, goodness, yeah. It's quite something. So the big news, though, I just sort of had to kind of smile because I told all of our listeners this was going to happen. I said just wait, this is a bad idea, we're going to see how bad it is. Trojans have - trojan software has been found in ATMs located in Eastern Europe.

Leo: Oh. Oh.

Steve: From many different vendors.

Leo: Oh, dear.

Steve: But what one thing do all of the trojan-infected ATMs have in common, Leo?

Leo: Let me guess.

Steve: Mm-hmm.

Leo: Windows?

Steve: Windows XP.

Leo: Ai yi yi.

Steve: The LSASS service is the manager of protected content in the system. It's not quite the right acronym. I can't think of what it is right now. But it's like the main security service. And fake ones have been found in the Windows directory. The LSASS EXE normally lives in the Windows System32 directory. They were written in Borland's Delphi.

Leo: You're kidding.

Steve: No.

Leo: Well, that's kind of sophisticated for a hacker. Wow.

Steve: And it's considered, I mean, it's commercial-grade code. It's good code.

Leo: Oh, boy.

Steve: These are not remote installation trojans. It's believed that somebody had to have access to the machines.

Leo: Oh, even worse.

Steve: But they have special credit cards. When they swipe the special credit card in the infected machine, it accesses the trojan software, which among other things allows them to dump out all the cash from the machine. But in the meantime it's logging all of the users' information and PINs, which it's able to dump out encrypted with DES encryption from the printer, from the ATM printer in the front of the machine.

Leo: Wow.

Steve: So the - and anyway, so it's interesting to me. Again, it's, you know, people defended the idea of implementing these things that I contend should never have been written in Windows. They say, well, but it's easier to write them. And it's like, yes.

Leo: For everybody.

Steve: And it's also easier to write trojans.

Leo: For everybody involved. We're all happy.

Steve: You can have a, yes, a really nice GUI interface and, you know, lots of third-party support with something like Borland's Delphi in order to write your trojans. But if it were instead written in one of these much more obscure industrial and industrial-strength multitasking real-time operating systems, it'd be far more difficult. They wouldn't be able to write the software on the machines that they received for Christmas. So anyway, I got a kick out of that. I mean, it's bad news. The good news is, as far as we know, it's relatively constrained to Eastern Europe, and - but many machines over there. But this is not the kind of thing you want on your own local bank's ATM, logging all of your ATM transactions and then being willing to dump them on command. So, yikes.

Also the Pirate Bay appeal was denied. Remember that the four Pirate Bay guys were found guilty. Then there was an appeal that was brought based on the membership of a couple of the judges in a pro-copyright organization. So it went to appeal. And the judgment quickly came back that, yes, we know that these guys are pro-copyright members. But that's just to keep them informed of the copyright law and what's going on. There's no indication that it in any way biased their judgment. So the sentence stands. At the same time there was recently an election, as you may know.

Leo: Yes. This is amazing.

Steve: And the Pirate Party won one of 18 seats in Sweden's Parliament. The Pirate Party was established three years ago in response to Swedish legislation that made filesharing a crime. So the party's publically declared aims are to reform European copyright law, abolish the European patent system - got to get rid of that pesky intellectual property altogether - and eliminate digital rights management, DRM. And also to allow, as a consequence of all that, free filesharing on the Internet. And what's interesting is that this wacky Pirate Party got a substantial percentage. I think it was 9.something, 9.1 percent of the vote for these guys, and they won a seat. So I don't know what that means in terms of their goals.

Leo: Well, these people are pissed off about DRM. I think people haven't thought through the whole issue of, well, if you don't have intellectual property protection of any kind, if you have no patent or copyright system, nobody's going to invent or copyright anything in Sweden.

Steve: Right.

Leo: But I think what this is, is the pendulum swinging the other way. People have been - are fed up by the ridiculous copyright, patent, and DRM issues.

Steve: By having the stuff really getting in their way.

Leo: Yeah.

Steve: Yes. I mean, again, we know that it's one thing to pirate, which we're all against. It's another to be able to have your own personal fair use blocked by things that get in your way. I mean...

Leo: This is what happens when companies assume and treat you as a pirate.

Steve: Right.

Leo: Then you're going to have some reaction. And I think that's what this is. It's really stunning that they got almost 10 percent of the vote. That's amazing.

Steve: I did pick up on the mail that you forwarded me yesterday, or I guess it was last evening, about the Comcast Social Security number insecurity.

Leo: Can you believe that one?

Steve: Oh. On the Unixjunkie blog, a guy named Greg Miller blogged - and Greg is a Google software engineer. This is not super recent. This was last October 26th in '08 he blogged. He was setting up his - he was going to be moving and setting up Comcast, a new Comcast subscription in the new location where he was moving to. And everything seemed to be going fine until the last step required that he do an online text chat with a nameless, faceless Comcast person.

So this Java applet downloads, and he noted that it downloaded over HTTPS. But he was skeptical of whether the actual interactive chat text was also encrypted, hoping that it was, specially because one of the things they required was that he enter his Social Security number into the chat. And it already had his name and address and other clearly personally identifying information. The chat-ee at the other end said, oh, don't worry, you can trust Comcast, it's secure. Well, being a Google software engineer and a techie, he fired up a packet capture and watched as unencrypted, in the clear transactions went back and forth. You didn't even have to guess what the content was because every field was clearly labeled, you know, Social Security number equals, and then first name, last name, date, address, city, state, zip, all the fields clearly identifying what their content was, just being sent right out of his machine with no security at all.

So he put the blog posting up to warn any other people that they absolutely didn't want to do that. Oh, and he gave the person a hard time because I remember looking at actually the chat text that was in the clear. And the person responded, "Well, if you don't trust us to do this, then you can call somebody and do it live over the phone." And he says, "Thank you, that's what I think I'm going to do."

Leo: Wow.

Steve: Unbelievable. When I turned on my Mac this morning, there was a whole bunch of updates, but for a change not security related. These were just feature updates. You know, "i" everything - iLife, iDVD, iMovie, iPhoto, and AirPort - all got updates. And of course there was a big Safari 4 update.

Leo: Yeah. And a nice one, too.

Steve: Yeah, it looks really...

Leo: They put the tabs back, I mean, where they belong, instead of the top of the window.

Steve: And it's got this whole panoramic sort of, what is it, is it 9 or 16 pages, it might be 16 pages, that it learns where you go the most and sort of shows you sort of that folder view that they've got for the iPod view of, like, your popular sites and what the current faces of those are. It's not real fast. I mean, when you fire it up, mine took - I was curious, so I ran Safari 4 after the update. It took a good minute, maybe, to fill in those things, so - and it was doing them one at a time.

Leo: It does it very fast once you have - the first time you did it, it was looking through your bookmarks.

Steve: Ah, okay.

Leo: It's instant. If you just do create a new tab now on Safari, you'll see that that's the default blank page.

Steve: Oh, good.

Leo: And it does it very, very quickly.

Steve: Okay.

Leo: Yeah. No, I think the, you know, they're using the latest WebKit, which is a very fast JavaScript interpreter. I think comparable to Chrome. And this is important for them because Safari is also on the iPhone. So they really want to keep this thing up.

Steve: Right. Well, and Microsoft. We have a little more after Microsoft, but it's time for Microsoft. 31 vulnerabilities in Windows. There's some in Windows Active Directory, in the kernel itself, in IE, in Word, Excel, in Works, in IIS - the web server - and in Search.

Leo: Unbelievable. Wow.

Steve: But believe it or not, with all of that, remember the bad zero-day exploit that we discussed last week? It was in DirectX's DirectShow component for displaying QuickTime, and we gave our listeners that link to the "Fix it" button that they could press in order to fix the problem there. For whatever reason, that's not fixed. So there is still a known zero-day exploit where it's a remote code execution problem, it is in the wild, it is being done, and it involves, if you went to a web page that brings up QuickTime player, that would be played by DirectShow, then there's a known vulnerability there that is being exploited. So disabling that in the meantime, and maybe we're going to need to now wait a month. I don't know if Microsoft will do an out-of-cycle patch for this. I don't know what the delay is because they had the "Fix it" button certainly at this time last week. But it's not part of the fixes.

What is part is two vulnerabilities in Active Directory, which are critical remote code execution; three vulnerabilities in the print spooler, which are critical remote code execution; eight vulnerabilities in IE, again, critical remote code execution, open a specially crafted web page and, bang, you're owned; two vulnerabilities in Word, open a specially crafted Word doc and something can take you over; quote, "several" in Excel. I think they just got tired of counting at this point. This didn't say, although I did go to another, their other page and counted. That's how I know that it's 31. A vulnerability in Windows Works, same thing, open a specially crafted Works file. The RPC, Remote Procedure Call, has a privilege elevation, so that's not what they call critical, it's just important. And the kernel has four privilege elevation bugs.

Leo: And that's serious. Those have to be critical.

Steve: Yeah. Two were publicly disclosed, and two were privately shared only with Microsoft. IIS, their web server, has one, a vulnerability that involves authentication. If you're challenged for authentication, there's a way that you can send a specially crafted response which will essentially bypass the normal privilege level that you would have authenticating at that level, and you're able to elevate. And Search, Windows Search, has a - depending upon, like if there's some way that you can search and return a file, and that trips it up and allows an information disclosure vulnerability is what they called it. So not real bad. But overall this is, I mean, it's a definite reboot of your system. I've got all of mine sitting here pending. I didn't want to do it before the podcast because who knows if the machine would survive and come back up afterwards. But I'll do it when we're through recording.

Leo: Yes.

Steve: And I want to tell all of our listeners, shut down things gracefully. But this is one - there's so much in this package that it's definitely worth a reboot in order to catch up. And we also talked last week about Adobe's formally announced plans of coming into synch with quarterly updates. Well, this being June, which is an even quarter in the calendar, and this is Microsoft's Patch Tuesday, it's also Adobe's Patch Tuesday.

What I found interesting was that my Acrobat on this machine was giving me no indication of new versions. But when I asked it, it said, oh, yeah, got something. And it also appears that it is needing to install these sequentially. Just the other day I installed a brand new install of Acrobat in my tablet that I'm using at Starbucks. And I was able to download, I think maybe it was 8.13, I mean 8.1.3. But then I had to successively have it do - have it apply .4, .5, and I think that's where it stopped. So what I'm noticing is when I ask my older Acrobat on this system, it says 8.15. But I know that 8.16 is available.

So what our listeners need to do is update until it tells you there are no more. So don't just do it once. So your target is, if you're using v9, major v9, with day before yesterday's, that is, Tuesday's update, we're at 9.1.2. Version 8 is at 8.1.6, and v7 is at 7.1.3. So in any event, what you want to do is just keep, you know, maybe it makes you reboot each time or not. But keep working at it until it finally says okay, whew, there is nothing else. And you should be at 9.1.2, 8.1.6, or 7.1.3.

Leo: Good.

Steve: We also had some interesting news in the DNS world. On June 2, which was, what, like last week, the .org root was the first major root of DNS to receive a DNSSEC signature.

Leo: Hello. Hello.

Steve: Oh, sorry, got distracted. I just had something come or go. I forgot to - I always forget to turn my speaker down.

Leo: I thought that was a Skype sound.

Steve: I ought to just turn that off because I don't care if people are coming and going. Anyway, so last week the .org root was signed, which is the first of our major DNS roots to be signed. But also the NIST, the National Institute of Standards in Technology has asked ICANN, you know, the main governing body of the Internet, to work with VeriSign, who has been given responsibility to get all of the root name servers signed with DNSSEC by the end of 2009. So by the end of this year we're going to finally get the root servers signed. The .gov servers were signed a few months ago, which is a good thing. That was mandated by the government, and it was made to happen. So now the public servers are going to get themselves signed.

Leo: And this will take care of that Dan Kaminsky security flaw that we talked about.

Steve: Well, you really need - it's moved forward. We've got a long way to go because you need DNSSEC at both ends. You need a DNSSEC-aware server in order to understand and verify the signatures that are being given. So the fact that the DNSSEC records are, that is, the signatures are available, that's the prerequisite to using them. But you still have to use them. But so we've got a ways to go before when you actually get a record out the other end, the whole chain has been, like, it's very much like the way we think of certificate chains, where you need a chain of certification all the way back to the trusted root. So the same thing has to happen here. But again, it can't happen until it starts. And so it has started, which is way good news.

Leo: Yeah.

Steve: And then last thing I just want to mention, I haven't mentioned this for a long time, I developed the Perfect Paper Passwords system when I was needing some way to allow my employees, Sue and Greg, to roam with their laptops, yet still have really, really robust, secure authentication to GRC. I'm now using it since I'm at Starbucks. So I'm a roamer, too.

Leo: Oh, that's neat. Oh, that's neat.

Steve: And it's just, I have to say, I mean, obviously the stuff I do in the future, like CryptoLink, will support Perfect Paper Passwords natively. But it just - it really works, Leo.

Leo: Are you surprised?

Steve: Well, you know, it's one thing to design it and write it. But then to actually be a user of it...

Leo: No, it's good to eat your own dog food once in a while, that's good.

Steve: And I was never using it. But so you go to the page, and it says, we need C7. And so you get your little - I have it in my wallet, my little index. And I go, oh, C7. And I type the characters in that it wants and then also give it my password. So it's something I know and something I have. And it says, okay, fine, here you go. It's like, wow, you know? And I can't give it the wrong one, and I can't give it the same one again because it's a one-time password system.

Leo: So just to clarify, this is a system you wrote, but it's logging into your corporate VPN, your corporate network.

Steve: Actually it's, yeah, it's secure web access to resources that we have at our...

Leo: So it's not a VPN, but it has - it's SSL, but it's via the web, yeah.

Steve: Correct. Correct. And uses a whole interlocking cookie system. I mean, you have to - I described it all in detail once. You have to receive a cookie when you're at your home address to, like, enable the machine to then be a roaming machine. And then so you have to have that. And then you get an authentication cookie when you are roaming which is valid for a limited period of time until you close the browser. So it's a session cookie, not a static cookie. I did, I went over the top with security. But is there such a thing?

Leo: Can you go over the top? No.

Steve: Exactly.

Leo: I think not.

Steve: And I do have a short little blurb about SpinRite. I have a really interesting and fun long one. But since I'm sure this episode is going to go long because it's one of our Q&As that are never short, I just thought I'd share a fun short story from someone named Nate Friedman. Dane wasn't sure if maybe he was somebody that you knew.

Leo: I don't think so. That doesn't - but I shouldn't say that.

Steve: Well, he's also in - I notice now that he's in Santa Rosa. And Dane said there was someone who is in the chatroom, I think, that is...

Leo: Oh, he might be a chatroom guy.

Steve: Anyway, so he said: "Steve, I wanted to share my SpinRite story while it was fresh, but would love it if you read this on Episode 200..."

Leo: Okay, deal.

Steve: ..."of Security Now!." So I was all set to read the other one, too. And I thought, well, okay, since Nate's asking for a specific episode number, he gets it. He said, "I've been listening to you and Leo for a few months straight now and wish I had started sooner. I recently inherited responsibility for a mission-critical server with a failing hard drive. The previous administrator had tried to clone the hard drive during nighttime hours three times, and failed each time. Having heard the many success stories about SpinRite, I talked the money people into buying a copy. It found a few bad sectors running at Level 3, so I ran it again at Level 4, just over that section of the drive. Then I went back and properly cloned the hard drive without any trouble to a fresh one, and the server's been humming along like a champ ever since."

Leo: Excellent.

Steve: So Nate, thanks for the story, and another success for SpinRite.

Leo: We have, Mr. Gibson, some good questions for you, 12 questions good and true from our wonderful listeners, questions compiled from GRC.com/feedback, including questions about BRIEF...

Steve: Oh, and actually there was a bunch of stuff from last week, some errata-ish questions, too. I just put my foot in my mouth at one point.

Leo: Uh-oh.

Steve: So that's question number one. I wanted to get it right off the top of the show.

Leo: All right. Starting with the foot-in-mouth - foot-in-mouth extraction. Jim Millard in Kansas City, Missouri picked up on - and you wrote this, not me - Steve's bone-headed statement about IPv6 and consumer switches. He says: Steve, I've posted this in the newsgroups, but you made a mistake - I'm sure he didn't mean this in a mean way, just wants to get it corrected - in your characterization that commodity switches will "break" - I'm paraphrasing - when we move, if we move to IPv6. I presume Jim works for somebody like Linksys or somebody. Recall that a switch, unless outfitted with extra functionality that is NOT typical in consumer equipment - operates at Layer 2, not Layer 3. A switch keeps a list of MAC addresses for each port, not the IP address. So v6 is not going to bug it at all.

Steve: I know. Oh.

Leo: Oh, don't feel so bad.

Steve: And it's not that Jim needs to work at Linksys or somewhere. You wouldn't believe, I mean, I'm impressed with our listeners because...

Leo: They picked that up right away.

Steve: When I picked up the mail I thought, wow, look at all this mail. Unfortunately half of it was people saying, eh, I don't think so, Gibson. Didn't you have your coffee that morning? I mean, I know. And we've talked about this so many times on the show. It's like, okay, I don't even know what I was thinking. I think this is what they call a "brain fart," Leo.

Leo: Yeah, yeah.

Steve: Just for clarification, a switch, exactly as Jim and many other listeners said, runs at Layer 2, meaning that it's at the MAC addressing level, not the IP addressing level. So it doesn't - it's not even aware of the payload of the Ethernet packets. It's switching the Ethernet packets based on MAC address, which is what we understand is unique on the physical local LAN, thanks to MAC addresses being unique.

When a machine, an active machine, not the switch itself, which is sort of passive, when an active machine is trying to send an Ethernet packet to a machine on the LAN by IP address, it uses an ARP packet - ARP, Address Resolution Protocol. And it broadcasts it to the last IP on the network, which is a special - I'm sorry, the last, yeah, I'm sorry, exactly, it broadcasts it on the network so that all devices on the network will hear it. And it says, which of you has this IP, that is, has that IP assigned to your adapter? Well, one of them will respond, it's like, hey, I'm the guy with that IP. The switch is passively monitoring this dialogue, and it knows which wires, that is, which ports these various chunks of the LAN are plugged into. So it memorizes the MAC addresses living on each of its different ports. And that's how, for example, a switch can - the same switch can be used, not only for IP traffic, but for example Novell's older IPX/SPX protocol, which is - it's IP related, but it's not the same protocol. So again, yes, Jim was right. I don't know what I was thinking when I rambled off the wrong statement.

Leo: You knew better. You knew better.

Steve: I wanted to correct the record for everybody who - first of all, for people who I may have confused with that. And I wanted to acknowledge the torrent of corrections that I received and to thank everyone for making sure that I knew what I was talking about.

Leo: No one doubts that anyway. Let's move on to question two. John Meuser in Indianapolis, Indiana wonders about non-VPNs. He writes: Steve, your non-VPN solution - this was somebody who had written this in; right? Oh, no, you were talking about - this was the third subject of last week's show, of course - sounds a lot like SSH port forwarding with something like a port-knocking daemon to hide that there is a port listening. Is it?

Steve: Yeah. There are - we also got a ton of responses to my talking about this notion of a non-VPN solution, sort of a secure connection approach. And I'm gratified because it's very clear from the response that our listeners have a huge interest in secure connections. So I wanted to clarify a little bit briefly about how this is different. With SSH port forwarding you run essentially a listening service on your local machine which is listening for connections to the so-called localhost address which we've talked about, 127.0.0.1. So it's a server inasmuch as it's looking for connections to that address. It's also a client of the remote end which is functioning as a server. So the idea is you tell your IMail client or your POP client or whatever to connect to your own machine at whatever port you have set up this SSH listener to be listening on. It accepts the connection, and then it in turn reaches out and securely connects, using SSH secure protocol, connects to the remote endpoint running at your actual destination. And so that's how the conversation works.

My approach that I talked about last week is a little different in that, very much like many UNIX solutions, this doesn't require anything in the kernel. This approach is essentially able to run a daemon which is listening to - that is, the SSH approach is able to run a daemon which is listening for connections to a port, and it then forwards them to somewhere else. So it's nice in that it's able to run in user space. My approach, and the one that I will be taking with the development of CryptoLink, will involve a kernel driver which inserts itself in between the NIC, the network LAN adapter, and the rest of the PC, that is, before the protocol stack that does all of the work, and just before the packets leave or just after they come in. So the CryptoLink driver will have raw access to the actual packet.

So what that allows is much more transparency of operation. So, for example, you would not need to reconfigure your email or POP or Windows networking or anything, telling it to aim at your own machine. You'd actually tell it, connect out there to the target. And so the CryptoLink driver in this case would see that happening, intercept the packets on their way, as they've left the protocol driver, just before they get to the LAN adapter, the machine's NIC, and it would change them and encrypt them on the fly so that this system just works without the added overhead of starting up the port forwarder and configuring it and so forth. So that's really how my approach is different.

And as for the port knocking side, again, that's another clever solution which is able to run in user land, sort of as a user-level application over on the receiving system, to allow packets to - and we've talked about port forwarding or port knocking in the past - to allow packets to hit a firewall and be noticed, and the sequence or content of those packets in some cases is used as a key that then causes something to open a port on the fly, which then allows the actual traffic to come in.

What I have the first of a number of patents pending on is I've figured out a way for the connection-opening SYN packet to self-authenticate so that it itself contains the authentication information allowing it to be identified as coming from a legitimate matching CryptoLink client that shares the same key as the recipient. And so while you've got CryptoLink running, your system is completely stealth. It appears that you have no ports open at all, when in fact they're all opportunities for CryptoLink packets to come in.

And that's the other cool thing is that, which will allow CryptoLink to establish connections more robustly, is you could send packets to any port number. And again, because the CryptoLink driver is listening before the computer, that is, just after the packet is received and before it starts being processed, it's able to look at these packets, authenticate them, and say, oh, hey, this is a special one for me. And then it's able to do its job. So there's a bunch of extra magic happening. I'm glad for our listeners' interest, and it's going to be fun to be developing this.

Leo: Be careful. You keep promoting it. Pretty soon there's going to start being people storming your doors, saying, "Where is CryptoLink?" We're getting - you're giving us more and more reason to want it. I'm excited, really excited.

Brad Beyenhof in San Diego, California says: NoVPN? Port forwarding? Sounds like NoVPN - or he probably means non-VPN; right?

Steve: Sure.

Leo: Would have to either, A, use port forwarding to let the secure ports through your router; or, B, put the non-VPN server machine into a DMZ and rely on the software firewall plus non-VPN authorization encryption to remain stealthed. Former might be a little tough for the average user, and it's difficult to provide help given the wide array of router interfaces out there. They don't even call it port forwarding on many routers. So you can't even say, well, you need to do port forwarding, because every router has its own terminology for what you're doing.

The latter, DMZ, is slightly less difficult, but still requires router reconfiguration to disable the hardware firewall. Anyway, didn't you tell us years ago not to rely on software firewalls? So what's the story, Steve?

Steve: Well, I wanted to - I selected this question because I realized I had inadvertently also confused some people. One of the things that I hope will make the VPN very popular is that in a very Hamachi-like way it will support NAT traversal. So it will not be necessary for you to preconfigure your routers. You'll be able to use NAT traversal. However, there's absolutely no way around the need for a third party if both endpoints are behind NAT. We've talked about NAT traversal in the past. I'm sure probably I'll bring it up again when I'm deep in the middle of actually solving the problem and characterizing various NAT routers.

But the idea is that, if one of the two parties that wants to connect is behind a NAT router, well, the one behind the NAT router is able to send packets out through the NAT and directly to the other party that's not behind a NAT router. That works. If neither party is behind a NAT router, of course you have no problem. If both parties are behind a NAT router, the problem is that they're not able to send packets to each other because we know how NAT routers function as very good hardware-level firewalls. They allow the egress of traffic. And then traffic that's expected back in can come in. But they don't allow unsolicited traffic to get in.

So the solution is for a third party, like a so-called rendezvous server, a third party that's not behind a NAT router, standing out where both of them are able to access it, like out on the public Internet, they send traffic to that third party. The third party analyzes the traffic, figures out how their NATs are mapping between the private network and the public network. And then, through the connection that they initiated to the third party, that third party sends instructions back to each of the parties behind their respective NATs, saying here's how you need to send packets to each other. They then simultaneously send packets to each other with the hope that the transparency that's been created by the presence of the third party providing them this extra information will allow their packets to get back into each other's NAT.

So it's tricky, but it obviously works. NAT traversal, Skype has it, Hamachi has it, our show sponsors have it. It can be done. I'm going to do it, too. But it breaks my TNO, my Trust No One model, because you have to have a third party. And the other problem that many people experienced with Hamachi is, if Hamachi was ever down, and unfortunately Hamachi spent a lot of time being down, you couldn't initiate new connections. If the third party's missing, then you're out of luck. So I recognize that there are more sophisticated users who can configure their NAT router, who don't need a zero configuration solution. What they would prefer is a, first of all, absolutely Trust No One solution, no third party involved ever, and in return for that they'll have to do a little bit of configuration. But then they also get the reliability of not relying on a third party, in addition to the additional security.

So CryptoLink will work both ways. It'll work in a zero-configuration mode and do NAT traversal. But for professional people, I mean, the way I would configure it for myself is I would configure my router to allow external traffic in to the CryptoLink server, knowing then that I'm able to accept connections from anywhere, whether behind a NAT or not, because I've essentially made the receiving end transparent. So I just wanted to clarify that, that it will work both ways - in a "just works" fashion where it may be necessary to rely on a third party, but for our sophisticated users you won't need to use that.

Leo: Very good. Again, you're making us salivate.

Steve: I am, too. I can't wait to get started on it. Have a little more work to get done, then I'm on it.

Leo: So you haven't actually started it. We should just make that clear.

Steve: No. I've got a ton of notes. I've got intellectual property stuff working. But I've got a short list of things that I just, I want to finish, that I've invested hugely in already, that it's just dumb not to finish them. So I'm going to get them finished. I talk about the third-party cookie stuff, the DNS spoofability, the DNS benchmark, and just a few other things. And once those are out of the way - and they're almost finished. I just need mostly documentation. So I'm on it.

Leo: I'm on it. Yeah, you forget that there's all this other kind of prep work that you have to do before you can even begin.

Steve: Well, and frankly I'm...

Leo: At least it'd be prudent to do.

Steve: I'm glad to have the time to think this through. I mean, remember, Leo, you and I were in Vancouver, and I ran through the list of several pages of things that CryptoLink would do that nothing else has ever done. And you looked at me like, my God, this is going to be really good.

Leo: It looks really cool, yeah.

Steve: Yeah. But I just, you know, it's just me, so...

Leo: Well, you know what carpenters say. You say measure twice, cut once. And planning is really important. We were talking about how I was looking at curricula for high school students for programming and so forth.

Steve: And we've got - we have a couple ideas about that in our Q&A, as a matter of fact.

Leo: Oh, good. Well, one of the books I found is called "How to Design Programs." It's a wonderful book that's been used as a curriculum at Yale and Rice and MIT and a lot of other places. I mentioned this before, it uses DrScheme, which is a stripped-down version of LISP. But it very much focuses on planning. Before you write one line of code. It really teaches you a methodology so that you think - you do what you're doing right now. You really think about what you're going to do, and you know what you're going to do before you write a line of code.

Steve: Oh, and I forgot that, as a consequence of this last week, of all the email and the dialogue in our newsgroups, I realized that this notion of a secure connection is not what I want to do. What I want to do is what I call "tunnel on demand." So it's just as transparent except that you're running a full VPN with all the additional features that it provides, like you can send multiple connections through a single tunnel. You can have many different types of traffic. What I was trying to avoid was the overhead of having to start something. And I'm used to having to do that with OpenVPN because - and it takes, with OpenVPN, it takes several seconds of starting it up, and it scrolls through all the stuff it's doing. There's all this happening before you get your tunnel established. And so it's definitely overhead. And so I realized that's - there's no reason it has to take that long. A tunnel can be established almost instantaneously. So what you want is, you want the ability to establish a tunnel on demand, I mean literally on the fly. And so there's no reason that that can't be done, and that's what I'm going to do.

Leo: Oh, great.

Steve: So it was a good exercise.

Leo: See? See?

Steve: To sort of think that all true.

Leo: You see? You see? See what I mean? This next is John Clayton in Billings, Montana. He wants to set the record straight, help us do so about Microsoft's ClickOnce and Firefox. This was that weird .NET extension that was added...

Steve: It was .NET 3.5. It appeared, people found it, discovered it, Brian Krebs writing for the Washington Post, in Firefox.

Leo: Right. And there was no uninstall button. Well, there is in Windows 7, but not...

Steve: It was disabled.

Leo: Yeah. And I, by the way, afterwards went back and read the whole, you know, there was a bug report filed with Firefox about this. And Firefox explained why it wasn't their responsibility to protect you against this, you know? It's a very interesting discussion. I encourage people to read the whole bug thread. But here's a good message that talks about it, as well.

Just listened to Episode 1999. As a .NET developer with first-hand experience with Microsoft's ClickOnce technology, I felt there were some misconceptions that needed to be addressed. It won't be winning any awards for easy implementation, but ClickOnce is, simply put, a way for packaging and deploying thick-client applications. These

applications run on your desktop like any other that you would otherwise download, not in your browser. They get an icon in the Start menu; they get an entry in Add/Remove Programs. The user is always prompted to install the

application, so there's nothing silent as you implied. ClickOnce even supports Authenticode signatures and gives a warning if the signature can't be verified, as in the case of, say, a self-signed certificate.

The plug-in that Microsoft installed into Firefox simply provides a handler for the .application files that initiate the install process. Without the plug-in, the user would just get XML. The browser wouldn't know what to do with this, you know, XML file. It wouldn't know that this is an installer prequel.

Steve: Right.

Leo: As a full-time Firefox user, if I want to install a ClickOnce application, the last thing I want to have to do is launch Internet Explorer to do it. While I'm not defending the way the Microsoft installed this update, I do have to present a Thurrott-ian argument. Paul Thurrott, of course, the host of Windows Weekly. If I look at the add-ons right now, I see that Skype, Office 2007, Adobe Acrobat, Apple iTunes, and Apple QuickTime all have their hooks in Firefox. These are all independent applications that installed stuff into Firefox without notifying the user.

If Adobe didn't install their plug-in, PDFs wouldn't open in the browser. If Apple didn't install QuickTime, the QuickTime plug-in, videos wouldn't play automatically. These work the same as the ClickOnce add-ins. They are just handling special files. And this is, by the way, this is traditional architecture for browser plug-ins. It's a BHO, basically, that's what the Internet Explorer term - I'm adding this. This is editorializing. Leo speaking. Back to the email. I should just shut up. These work the same as the ClickOnce add-ins. They're just handling special files. Skype, iTunes, and Office? Those have even less business being there without notification. My point is that when anyone else does this kind of thing, no one cares. When Apple does it, people thank Steve Jobs. When Google does it, people wonder why it wasn't there in the first place. But when Microsoft does it, all hell breaks loose.

I hope this helps you and your listeners better understand Microsoft's technology. Always look forward to Security Now every week. Keep up the good work.

Steve: And Rob, our next questioner, has sort of a different take.

Leo: Shall I just go right - segue right into it?

Steve: Let's go do that.

Leo: Okay. So Rob is near Ottawa, Canada. Here are his thoughts about reducing the security risks for ClickOnce. He says: Steve, thank you for pointing out the monopolistic behavior of Microsoft by installing a Firefox plug-in for ClickOnce that significantly reduces security for Firefox browsers. Microsoft should be shot and raked over the coals for this. You should have noted that for quite some time now there has been a third-party extension for Firefox that adds the ClickOnce capabilities to Firefox. It's called FFClickOnce. So the MS .NET update that included its own Firefox extension wasn't even necessary. In my opinion, if MS wanted to include its own Firefox extension, it should have made it a separate optional install in MS Updates. Since it did not do this and made it a mandatory installation when you installed .NET 3.5, one can only speculate that MS wanted to make its .NET framework more desirable to developers by increasing the number of users who use a browser with ClickOnce capabilities.

As you noted, ClickOnce is very dangerous. Just by clicking on something on a webpage, an application will be downloaded, installed and run. It would be easy to trick someone into clicking something on a web page that would run

with a malicious webpage. Best to disable or uninstall the MS ClickOnce Firefox extension. But if you need to use it, I would highly recommend clicking the Options button right next to the Disable button for the extension and enabling the option that asks the end user for confirmation before running the ClickOnce app. In other words, it's best to turn ClickOnce into ClickTwice. I think he's right. Much safer.

Steve: Yup.

Leo: By the way, that's the default behavior of the third-party FFClickOnce Firefox extension, however, not the default behavior of the Microsoft add-on. P.S.: ClickOnce is very similar to Java's Web Start. But unlike Java Web Start, ClickOnce is unproven technology and likely has security holes. By default, Java's Web Start apps run in a sandbox in a "restricted" mode, which means they don't have access to some system resources like local files. That's to me always been the benefit of Java over ActiveX and these other Microsoft technologies.

Steve: Yup.

Leo: You're installing an application with Microsoft that has full application privileges.

Steve: Right.

Leo: No sandbox. But publishers can remove those restrictions by signing their Web Start applications with the jarsigner tool that comes with JDK. So you better

trust the developer noted in the digital signature before running a signed Java Web Start app, too. So there's two different points of view on this issue. What do you think?

Steve: Well, I just liked them because it was lots of good information.

Leo: Yes.

Steve: We understand now that what this is doing, Microsoft is promoting a brand new standard for the industry, saying to .NET developers, rather than telling people to click this link to download the setup file, then you get that on your system. And many people are like, okay, wait. I downloaded it, but the box disappeared, and I forgot where, I mean, I've had that happen, you know, it's like, okay, where did that darn thing go that I just downloaded? And so they're trying to make this a simpler process so that you, if you want to run something on your machine, you want to install it locally, you just click it, click the button on the browser, and it does so. It's hard to say what the long-term future of that will be. But it's another thing Microsoft is doing for making their software more accessible. That is, software built with the .NET framework.

And I really do like what Rob suggested, and that is, if you're going to leave this in Firefox and have it installed, we now understand clearly what it does, that by all means, selecting that option to ask for confirmation, that makes much more sense than being fooled into clicking something, and that's all you have to do, and have this thing run. So click twice makes a lot more sense than click once.

Leo: Yeah. And what is your response to that whole thing that, well, Skype does this?

Steve: Well, I looked, and I don't have Skype running on my system, so I didn't have an add-on. I was curious what other things were there. And I did not see the list that the first questioner had. But certainly there are things, I mean, it certainly is reasonable that that would be happening.

Leo: Right. Okay. Let's move on. Interesting subject, though.

Steve: Yeah, it is.

Leo: And I think you're going to always have a debate over what's appropriate and what's inappropriate.

Steve: What needs to be automatic and what needs not. I mean, and, okay. So say that Skype asked you before installing this, given that we are correct and there is something installed on Firefox. I would opt, I think, for not having more junk in my browser. You know, people talk about how long it takes Firefox to get started sometimes. And it's having to parse and load and find and set up all these different things so that there are extensions to its core functionality. And I know that a freshly installed Firefox tends to launch a lot faster than an old one that's had a chance to collect all these barnacles all over it.

And so as a minimalist I would generally, if given an option, say no, I use Skype for this particular purpose. I don't need links to be Skype-enabled or enhanced in order to launch Skype sessions. That's not something I'm going to do. So I'd rather not have that in my system. So, you know, I would opt, and I imagine our listeners would generally opt for being informed and being trusted with having the ability to answer the question. On the other hand, I know lots of users, some who are friends of mine who are non-computer people, who, like, uh, do I want that or not? I mean, they would have no grounds for determining...

Leo: No idea, yeah.

Steve: ...even for them what the right answer is. Do I need that or not? And their sense is, it's need. Do I need it or not?

Leo: Right. I don't even know if I need it. I don't what these are.

Steve: Right. So if you told them you don't need it, they're like, oh, okay, good, thank you very much. I'm glad I don't need that. Because they're a little anxious about needing anything that involves the PC. But for some people who are, like, avid Skype users, the convenience of that would offset the cost of having it installed in their browser. So, I mean, it really is a problem because what we're doing, and here Microsoft has just taken us to another level, is we're making these systems ever more complex. We know that complexity is the enemy of security. We know that especially today because Microsoft has just patched 31 vulnerabilities that are a consequence of the complexity of the software that they are creating. So it is certainly a tradeoff. I would opt for having the choice. But that's not a choice we're being given because it causes more problems for these companies that then have tech support. It's like, well, do I need to click yes or no on this? It's a mess.

Leo: It's a mess. It is. This is always the tradeoff. We've mentioned it many times before, convenience versus security. Michael in Missouri takes issue - boy, we have a lot of unhappy...

Steve: Yeah.

Leo: But I commend you for putting this stuff in so we can respond to it. That's good, I like it. He says: Hello, I've been listening to you for some time, and I feel the need to write to you to address some issues. You claim to address security, yet you fail to truly take to heart the Linux on the desktop platform. I started out a very proud Microsoft fan myself, and I understand that many users are Microsoft users.

But as a technical person I learned the true power of Linux and other open source technologies. Let me make it clear, Ubuntu can be used with no experience or knowledge. However, Linux provides a degree of control that

you just can't get with Windows. Linux is modular, does not hide its code in 1's and 0's. You know what you're getting when you use Linux. Please do not dismiss me as a Linux fan boy. I do program, and I understand the choices that both systems make. But even Microsoft declares Linux as its competitor. Microsoft CEO Steve Ballmer admitted Linux is a serious competitor. In any case, I just wish you would truly look into what Linux has become. Even spend an episode addressing it as a security option. You did so with Windows 7.

The next problem I have is with you, Steve Gibson, not making your code open source. You can do whatever you want with your code, and it's nice to keep some code closed source. But I just can't get over your reason for doing so. In one of your episodes you said you don't make your code open source because you were worried it would allow hackers to use it. To me that seems arrogant, to assert that your code is above any others without merit or reason. However, in a security sense, it seems you're condoning security by obscurity - something we mock. I thank you for your podcast, and I enjoy listening to them. I just wish you'd be a little more open - capital O - minded.

Steve: Okay.

Leo: Can I say one thing?

Steve: Yeah.

Leo: From day one we've always said this is a Windows security show. I mean, we really focus on Windows because most users are using Windows. We don't cover security on the Macintosh particularly. We haven't covered security on Linux particularly. We do have other shows that talk about those subjects. In fact, we're going to talk about Ubuntu next. Right after this show we've got Jono Bacon from Ubuntu to talk about [indiscernible]. So we do have other shows that cover this stuff. I don't know if this is something you consider part of your portfolio.

Steve: Well, it's not. And for a couple reasons. I mean, there is a limit to my scope, to what I am able to cover. And I live in Windows, so it's accessible to me. The way I deal with things like Linux is we spend a lot of time on fundamental technology, on the way things work, which is universally applicable to all platforms. You know, when we've talked about crypto, that's not Windows crypto.

Leo: Sure, sure.

Steve: It's as much Linux and Mac crypto as it is Windows crypto. It's fundamental stuff. So there's - I recognize the validity of what Michael said. However - and I also know that there is certainly a Linux population who listen to the show. And I can only say, well, as you said, Leo, we're primarily a Windows security-focused presentation, but also heavy on fundamental technology, fundamentals which are universal and apply to everyone, which is why I think we've got Linux users who are listeners.

Leo: Right, right. We're Windows except when we're not.

Steve: And as for my source being open code - my code being open source - maybe someday. I mean, I can easily foresee a point someday when I'm no longer a commercial producer of code, that I'd be happy to share my stuff openly. But this is all, you know, SpinRite is a commercial program that is my bread and butter. CryptoLink I hope will grow into the same thing. There was some discussion even of this forthcoming DNS benchmark, which has turned into just a beautiful piece of work that I'm going to be very proud of and have a lot of fun sharing with our listeners as soon as it's finished. And somebody posted in the newsgroups, Steve, you're doing this thing, and it's freeware. Why aren't you making it open source?

And Michael did misunderstand something. It certainly wasn't arrogance, me believing that my code is better than anybody else's, that's why I don't want other hackers to use it. What I was concerned with is it would be very simple for malicious versions of my code to be created if here was the whole package, and you assemble it, and you get something that looks exactly like what GRC produces. Then they could stick their own stuff in and say, oh, look, this came from GRC. Obviously you can trust it because it's from Steve. Now, it's certainly true that someone could start from scratch and create something that looks exactly like mine, but that's substantially more work. I mean, that's a lot more work than just adding some little bit of something to an existing body of code. Also the fact is, the fact that it's all in Assembler means that it's much less accessible to a large body of people. So I wonder how much use it would be as open source.

But more than anything, I'm not done leveraging my work for the support of myself and Greg and Sue. And so the DNS benchmark is free, but I regard "free" as separate from "open." I mean, it is, it's been a labor of love. Fundamentally it's all I've done to this point this year. So I'm hoping it's going to be useful and valuable. People will come to GRC in order to get a copy. And some will go, oh, look, SpinRite, that thing's still around. I'm going to read some testimonials, and maybe they're going to buy a copy. And so that's just really been the model for GRC for a long time. And thus ShieldsUP! and all the other freeware that I've produced, and all the things I give away, we're still a commercial enterprise despite the fact that I'm trying to do everything I can to give back to the community.

Leo: Would you consider, because I would like to see your code, open sourcing end-of-life applications? Stuff that's obsolete? Or are there macros in there that...

Steve: Yeah, there's still, I guess...

Leo: ...you don't want to - proprietary code that you don't want to release.

Steve: Yeah, I do feel proprietary about it. It's stuff I've developed carefully and painfully that - and often put a lot of time into that makes my stuff small and fast and special. So...

Leo: Yeah. No, this is frequently the case, even though maybe that program is not in use, there's libraries in there that you still use that you don't want to release and things like that. I don't think anybody should be - I don't think there's any onus on anybody to do open source. If you choose to do open source, that's great. But no programmer, there's no - I think it's actually inappropriate to kind of urge programmers to release their stuff.

Steve: Well, and I think...

Leo: That's a choice you have to make.

Steve: I think the other thing that's a little confusing, certainly this was sort of the genesis of the comment in the newsgroup recently, is what I have is a very, I mean, a phenomenally open development process. I mean, it's already, if you were to look at GRC.dns, which is the newsgroup at news.grc.com, you'd see literally thousands of postings with us interacting, me interacting with a group of interested people who are saying, hey, Steve, right-clicking here really ought to do this. And I think about it, go yes, that's a very good point. And now the app has that.

And it's now locale aware for people - it'll export a CSV, a Comma Separated Values file, so you can save all your benchmark results. And basically they've been responsible for hugely broadening the scope of what I was going to offer. They've also been responsible indirectly for this thing never ending. But we're still getting closer. And I am really pleased with what it is because I think a DNS benchmark is going to be very valuable for people.

And I know that the same thing is going to happen with CryptoLink. We'll set up a newsgroup. Actually it's already there, GRC.cryptolink. And people will say, hey, I have this need. And if it looks to me like that's something I hadn't thought of, that more than this one guy has, it's like, oh, good point, I'm glad you brought that up because I'd rather know now than later. And so it's a little confusing that the development process is so transparent and so open. And, I mean, it's fantastic for me because I'm able to just churn through, I'm able to implement things and say, okay, here's a new version. And these guys look at it, and they pound on it, they go, yup, you got it; or, oops, you forgot this over here.

And so it's an amazing environment for developing. Yet I'm the final arbiter. I'm not sharing the code. I'm not sharing the decisions. I'm taking responsibility for it. Ultimately what I produce is mine. Yet in the case of the DNS benchmark I'm making it freely available for everyone. So it's a model that we've proven, and it's just fantastically productive. I really like it. And boy, I never get things finished because everyone keeps wanting new things. But it just ends up getting better and better.

Leo: Right.

Steve: And I know that that's going to influence CryptoLink's development, too.

Leo: Good. Mike Potts, Columbus, Ohio has a recommendation for teaching programming to young people. We've been talking about that a lot because I just have the fantasy that I might sometime want to teach a class on that to my kid's high school.

Steve: I think that's going to happen.

Leo: I have a feeling it might. Hi, Steve. This note is more for Leo and also for other listeners who want to help younger people who are learning to program. I like an open-source program called Basic-256. It's available at kidbasic.sourceforge.net, means it's open source. Pre-built Windows binaries, source-code, and Ubuntu packages are available. The article "Why Johnny Can't Code" on Salon inspired Basic-256. This program is easy to learn and use. There are reasonable tutorials available. Some sleuthing in the source material will find some nice extras.

But really what sets this program apart is that it has a small graphics window. The programmer is given an approximately 300x300 pixel graphics window with all 16 colors, and enough primitives to do some real work - circles, rectangles, dots, et cetera. Sure, you could teach the classic algorithm of "have the computer guess your number between 1 and 100," which is my favorite beginner problem. But nothing generates more excitement - and I agree - or holds their interest more than having them generate their own primitive civil war, dodge ball, or soccer game.

This program held my 14 year old's interest for several months. What he really learned from this experience was one very important programming lesson: The computer did exactly what he told it to do. Not what he wanted it to do, but what he said to do. You can always get more experience, but I don't think you can get a deeper lesson than that. That's true, that is the fundamental lesson of programming.

Steve: Yeah.

Leo: Computers just do what you say. You can't get mad. They just did what you told them to do. When Basic-256 was starting to wear off, I spent a couple of days explaining, book-only, processor memory cycles and 8-bit Assembly language to him. I didn't think of looking for an emulator, but maybe now I'll look for that PDP-8 emulator you guys found. Thanks for a great netcast. My copy of SpinRite=1, bad_harddisk=0.

Steve: I took a look, and I really think this looks pretty neat, Leo. You may remember the other sort of graphic-enabled learning language was Logo.

Leo: Right, Turtle Graphics, yeah.

Steve: Turtle Graphics. And what it offered was that immediacy of feedback. And we were talking about the Tower of Hanoi relative to the PDP-8 and the teletype going chug-chug-chug-chug chug-chug-chug-chug, you know, putting out X's. But this thing, this Basic-256, it's a really nice integrated environment where you have code, you write code on the left, and then in the upper right are all your variables. So you're able to see, which I think is really important, the current state of all the variables in real time, that is, at that point. And then below is a graphic window. So you're able to write some code that says circle at this location of this size, and it goes blunk, and there it is. I mean, and that's like, hey, cool.

Leo: Yeah.

Steve: How about, how, how would a loop work, where I change the X and Y coordinates, and suddenly [vocalizing], and you've got circles all over the place. And again I just - I wanted to share this because I think it is an interesting, very accessible way of, like, letting people play with a simple programming language that modifies variables and has loops and actually does something visual.

Leo: There are a lot of debates, you know, as I started researching this I found so many debates about how to do this, what's the best way to do it. And I think you're right, I think you can't knock getting a kid excited some way or other. Certainly that's how we did it.

Steve: If it's too dry, you're going to lose a lot of people because they're going to go, oh, who cares about...

Leo: Every single one of us in our generation got excited because, you know, learned because we got excited by usually small computers - in your case a PDP-8, but my case an Atari, many people Apple or Commodore - and figured it out on our own. We weren't taught, but we figured it out. But I think now that's the debate is, well, is that the best way to start out? But I think you're right, if you don't get excited about it, if you're not inspired and turned on by it, there's no point...

Steve: And also, Leo, times have changed. Back then a teletype spitting out X's was...

Leo: Was exciting.

Steve: That was a lot of - that was sensory overload for us back then. Now...

Leo: These kids are a little different, yeah.

Steve: I don't think you can possibly sensor overload a modern teenager these days.

Leo: No, you're right, you're right on that one.

Steve: So it takes a lot more to get there, to get to the same level.

Leo: Well, thanks for the input. I really appreciate...

Steve: Yeah, do check it out. I think our listeners should, too. Anyone who's interested in, like, dabbling around with simple programming, this Basic-256 is very cool.

Leo: I will take a look. Greg M. in Fort Wayne, Indiana wonders about expired SSL certificates. Danger, danger, Will Robinson. You wrote that in. But I had...

Steve: No, he...

Leo: He wrote that in?

Steve: He wrote that, yeah.

Leo: Danger. By the way, somebody took your backwards...

Steve: My "sna-na ba-na-ni, pa-shor-yor-nar-ros"?

Leo: And sent me the reverse. And it did, it was perfect. I was very impressed. Sounded just like an alien. Hi, Steve and Leo. First, I want to think you for a great podcast. You helped me to ace my Security+ certification. Yes. The podcasts on encryption came at a perfect time.

After listening to the SSL and TLS podcast, a question came to mind that I do not recall your ever talking about. Neither was this covered in my Security+ class. What about an expired SSL certificate? Not that I'd use a site with an expired certificate, but many everyday users might just click by any warnings and continue on. If everything else with a cert is okay, would a user's data still be encrypted? Oh, that's a good question. Thanks for the podcast. Keep up the good work. What happens when they expire?

Steve: A really good question.

Leo: Yeah.

Steve: We've never talked about ever exactly that, this notion of expiring certificates. I'm annoyed by it because it seems primarily designed to guarantee revenue to the certificate issuers, number one of course in the world being VeriSign. And believe me, every two or three years they get more money from me. However, there is nothing insecure about an expired certificate other than the fact that it has expired. I have visited websites where I got an expired certificate warning. Now, I'm immediately worried about, okay, well, what does this mean about the management of the website that they would have an expired certificate? If I check, and when I have, normally it's only been expired like that day, or maybe the day before. So it's not like they've survived a long time. They're probably scurrying around...

Leo: Oh, boy, it expired, we forgot.

Steve: Exactly. They're, like, trying to cram an emergency recertification through VeriSign or whomever as quickly as they can in order to replace it. But the certificate continues to work. It is just as secure. Now, I'll add a caveat to that to say, well, technically the longer something like this exists, the greater the chance is for it being hacked.

Leo: Right.

Steve: So in theory you expire these because, well, also there's that the certificate implies a representation of you, your existence, your organization, your location, your going concern, which is being made by the signer of the certificate. So the signer, in my examples often VeriSign, is saying, well, yes, we've checked you out, and you're breathing, and you're a good person, and you were a Boy Scout once, so we're going to sign the certificate for you. But if you suddenly go to the dark side, within a couple years this certificate will have expired. So our representation about you is time-limited.

Leo: It's good security policy. You know when I create a PGP certificate, I usually say expires in a year. Frequently the passwords that people assign in a corporate environment will expire in a month or two months or three months. That's not a bad thing.

Steve: No. And again, exactly, so stale information that's getting older has just a greater chance of exposure, greater chance of leaking out, getting out and so forth. So there's some reason to understand, I mean, there's some justification for certificates expiring. I would just say that, well, to answer Greg's direct question, there's nothing in any way more dangerous. I would say consider the circumstances, consider the website, maybe ask or determine why it's expired, how long it's been expired. But it actually can be that expiration can sneak up on the owner of a certificate. I'm knocking on wood here that that's never bitten me. We've always had plenty of time.

The other thing I will mention, I've been worried about renewing a certificate well before the expiration date, wondering if I would get credit for how much earlier I am over the expiration. The good news is I've confirmed I do get credit. So sometimes I remember, before I was sure of that, I'd wait till not many days before expiration because I was already pissed off that I was having to pay so much money so often for, like, nothing, for some bits, essentially. But now I know that when I get my first email notice from VeriSign that, hey, you've got some certificates coming due here, it's like, okay, good, do it now because they're going to extend me from the actual time of expiration, not the time of renewal. So for people who have certificates, that's something good to know.

Leo: Moving along. Jonathan the IT student from Roseville, California doesn't like NAT. It sounds like a kid's book. Jonathan doesn't like Nat.

Steve: Get those gnats off me.

Leo: I just listened to episode 199. And while I liked the episode, I have to take issue with what you said about IPv6 vs. IPv4 with NAT. While NAT is an acceptable stopgap before we run out of IPv4 addresses, it is not an acceptable permanent solution. We said that IPv4, we were running out of numbers, but thanks to NAT it's not the critical situation it was before. The problem with NAT is that it breaks up the Internet into segments, while the purpose of the Internet is to bring people together through the computer. Not through my router, however. But all right, we'll get to that. Essentially it cripples, while not breaking, the Internet. For example, my brothers and I like to play Battlefield 2 together in Co-Op mode. To do so one of us has to act as the server and the others connect in through their IP addresses. My parents' house, however, is behind a NAT router at the ISP level, so there is no way they can serve as the server. With me playing behind my college campus NAT, fortunately I have at least one of us, at least one of us is on the Comcast network and has a public address. NATs are not a solution, they are just a patch. Is the IPv6 solution better than NAT?

Steve: Well, okay. This brings up an interesting point, and something that I failed to mention when I was talking about this last week, and Jonathan's note reminded me. And that is that another aspect of this issue of addressing really goes back to sort of the original Internet sort of UNIX guru purists. These are people who fundamentally believe what Jonathan is saying.

Leo: One CPU, one IP address.

Steve: Yes. That every single machine on the Internet should be available by a unique IP address. I mean, that's like - that was the fundamental concept in the beginning was every one of these machines would have an IP address, and this amazing routing architecture with packet forwarding would allow by this incredible technology any machine to reach any other. These are the same people who are really annoyed by the notion of stealth because, oh, that's breaking the fundamental structure of the Internet.

Leo: You're talking about the thing - you coined this term, the idea that a router might not respond to a ping from another router.

Steve: Yes. Yes. Or, exactly, or, well, yes, precisely. And then the idea that an ISP would be blocking the ICMP because they don't want their internal networks to be tracerouted. And again, the purists are, like, infuriated by this. It's like, wait a minute, that's - any machine, any router, any hop, any endpoint needs to respond to ICMP because that's the underlying low level intermachine glue that allows us to figure out what's wrong with the Internet, where something's broken and so forth. And, sadly, this is not the way the world turned out.

What they're saying is, that purist view says everybody on the 'Net is a good guy. And everybody has everybody else's best intentions and doesn't wish anybody ill. Unfortunately, we really know that's not the case. So here's Jonathan, for example, with his example in Battlefield 2. He's got three people. He would like them all to have access to each other. Well, what he's saying is, then, we need no routers and no firewalls.

Because if you have firewalls you still have to configure the firewall to allow access in. Just as you could, you know, he's saying, well, his parents' house is behind a NAT. Well, although actually he said "NAT at the ISP level." Now, that is a particularly extreme form of problem because that means that his ISP is giving him probably 10-dot addresses or, like, all of their customers, or maybe 17-dot or 192.168 or something. So that's a problem because there's no NAT for them to configure.

So but in the normal case with a residential NAT router, well, okay, it's a firewall. And so if you want incoming penetration, you need to configure it, just like you would a firewall. And we now know that there's just no way that Internet use is safe without a firewall. That is, your system cannot be wide open to the entire Internet. So again, unfortunately, reality impinges on the Internet UNIX gurus' original immaculate conception of what the Internet could be. And we know that the 'Net has lots of bad people who do not wish us well. And we need to protect ourselves from them.

Leo: Is it possible that you could have that kind of an open system and still be secure, I mean, with other techniques?

Steve: Well, you could certainly have per-machine addressing, i.e., or la, IPv6. You could have per-machine addressing. But you're absolutely going to have a firewall. I mean, to not have a firewall means to trust, I mean, absolutely trust all of the code that you've got running on your machine, I mean, which are increasingly complex today. One of the things I don't do when I'm running through security news is talk about the 5,000 other applications per week that have security problems, I mean, because they're not major problems for people. Our OS is our major problem. But, I mean, all kinds of other security problems exist. But we don't have time if we broadcasted 24/7, Leo, for talking about all the other problems.

So I just - with our systems being as complex as they are, I can't imagine it being safe not to have a firewall protecting it, that is, to allow a system to be wide open. It would mean that any applications that people used that were receiving packets would have to be secure against everything. I mean, Battlefield 2, if it's acting as a server, lord knows what kinds of security problems it has. I guarantee you it has them. But because its exposure is limited, thanks to the fact that it's behind NAT routers and behind firewalls, and people have to go to some pain to configure it, that pain is allowing access in a controlled fashion. If it were uncontrolled, all hell would break loose.

Leo: We wouldn't want that.

Steve: The world is just not - it's just it's not the way it was originally conceived when the Internet was born.

Leo: All right. Are you ready for the Tell-Tale TPM? Jeff in Michigan asks about that. He says: Hi, Steve and Leo. Thank you so much for your great podcast and the wealth

of information, security and otherwise, every week. My question is if there's an easy way to tell if your computer is equipped with a Trusted Platform Module, or TPM. As I understand it, TPM is required to use the Windows Vista/ Windows 7 full-drive encryption. When I tried to enable this feature on my HP tx2500 laptop - that's fairly old, I think - it complained about not having a TPM available and

would not continue. For some reason I thought that almost all computers in recent years - oh, no, I'm sorry, this laptop is less than a year old, he says - have shipped with TPM, but I guess I'm wrong. Since I only wanted to play around with the encryption, it wasn't a big deal, and there's always the great whole-drive encryption available

through the free True-Crypt we talk about all the time. Anyway, do either of you know if there's a handy utility

or website that will help me figure out if a computer has TPM available without poring though the chipset and motherboard specs?

I've been a listener of Security Now! since your episode on Internet Anonymity and have loved it ever since. Thank you again for all the work you put into the show. It's worth every second. Signed, Jeff.

Steve: Well, here's a few more worthwhile seconds. Every single machine I have ever seen with TPM has it disabled.

Leo: Because it would be problematic if you didn't know it was enabled. Is that right? No.

Steve: Frankly, I don't think so. But for whatever reason, I mean, I can't think of any good reason to always have it disabled. But every machine I have, every laptop I have, and I've got a bunch, and also desktops with TPM on the motherboard, if you go into the BIOS, typically under Security, and then it'll have maybe trusted platform or some lingo of some sort that will sort of guide you there, you'll find that it is turned off. So you have to go through some - you have to basically just mess with the BIOS. So Jeff in Michigan, what you want to do, you don't need to really do anything except get into your BIOS by F2 or Delete or F1 or whatever it is you do to get into the BIOS, and poke around under the Security section, which most BIOSes these days have. You'll probably, if your system has TPM, that's where you'll be absolutely definitively determine it.

And if it's off and disabled, as it will have shipped almost certainly, then Windows doesn't know anything about it. It hasn't installed the TPM driver. It has no information that there's a TPM underneath that is disabled. It's completely hidden until you enable it in the BIOS. And normally it's a multiboot process. You first enable the TPM. Sometimes you then have to reboot after saving the new configuration. Then you need to initialize the TPM, which is sort of a flushing out and cleaning out process to sort of put it into a known state. So you do that.

And then when you get back into Windows, Windows will pop up and say, oh, new hardware is found, Trusted Platform Module. Well, what do you know? And normally there will be a companion driver that came along with the laptop. Maybe it's already installed. You may want to check and see if you can find a Trusted Platform Module driver for the particular laptop or desktop model you have because you probably at some point need to let the system sniff that driver in order to incorporate it into itself. And then you're good to go.

But in every case, for whatever reason, these things are disabled when they ship. You need to go in and manually say I want Trusted Platform Module services. Turn them on in the BIOS. Then they'll surface in the OS, and you can precede from there.

Leo: Yeah. I wonder if there is a tool that would tell you. If it's disabled, it wouldn't tell you anyway.

Steve: There are probably...

Leo: Go into the BIOS. If you've got TPM, the BIOS is going to say "Turn on TPM," right? It's going to have some switch like that.

Steve: Yeah, well, it does, absolutely. Every BIOS I've seen makes it very clear. You just have to dig around in there, typically under security issues, and it'll be an option there.

Leo: There you go. Go into your BIOS, look in the Security tab, and see if there's anything about TPM. That's a good fix.

Steve: Yup.

Leo: Don Daniels in Evergreen, Colorado discovered something about Skype and Universal Plug & Play: Dear Steve, I was poking around through the settings on Skype, the current version 4.0.0.226, and noticed in the Tools, Options, Advanced, Connection tab there is a block that says "Enable uPnP" that is checked by default. Panicsville! I unchecked it, but then thought, well, I'll try an experiment. So I re-checked it, shut down and restarted Skype, ran UnPlug n' Pray, and it showed no problems. It said, "UPnP is safely disabled." Oh. I see a problem here.

Steve: Yup.

Leo: What is going on? And thanks, Microsoft, for confusing this issue. What is going on, and does this default setting in Skype open us up to any security threats? Don Daniels, listener since Episode 1. I found you at Episode 18, but I went back and caught up. So this is really a confusion of terminology is what this is.

Steve: It's got confusion coming and going here. First of all, when he referred to UnPlug n' Pray, he's talking about one of my closed source, but free, security bits of freeware that I wrote a long time ago in response to a very bad security vulnerability in Microsoft Windows Universal Plug & Play service running in Windows XP before Service Pack 2, when most people didn't yet have a firewall running. And so there was a remote code exploit which bit all kinds of people, where you could simply receive a packet at your Windows XP system, and because Microsoft's own Universal Plug & Play service was vulnerable, it was a remote code exploit. You could get your system taken over.

So I wrote UnPlug n' Pray, which all it does is stop the service. It just disables it and then tells you, okay, that service is safely disabled. And at the time, nothing was using Universal Plug & Play, so it was unfortunate that it was on, open, running, and vulnerable. Okay, so what's happening with Skype is different, but it's also a concern. What you're doing now, and this is an example of sort of Universal Plug & Play beginning to come into use, Skype is saying, hey, you know how Steve and Leo have ports mapped through their routers, so they're able to get really clear connections. Because Steve and Leo have ports mapped through their routers for Skype, they never use a relay, a third party to relay their traffic. Remember we were just talking about NAT traversal. There are some situations that Skype cannot handle. So in that event, if both of us are behind NAT routers without any ports mapped, and telling Skype to use those mapped ports, Skype may have to bounce our traffic off of a third party that's somewhere out on the Internet that we can both reach because we can't reach each other.

Leo: They call those the supernodes; right?

Steve: Well, exactly, those are so-called Skype supernodes. So what Skype has now done with this most recent version, I don't know how far back it goes, is they've said, hey, routers are generally Universal Plug & Play aware. And we've talked about that, how routers enable UPnP also. So if the router's enabled when Skype is started up, it's able to send a broadcast out onto the network to say, hey, what Universal Plug & Play guys are out here? And basically every Universal Plug & Play device on the 'Net will enumerate itself. It'll report in and say - and one of them will be, hey, I'm a router. And Skype says, hey, great. I got a little port I want you to open for me.

And so without you doing anything, and remember this is all default, so Skype's default was enabled, your router's default is enabled. Without you doing anything, Skype has opened a port on your router which is good for Skype, and it's good for your Skype connections. It means, hey, look, I upgraded to a newer version of Skype, and somehow my connections are better. What's happened is it's allowed Skype to receive traffic through that port to itself. That requires that Skype has that option enabled, and your router has that option enabled. Then they both get together. And without you knowing it, they do this port mapping through your router.

So that's what that's about. I can't say that that's a huge safety concern. I hope that when you shut down Skype it closes the port behind itself; that if it's not using it, it doesn't leave it open. We hope of course that Skype doesn't have any bad security problems because what we are saying then is that this incoming port is always going to be forwarded to Skype running on your machine. If there were a security problem, then that could be a security exposure. On the other hand, in order to get the kind of communication clarity that you want, it's necessary to let two Skype endpoints find each other without going through a relay. And this certainly does make it much easier to have that happen without needing to go do it manually.

Leo: So we have, for best results, that's what we do. We have a dedicated port. I use 22222. And so you can use any arbitrary port, usually I would say between 1024 and 65535. But...

Steve: Yes, it wants to be up in the client port range.

Leo: Higher side, yeah. So that's why I use 22222. That's not claimed by anything. And so that's the same idea. And then I have to, by the way, port forward any incoming traffic from the router to that machine, the Skype machine.

Steve: Correct.

Leo: And then in theory we are both preventing being a supernode, which could impact your performance, and getting a dedicated port, which should give us better results; yeah?

Steve: Yes, exactly. We have a non-relayed connection because our Skypes that are - Skype uses a trusted third party, that is, there is Skype Central which does all the so-called presence management.

Leo: Right.

Steve: When I start up my Skype, it shows me, of my contact people, who's online. So that happens by me logging into Skype Central and sharing my contact list, or maybe Skype maintains it for me, who knows in detail how it works. It doesn't really matter. But one way or the other it says, oh, here are the people that are also currently logged on that you know. So it tells me about that. Then it also knows if I have opened a port for it to use. So when somebody wants to connect to me, like Leo, when we're initiating this recording and you want to connect to me through Skype Central, it tells your Skype that you're able to reach me on a certain port. So your Skype sends its traffic to my specially pre-opened mapped port, which comes directly to me, and we're able to get a non-relayed communication.

Leo: So that's a good thing.

Steve: That's a good thing.

Leo: And that's not any - that's not a security hole unless, as you say, Skype has a flaw.

Steve: Yes. What it means is that incoming traffic will be received by Skype. And we want to believe and we hope that Skype was written carefully and securely, that it, for example, isn't listening to that traffic when we're not actually in a communication, in a conversation. I'd be much more comfortable if the packets were just bouncing off of my machine and there was nothing listening, than having, like, Skype always ready to receive something because that creates a larger security target.

Leo: Our last question. Are you ready, sir?

Steve: It's a quickie and easy one.

Leo: Jeffrey Dunn in Riley TWP...

Steve: And I don't know what that is. I Googled it, TWP, where is that? What is that? That's where he said he was.

Leo: Don't know. I'm sure we'll find out.

Steve: Yeah.

Leo: Anybody in the chatroom have an idea? What is TW - oh, Township. Riley Township.

Steve: Okay.

Leo: Thank you. We have a lot of - I tell you, our chatroom...

Steve: Smart people.

Leo: That's my brain. They're good. Yeah, Riley Township, we don't know where that is, but it's somewhere they have townships. Which could be almost anywhere. Wonders about BRIEF and syntax highlighting. Steve, I heard you mention that you use BRIEF, which is an ancient, I mean ancient, text editor. I was curious, does your version have syntax highlighting? It had been a while since I used BRIEF, but I dug out the floppies and have started using it again. I forgot how productive I could be with it. This is a DOS program.

Steve: Yes, it is. And it will not run in Windows 7 because...

Leo: Oh, no.

Steve: Yeah.

Leo: So finally it comes to the end of its life.

Steve: Not yet, because I'm not running Windows 7 yet, either, so.

Leo: And may not be for some time to come.

Steve: Long time ago, far...

Leo: This was a programmers text editor that Steve still uses.

Steve: Yes. I mean, it's even an acronym for Basic Reconfigurable Interactive Editing Facility.

Leo: Because it's totally configurable.

Steve: Oh, it is. It's phenomenally. And I've completely reconfigured mine. However, syntax highlighting is a fantastic thing to have. And editors at the time did not offer it. There was a piece of freeware that came out called Colors. And it's interesting because you would run it, and it would kind of go resident in the same memory as other applications that would then load on top of it. It would turn around and then load BRIEF. But it had hooked a whole bunch of so-called interrupt lines to the BIOS for the keyboard, for video output, for the timer and a few other things, so that it was able to determine when text changed on the screen. And when text changed on the screen, it would quickly go up, check to see what had changed, check a configuration file that the user provides, and color it. So essentially it provides sort of third-party instantaneous coloration of text in a DOS box. And it works perfectly. I've been using it for nearly 20 years.

Leo: Wow.

Steve: It was freeware. Excuse me?

Leo: Wow.

Steve: It was freeware. But I was so in love with it that I didn't want ever to not have it. So we paid, as I remember, a couple thousand dollars for the source code.

Leo: You're kidding.

Steve: No. Because, I mean, this thing was so cool that I thought, well, if it ever - if there's something I need that it won't do, I want to be able to do it.

Leo: What's it written in, C?

Steve: I don't even remember. I never compiled or assembled the source. I never needed it because the thing has worked perfectly all this time. I mention all this because, being free, I can let people play with it. And so it is available. Several other people, listeners and people in our newsgroups, have asked, hey, Steve, I love the screenshots you've shared of the way your code looks. I want mine to look the same way with colors.

So the URL is GRC.com/miscfiles, as in miscellaneous files, /colors.zip. And in there - I spent a couple hours last week putting together some documentation just out of my own memory and gave some examples. The configuration file that I use is there. The little couple hundred K, I don't even think it's that big, Colors EXE is there. Again, it's free. The source is not there because that we purchased, and I don't have the author's permission to share that. I don't even know who it's written by. I looked through the EXE trying to find any name or anything. But that's been lost to history. But I do know that it was freeware. So GRC.com/miscfiles/colors.zip. And anybody who wants to play with it is welcome to do so.

Leo: Wow. That is great.

Steve: And it colorizes DOS boxes.

Leo: Really cool.

Steve: Yeah, it's really neat.

Leo: Really cool. And of course any modern text editor does that without any additions. But you don't have all the key bindings. You can't use dot commands and stuff like that.

Steve: Well, and nobody supports Assembler. Imagine that, Leo.

Leo: Nobody? Nobody supports Assembler?

Steve: Nope. Nope.

Leo: Oh, that's interesting.

Steve: Yeah, well, because no one...

Leo: So you do need that.

Steve: ...[indiscernible] with crazy people, yeah.

Leo: Huh. Well, that's kind of intriguing. Just the crazy folks.

Steve: Just the crazy - just us crazy people.

Leo: I think more people should write in Assembler.

Steve: Well, actually there's been a little bit of a groundswell of interest as a consequence of me having talked about it as much as I have.

Leo: Oh, that's nice to know.

Steve: So, yeah.

Leo: That's great. All right, Steverino. We have come to the end.

Steve: Episode 200 is behind us.

Leo: 200 episodes. It's exhausting.

Steve: Eight more, and we wrap up year four at 208.

Leo: I don't know which is a bigger deal, year four or 200th episode. Anyway, congratulations. I'll toast you with a glass of Cab.

Steve: Sounds great. Now, hey, speaking of which, I wonder if that 208 episode is you in China because that's two months from now.

Leo: No, no, I'm going to China in about three weeks.

Steve: Oh, okay. Then we're going to have to be doing double episodes here soon so we can continue never to miss one.

Leo: We will. I'm going to be in China July 2nd through the 18th. And while we will miss some shows, we're not going to miss this one because Steve's unswerving commitment to never missing an episode means that we'll have to double up before I leave. So if you do watch this live, and we do do it live every Wednesday at around 11:00 a.m. Pacific, that's 2:00 p.m. Eastern, 18:00 UTC, and probably the next couple of weeks we'll do two in a row.

Steve: Yeah, actually I have a note here. I just fired up my little Post-it Notes app. It says Security Now! double recordings June 24th and July 1st.

Leo: Okay. So we're going to get right up to the edge there.

Steve: Does that sound right to you?

Leo: That does sound right because I leave July 2nd.

Steve: Perfect.

Leo: Thank you, Steve.

Steve: Okay, my friend.

Leo: Happy 200th. And we'll see you all next week on Security Now!.

Steve: Thanks, Leo.