SECURITY NOW!

Transcript of Episode #32

Listener Feedback Q&A #5

Description: Steve and Leo briefly review last week's topic of symmetric stream ciphers, then pose the first Security Now! Puzzler/BrainTeaser which proposes a secure means for sending encrypted messages where neither party knows the other's key. The Puzzler/BrainTeaser will be answered and resolved at the start of next week's episode. Then, as always in their Q&A episodes, they answer questions and discuss issues raised by listeners.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-032.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-032-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode 32 for Thursday, March 23, 2006. Your questions, Steve's answers. Hello, Steve Gibson.

Steve Gibson: Hey, Leo.

Leo: Leo Laporte in Northern California, Steve Gibson in Southern California, and we meet in the middle via Skype.

Steve: Which is really working well for us.

Leo: It is. It's remarkable. Has something to do with the fact that you have spent some money on a good Heil PR-40 mic and an M-Audio interface.

Steve: Oh, I have a very nice microphone, yes.

Leo: Bob Heil says hello, by the way. He's going to be at NAB, and they're throwing a big party this week, or I guess it's next week.

Steve: It's so funny, I watch Letterman, and he's got the Heil PR-40 there. And he's banging it with his pencil all the time. And it just bugs me to see him, like, whacking on it with his pencil. It's like, David, stop that.

Leo: That's the way David is. He has a mic - they use a Shure mic for his audience warm-up. Because, you know, he comes out before the show.

Steve: Right.

Leo: He does an audience warm-up, and it's on a long cable. And it's all dented up. You know, it has the round ball, the wire cage on the top of it.

Steve: Right.

Leo: It is messed up. And I thought, why is his mic all dented up? And then I saw. He hits things with it. He bangs it around. But that's part of his act is, like, banging on this stuff.

Steve: Right, right.

Leo: It's pretty funny. So we talked about crypto last week, and I know we want to tie up some loose ends before we head into questions and answers.

Steve: Well, yeah, there's a couple things. First of all, people are wondering whatever happened to the OpenVPN guide that I have promised. It's on the way. I haven't forgotten it. I mean, it's like my number one thing to be working on. I had a bunch of, you know, year-end tax stuff to deal with, you know, financial, over on my GRC business side. But also I'm working with a very talented developer in Russia who's got a bridging driver. And we're basically coming up with a custom solution for OpenVPN which solves a number of problems that dramatically improve the robustness of its use and make it much easier to configure. So I haven't been sure how that was going to proceed. You know, he's got other consulting he's doing. I've got, like, an 18-page template laid out. I'm fleshing them out. You've, in fact, seen a couple of the graphics that I sent you, Leo.

Leo: Yeah, they look good, yeah.

Steve: Anyway, I just want to let people know I haven't forgotten about it. It's going to happen. And it's going to be just a tremendous reference once it comes together. But anyway, it's on the way.

Leo: He's working on it, folks.

Steve: I'm working on it.

Leo: Patience.

Steve: We had a guy referring to last week's discussion of - remember you asked me, Leo, when was that secret decoder ring kind of idea first used, you know, who did it first? And I don't know that this is accurate. But someone said that Julius Caesar...

Leo: That's right.

Steve: ...did use it to communicate, you know, basically used a simple transposition cipher where letters were shifted some number of places, which is exactly what two concentric rings do, in order to communicate with his generals.

Leo: That's why they sometimes call it a "Caesarian cipher."

Steve: Right.

Leo: Yeah, yeah. So it's pretty old. In fact, I bet it predates that, even. I mean, we know he used it. But I bet it predates that. It's kind of an obvious thing once you have an alphabet.

Steve: Yeah, or if you asked your gurus to come up with a way of, you know, easily encrypting data that a general who's not too clueful would be able to decrypt. I mean, it couldn't be too complicated or some guy on horseback wouldn't be able to figure it out, so...

Leo: Some barely literate guy on horseback.

Steve: Yeah, exactly.

Leo: All right.

Steve: Well, now, we're going to announce today the very first - and I'm very excited about this - Security Now! Puzzler/BrainTeaser that relates to cryptography that comes from something last week. A guy wrote from the United Kingdom, Roger Cuthbert. He said, "Hey, Steve, I had an idea about one-time pads." He said there's an old, I don't know if it's a riddle or a trick or what, where you want to send somebody something securely using a padlock that he doesn't have the key to. So you put it in a box, and you lock it with your own padlock. You send it to him. He adds his padlock to that, so now the box is doubly locked. He sends it back to you. You take off your padlock, so now it is still locked only with his. And then you send it back to him, he takes off his padlock and is now able to open the box.

So what was so clever about Roger thinking about this is he says, you know, why couldn't you do this with one-time pads? You have a one-time pad. He has a different one-time pad. So you encrypt your message using your one-time pad, send it to him. He encrypts that encrypted message with his one-time pad and sends it back to you.

Leo: So only he could - go ahead.

Steve: Well, okay. So now you decrypt the one-time pad that he encrypted with your one-time pad. Now the message is only encrypted with his.

Leo: It's half-encrypted, right, yeah.

Steve: Then you send it back to him. He decrypts it with his one-time pad, and now it's back into plain text. So essentially, every transport of the message was encrypted, either by yours, by both of yours, or by only his. And neither of you had to share your one-time pads with each other. And what's wrong with that? Isn't that, like, a really cool solution for sending an encrypted message?

Leo: This eliminates the problem of exchanging one-time pads, which is the weak link in a one-time pad system.

Steve: Exactly. So...

Leo: If you draw a picture of this, it will become clearer.

Steve: Well, yeah. Or if you think of it using that lockbox analogy, you know, the idea. And what's cool about this, the encryption that we've talked about - and I'm going to run back real briefly a little summary of last week. You know, we talked about Exclusive OR where you take a byte of data, and you Exclusive OR it with a random byte. Another way to think about that that I didn't specifically say, but it sort of makes it clear, is that you have sort of a - think of having a stream of bits of your message. What the Exclusive OR does is it conditionally or optionally or maybe inverts a bit of your message. So basically you are - when you XOR your message with a stream of random bits, basically you are randomly inverting your message bits. And what's cool is that you can do the same thing again. If you re-XOR with the same random bits, then you're going to flip the ones that were flipped before back to where they were, restoring the original message.

Leo: Right.

Steve: Now, that's why this idea of being able to double-encode a message works, because your pad [glitch] bunch of random bits. Then you send it to the other guy. He uses his different pad to flip a different set of bits. He sends it back to you. You use your pad to again flip your specific set of bits. And then, when it goes back to him, and he does it a second time, it flips everything back the way they actually all really were. So you could sort of see how it might work.

Now, if someone's confused about this XOR thing, we could fall back to this notion of a ring. Remember we initiated this whole dialogue talking about the secret decoder ring, where from a given letter you go some number of letters or symbols forward to encode, and then you go that same number of symbols backwards to decode. So in the same way that an XOR can be used reversibly, you could also just use addition and subtraction on the ring, where you take the random number and add it to the symbol to get a new one. And then, for example, if he did the same thing, then you've sort of both added your own random numbers. Then when it comes back to you, you subtract your random number, the same random number that you had added before. And then when you send this back to him, he subtracts his same random number, which will bring you back to the original symbol. So anyway, that's what we're going to leave people with for one week.

Leo: So you're saying it doesn't work. There's a flaw.

Steve: No, I'm not saying - I'm saying maybe this is really cool.

Leo: Oh, okay.

Steve: So maybe it works, maybe not. People should think about it...

Leo: Well, what do you think? All right.

Steve: ...and decide what they think, and we'll have the answer at the beginning of next week's show.

Leo: You are sneaky.

Steve: Okay. Q&A time.

Leo: Well, I've got the questions. Do you have the answers?

Steve: I hope so.

Leo: Let's go. First, two related questions touching on the same very important conceptual issue. You mentioned that 128-bit website encryption is safe to use and near impossible to crack. But if that's the case, why is it copy-protection is so easily broken? What's the difference between that and website encryption? Is TruCrypt or any other crypto program open to easy cracking? And before you answer that, let me just mention that, in your introduction to the cryptography episode a couple of weeks ago, you said that DRM was doomed. And I agreed with you, it could always be cracked. But then, you know, you keep saying "uncrackable Voice Over Internet" or "uncrackable email." Well, huh?

Steve: Yeah, yeah. That's a really good point. And the reason I wanted to present these questions is that there's a fundamental concept here which is crucial for people to get. And that is that the reason DRM can be cracked is that the device which knows how to decrypt the encrypted data is right there, and it's doing it in front of you in order for you to get the content which is being displayed or listened to or whatever. And similarly...

Leo: So it has the private key somewhere...

Steve: Well, it has...

Leo: ...stored away.

Steve: It has whatever. I mean, whatever it is, it knows how to bring it back into usable form. And copy-protection is the same way. If something is copy-protected, it's got to be unprotected for the computer to run it. So when it's running in your machine, it's in a run-able state. That is, whatever it is that's been done has been undone to it in order for it to be usable. And that's the key. So what's different with website encryption is - and we'll get into this in the coming weeks here when we explain how symmetric key and asymmetric key and public key and all this stuff all interrelate. But the idea is that you're encrypting information which is going somewhere else. And nowhere in between, in the middle, no eavesdropper has sufficient information to do the decryption. But when it's coming to a device which is going to run copy-protected software or display DRM data, that device has to know how to display it. And so the point is, anything can be reverse-engineered. So it's, you know, anything, I mean, this is how DVD encryption just got cracked immediately was a very clever engineer figured out what the DVD players were doing in order to do the decryption. He said, oh, here's how it's done.

Leo: A very clever 16-year-old engineer.

Steve: Yeah.

Leo: So they didn't have to be that clever. Because they're doing, and so he wrote DCSS, which reverses that process. It allows you to kind of do the same thing as a DVD player would do.

Steve: Right. But, for example, if you're just exchanging a - like you have an opaque token that someone has given you. It's like, here, hold onto this and give it back to me, sort of like a web browser cookie. You have nothing to go on to decrypt that. You have no traction. You know, you don't have in your computer the code which is going to turn that into something that makes sense. Instead you're just giving it back, if it's been given to you...

Leo: Right, makes sense.

Steve: ...and you just have no traction.

Leo: Yeah. Eric Bolt writes from Hotmail, a Hotmail account: "I have a wireless connection enabled, but it isn't connected to a wireless access point or another computer. For example, I'm at work with my laptop." He's hardwired into the LAN, you know, RJ45 plugged in there. His wireless connection is still enabled, but he's not using it. Is there a security threat there? Could somebody get on his computer through that connection?

Steve: Yes. Yes yes yes. And in fact there is a known - there are several known hacks. Wireless can work in two different modes. In infrastructure mode, where you have the traditional type of access we've been talking about where the infrastructure is a wireless access point or base station, the alternative way to operate is called "ad hoc," where, for example, two laptops are able just to talk to each other without using a - instead of talking through the access point, they are able to connect directly. Windows XP's default mode is both, to allow both infrastructure and ad hoc. And there have been reports that are confirmed of people on airplanes whose laptops are being used getting hacked on the fly, if you'll pardon the pun.

Leo: Even though they're not on the Internet, just because their AirPort is still on.

Steve: Yes. Because they're...

Leo: AirPort, that's a very Apple-specific thing. Their Wi-Fi is still on.

Steve: Exactly. So it is the case that a computer, even as it was in Eric's case here hooked into a LAN with an RJ45 hardwire, if his radio is on and the computer is configured to allow ad hoc operation - and that's something you can change in Windows. But it's really better, and this is what I always do, is just to turn that off when you're not really using it. I mean, it saves battery power, for one thing. And it's just sort of a good, secure practice to shut down any radios that you don't actually need to be using at any time because it's an exposure.

Leo: If you had WPA turned on on your client-side, would that...

Steve: You are going to be safe.

Leo: You are safe then, okay.

Steve: Yes.

Leo: So if you don't want to turn it off, if you're using - you should just be using WPA. Well, wait a minute, though, because you can't have it turned on all the time because otherwise you wouldn't [glitch] on to, oh, I don't know. It's too confusing for me. I mean, in other words, I don't turn it on, you can't turn it on for all connections because otherwise you wouldn't be able to get on when you got to Starbucks. So it's only when you're connecting with a wireless access point...

Steve: That's a good point. And it may be that, for example, someone has had to turn their security off when they were doing something else, when they were, like, away from their home, in order to use an open connection.

Leo: Security isn't really blanket for any connection into your card. It's just saying, when you connect with this access point, use security. But your card is kind of promiscuous, if you want to use that term.

Jason in Orlando writes: "I just ran into something I've never noticed before. I'm a college student in Orlando, and I'm on a school PC, still using Internet Explorer. They're slowly making Firefox available, but not here in mine. I logged into Gmail, checked my mail, then went to another site. After browsing for a few minutes, I headed back to Gmail" - you know, mail.google.com - "to see if I had a response. And it went right there. I didn't have to pass muster. I didn't have to give it a log-in or a password or anything. It just opened my inbox. I closed the window, opened a new one, headed back to Gmail, and then I had to log in again." He said, "I simulated the thing a few more times and realized that, if I didn't close the window, I could always just go right back in without a password. I can't remember how times I've left a terminal at school and just clicked the home page button for the next user." And he said, "I'm going to try this at home." Is this something to be aware of?

Steve: Well, actually, yeah, I think that's a very useful bit of caution for anyone using web browser-based email. What he's seeing is that he's got this Google mail account configured to, by default, just sort of stay open. He wasn't explicitly clicking Logout. And on any of those pages, when you're in Google mail and many other web-based mail systems, you can explicitly log out. That invalidates your login for that session.

Leo: It's using what's called a "session cookie." So it checks the session cookie, says, oh, he's still here.

Steve: Well, exactly. And in fact we will be talking about cookies also in the future because there are session cookies and much more persistent cookies. And you can tell Google mail that you want to stay logged in on the current machine unless you log out. But even if you don't have that option set, unless you explicitly log out, and you keep that browser session, which is what was happening in this case, you're still logged in.

Leo: Right.

Steve: So it's a useful cautionary note that people should explicitly log out whenever they're using a public terminal.

Leo: Many sites, I think Google mail included, have a little checkmark under the box saying "public terminal log-in or not."

Steve: Oh.

Leo: I'm not sure if what that's saying is don't save any session cookies, or if it's saying don't save any long-term cookies.

Steve: Yeah, Google mail doesn't have that, that I've seen.

Leo: Doesn't do that? Oh, okay.

Steve: You know - yeah.

Leo: I've seen it before on many sites where it's saying, is this a private machine? In which case, just for convenience we'll maintain your log-in.

Steve: It sounds like, yeah, it does sound like they may have taken extra measures. Like, for example, maybe it times out after a much shorter period of time.

Leo: Yeah, yeah. I have that on my FastMail account. Let me just - I thought for sure that Google would do that. I mean, it would seem to be sensible. Anyway, that's a good thing. So your recommendation is close the browser; or, better yet, just sign out.

Steve: Explicitly click the...

Leo: Sign out.

Steve: ...yeah, the log-off or log-out link before you close the browser window because you just, you know, you want to make sure your logon credentials have [glitch] with that remote server.

Leo: Very, very important. All right. And I'm just looking through the Google settings, and I don't see anywhere that you can - there must be somewhere where you can say, look, don't save anything. But maybe not. It is a convenience.

Alan Aykings [ph] asks: "Are there any issues with using PPTP" - that's Point-to-Point Tunneling Protocol, which Windows uses as its VPN protocol. "I currently use DDWRT with a PPTP server enabled, and I was wondering if there are any security issues with it. Is it also okay to open the PPTP port, which is 1753 to a Windows computer, and set up the Windows computer to accept incoming connections for a VPN?"

Steve: Well, he asks are there any security issues. That's a huge question, of course.

Leo: There's no right answer, either.

Steve: Okay. Opening a port to Windows means that it will accept incoming traffic, incoming connections. And we've seen, I mean, time after time after time, buffer overruns in Windows machines. And so anytime you open a port to a Windows machine, you are hoping and praying that there isn't some known or unknown exploit that would allow that buffer overrun, if there was one, to be taken advantage of. I mean, as we know, you put an unprotected Windows machine on the Internet, and it's just hosed in a matter of minutes. It's the really good thing that Service Pack 2 for XP did for us is that they've got their own little firewall running all the time. So, for example, for him to open this port he'd have to explicitly allow traffic through the firewall. It's that firewall which is responsible for, you know, making Service Pack 2 of Windows XP a much better machine.

Now, the one architectural security problem is that Point-to-Point Tunneling Protocol does not have strong authentication. That is, it's very possible for, if somebody was determined and wanted to, for someone else to be able to log on to his machine remotely, which is certainly not what he wants. Microsoft has tried several times, they've strengthened their algorithms, but they just don't want to take the step to provide strong authentication because it's not as simple as not having strong authentication. It's one of the things, for example, that makes OpenVPN a little trickier to configure because OpenVPN, you know, our chosen solution has industrial-strength, military-strength authentication. Nobody who doesn't have your credentials will be able to log onto your system. That's not true of point tunneling protocol.

Leo: So your recommendation would be not to open that port and, what, not accept incoming connections?

Steve: I would call that a security issue.

Leo: Yeah.

Steve: You know, he was saying are there any security issues.

Leo: Yes. You created one.

Steve: Yeah. I mean, if he really wants to use Point-to-Point Tunneling Protocol, see, he's got a router. And I guess it's maybe a little safer if he uses his router as his endpoint, that is, connects using PPTP to his router. That would get him into his network, as opposed to trusting the Point-to-Point Tunneling Protocol server on Windows, which just, you know, always makes me think twice.

Leo: Right. Are there any particular flaws you know of right now, or are we in between...

Steve: No.

Leo: We're in between flaws.

Steve: Yup. And of course no one knows about a flaw until we all know about it.

Leo: Right.

Steve: So who knows.

Leo: And you can almost, I mean, I think with any software you can pretty much say - I don't think you can ever say that a software is guaranteed flawless. You write software. You know. It's just - it's too complex.

Steve: It's very, very hard.

Leo: Yeah, yeah. Unless we're writing in Ada or something, which nobody does. Does anybody write in Ada anymore?

Steve: I think it's gone.

Leo: That was something, a language the Department of Defense proposed, based on Pascal, that was intended to be a secure language, you know, much harder to create bugs in.

Steve: And support big projects and team building and all these other things, yeah. It was just a big goof.

Leo: It was fashionable. Nobody uses it. Of course, I'm talking to the guy who programs in Assembly language. Which probably is a good, secure way to do it. At least you know what's going on.

Alan in Des Moines writes: "If the limiting factor for crypto security is computing power, how about distributed processing? Things like, you know, SETI@home or the Folding@home. Couldn't some Trojan horse virus infect millions of PCs and have them all act as processing units for a crypto...." Wait a minute, this is too good an idea. I'm not sure I want to read this question. "...acting as processing units for a crypto cracker and get around the processing power requirement?" You could create a pretty massive supercomputer, couldn't you.

Steve: Yeah, well, in fact, that's what the SETI@home project does, of course, is they're borrowing everyone's screensaver time while their machines are on, doing all this number crunching, signal processing of signals received from outer space. You know, basically a massive, distributed computer whose overall computing power is extremely high.

Leo: Right.

Steve: A number of people suggested this idea, that is, you know, because one of the things we talked about last week was that it's only a function of computing power. If you had long enough or fast enough, brute force attacks can function. The thing that renders that infeasible is the scale of difficulty as the number of bits that are used in the key increase. I've got a table here, and I'll just give you a sense for how quickly these numbers get big. And the point is that, for example, we glibly talk about 128-bit key. But it's not, for example, twice as hard as a 64-bit key. It's the squared as hard. It's the total number of combinations squared more difficult. So, for example, if a given computer at a given speed took two seconds to crack a 40-bit key, that same computer would take 35 hours to crack a 56-bit key...

Leo: Wow.

Steve: ...one year to crack a 64-bit key, 70,000 years to crack [glitch] key, 10 to the 14th years to crack a 112-bit key, and 10 to the 19 years to crack a 128-bit key.

Leo: So you'd have to get a lot of computers in your little cluster to solve that one.

Steve: There aren't that many computers. I mean, so my point is, that's the rate at which the computing burden for brute-force cryptographic cracking, that's the rate at which it scales as key length goes up. And, for example, just for the hell of it, I'm using 256-bit keys...

Leo: Yeah, me, too.

Steve: ...in my OpenVPN system.

Leo: Right.

Steve: Because why not?

Leo: Why not?

Steve: I mean, now we have computers that can handle that length with no trouble at all. And, I mean, lord knows how many years, I mean, it's to the point where, you know, your home will be attacked and ripped apart by someone...

Leo: By the explosion of the sun.

Steve: Exactly.

Leo: You can stop worrying about your encryption, your message being stolen. Worry about the sun exploding because that's only a few billion years off.

Steve: Yes.

Leo: I just created a new key for my PGP signing. And I use a 4,096-bit key.

Steve: Well, now, that'll be the public key, which is different than the bulk crypto.

Leo: Right.

Steve: But that's way big, Leo.

Leo: And it used to be that you would choose a smaller one because of the computational penalty. But you don't have that anymore.

Steve: Right.

Leo: These things are - computers are so fast. So is the default key still 128-bit, or...

Steve: 128-bit is really safe. And that's what SSL is using for its crypto. And, I mean, it's, you know, 10 to the 14 years. We're safe.

Leo: Right, right.

Steve: No matter how many, you know, SETI computers we commandeer for crypto. And notice that all that work would only crack one key. If you did all that, you would crack one key. As we're going to be learning in coming weeks, the new technologies, which use public key technology, sort of piggyback per session symmetric keys like we've been talking about. So they're changing all the time. And in fact, I've got OpenVPN set up, and this is what I'll be explaining to people, how it's changing the key automatically periodically, even during a connection.

Leo: Wow.

Steve: So it's just - give up. I mean, you just no longer have to worry about the security of this stuff.

Leo: And if you update your key every year, you're probably safe.

Andy from Chicago, Illinois writes - boy, we've got quite a few here. We're going to start moving a little faster. "My ISP, Cyberonic, recently switched backbones and issued new IP addresses to subscribers. What I've learned since is that the range of IPs I'm now in was recently in the 'bogon' address space," which is, he says, a new vocabulary word for him. "This is causing big issues with anything going upstream - webmail, FTP, WebDAV, and some sites have trouble loading images. I've contacted the admins of some of the sites and come to the conclusion the packets are being blocked somewhere en route. I'm not certain where to even start looking for solutions. Any ideas?"

Steve: Well, this is really an interesting question. I thought it was really fun. First of all, a bogon - he was talking about this being a new vocabulary word - a bogon is the measurement of bogosity, or bogusness. So it's like it's the quantum. You can have so many bogons' worth of bogosity, you know, technically. I have said before that we're not running out of IPs on the Internet, and that there's even a lot of IP space still not being used. Remember that we've talked about how, for example, anything beginning with a 10, you know, 10.anything is - we know is reserved for private networks.

Leo: So that's a bogon.

Steve: So, yes, that is one of the bogons. What happens is, Internet routers are configured to drop any traffic which is destined for the so-called "bogon space," which is a term that the network engineers use. Get this: 40 percent of our current Internet IP space is bogon.

Leo: Wow.

Steve: 40 percent is non-routable, unused space.

Leo: So that's the 10., the 192.168, 5.

Steve: In fact, even looking at just - yes, exactly. Even looking at just the first byte, anything beginning with a 0, a 1, a 2, a 5, a 7, 23, 27, 31, 36, 37, 39, 42, 48, 50, 77, 78, 79, 89 through 123, 127 - which of course is the local host range - and then 173 through 187 and 197, all of those are just - they don't exist. They're bogons.

Leo: How did Andy get a bogon IP address?

Steve: Well, what happened was someone somehow convinced ICANN to give up one of these big networks. The problem is, all over the Internet there are routers configured with this bogon space, and firewalls and all kinds of equipment that just have these, like, hard blocked.

Leo: Yeah, don't let anything in from that address because it's not a real address, so...

Steve: Exactly. So essentially the problem is that a recently allocated IP region is no longer bogon space, but it's taking the rest of the Internet some length of time to get the message, to get the news. Meanwhile, equipment will just be dropping his traffic, and there's nothing he can do about it.

Leo: Wow.

Steve: I mean, it's annoying.

Leo: Wow. So does this happen frequently, that bogon IP addresses are reassigned to the real world?

Steve: No, not frequently at all. I mean...

Leo: I can see why.

Steve: The people who keep these things allocated and assigned, I mean, these IPs are just precious treasure on the Internet. And of course, as we know, the emergence of NAT technology that allows corporations to run many computers on few public IPs, and even individuals, you know, I've got one public IP on my cable modem, and I can run a whole network, as do all of our listeners. So, you know, of many computers. So the pressure has been hugely reduced thanks to NAT routers. And so, you know, there's, like, 40 percent of the 'Net is still unallocated.

Leo: So you can appeal to ICANN and the RIRs and say, look, you know, we're using this, and please...

Steve: You've got to make a, yes, you have to make a really good case. And if you make a really good case...

Leo: But you still have all this legacy hardware that's not going to be updated, and...

Steve: You're going to have some pain for a while, while network administrators all over the 'Net go, oh, look, you know, this is no longer a bogon address in the IP space.

Leo: We have our first question from the Amish country. We don't get a lot of questions from the Amish country, I'm not sure why. Just a joke.

Steve: Do they have computers? I know.

Leo: Dave Solan [ph] in Amish country, Mannheim, PA writes: "I have a question for you regarding hotel LANs. I was recently at a hotel where you could hook up to their cable modems for free access. The cable coax line from its TV was split so one line went to the TV, one to the cable modem. They actually had a cable modem in each room."

Steve: Isn't that neat?

Leo: I'm guessing this hotel didn't have their own LAN, and this might be a little safer than plugging into a hotel where you simply plug into their network. Is that true?

Steve: Yes, it is true. I had never heard of a hotel giving every room its own cable modem.

Leo: Now, I should say that I have heard of hotels doing this, but they aren't actually doing that. They have their own private network which provides both TV and data, and they're putting decoders in the room.

Steve: Ah, okay.

Leo: I think it's more likely what was happening.

Steve: Even so, if they're using traditional DOCSIS-style cable modems, that is, you know, the universal standard for cable modem technology, that is an encrypted signal over the coax.

Leo: So that modem would be kind of a barrier to anybody trying to look at you.

Steve: Exactly. So the things we've talked about, the dangers we've talked about before of people easily, for example, sniffing your traffic and playing games, I can't say for sure that they would be defeated because you'd have to really look at the network to see, for example, how ARP traffic was being handled within that network. It might be that these cable modem-like things are just really bridges, which are bridging the LAN through the coax segment. That's a possibility. But it is potentially more secure than just a regular, you know, 10BaseT-style LAN that we see in most hotels.

Leo: That's an interesting...

Steve: Yeah, I thought it was cool.

Leo: ...point. In fact, staying on that subject, another listener has asked: "If I want a secure connection in a hotel, can't I just carry my NAT router with me?"

Steve: I like that one too, a lot. Because it was like, well, we've talked so much about how secure NAT routers are, that is, what security they provide. So someone thinking about, okay, hotels are a problem, how about if I just use my own NAT router? The problem is, that would secure you sort of within your own room, where you're already secure. But your traffic then would still be communal on the public side of the NAT router. So you would - unfortunately it provides no security from, like, interroom eavesdropping sorts of things, which is the problem in a hotel setting. One thing it does give you, though, of course, is a very good firewall, in the same way that a NAT router always blocks unsolicited incoming traffic. So it would potentially firewall your computer if you didn't already have a software firewall. It would be useful for that. But it wouldn't completely protect you from, you know, the types of interroom LAN problems that we've talked about in recent weeks.

Leo: In fact, I do that because I carry my AirPort Express when I'm in a hardwired hotel to give myself kind of my in-room AirPort. And I use...

Steve: So you can sit on the bed with your laptop and so forth. Yeah.

Leo: Yeah, exactly. And I use WPA. But it's important to remember that, once it gets on the other side of that router, it's just unencrypted plain old traffic, just like anywhere else.

Steve: Right, right.

Leo: And that's why I use HotSpotVPN, thanks to you.

Steve: Exactly.

Leo: Pat Deery [ph] and many other sharp-eared listeners have mentioned, during Episode 29 that was our Ethernet Insecurity episode you said - you said, Steve...

Steve: I did.

Leo: ...I'm pointing at you, that "...ARP cache poisoning occurs when a malicious computer sends out a false ARP response to two other computers, making the malicious computer the man in the middle. My question is, do the two computers just accept the responses blindly, even if they didn't send out a request? Couldn't this be avoided if the two computers in question only accept responses for requests they send?"

Steve: That was a neat idea, and many listeners had it. That is, the idea being that the way ARP is implemented today it's trivial for a malicious machine to send unsolicited responses, ARP responses, to computers to cause them to change the MAC address with the IP that it was previously associated with in order for them to send their traffic through it. And so people said, hey, wait a minute, what if computers just ignored responses that they didn't send? Well, that kind of could work, but then you get a problem of the bad guy responding first, before the good guy, to a valid ARP request with their response. So, and then you would ignore the second one. But then you've got the problem of not being able to detect IP collisions on a LAN. You want to be able to detect when two machines have the same IP. I'm sure, Leo, with all the networking you've done over time, you have seen the little Windows dialogue box that comes up and says, wait a minute, there's another machine with the same IP somewhere.

Leo: Right.

Steve: So the problem is really not tinkering with the protocol, but the problem is a lack of authentication. And the right way to solve it, we'll be talking about it downstream. There are some forthcoming specs, I think it's 802.1x begins to do this, where we get authentication of endpoints on a LAN which will begin to solve this problem. And that's, like, the right way of doing it, instead of sort of trying to play with the protocol. Then you get all kinds of weird side effects, and things break in bad ways.

Leo: Okay. But it would kind of be like a router rejecting stuff that it didn't initiate; right?

Steve: It's very much the same sort of concept, yeah.

Leo: Yeah. But there are issues with it.

Steve: The problem in this case is, because there's no authentication, you don't know that the first answer you get is not the bad guy.

Leo: Oh, that's right, of course. So it could just interrupt, say excuse me, I'll take this from here. I got it. I got it.

Steve: Yup.

Leo: Let's see. Kevin Hooper writes from his Hotmail account: "I had some suspicious activity yesterday when I loaded drivers for a USB video capture device, and then Microsoft AntiSpyware flashed a LSP, Layered Service Provider warning." That sounds pretty dire. "Subsequent behavior of ZoneAlarm only increased my anxiety. It disabled it spontaneously. What's that? What's an LSP, particularly when you're talking about a USB video capture device, and what should a user look for when using a layer inspector like the add-on from Ad-Aware?" I don't even know what he's talking about, but you're the spyware king, so I'll let you...

Steve: Well, a Layered Service Provider is a feature that's unique to Microsoft Windows' implementation of a layer that runs up in the user's space. There's, you know, there's user space and kernel space. And so there are kernel drivers that run down there that interface with the hardware. Up in where the programs themselves run, in so-called "user space," there are DLLs that get loaded in order for user applications to talk to the network. Microsoft, in their infinite wisdom...

Leo: Infinite wisdom, yes.

Steve: ...created something called a Layered Service Provider as a feature of Windows. What it allows you to do, essentially, is it allows any software - and of course there's no protection for this, no authentication, no notification, nothing - it allows any software to insert itself as a shim in the networking traffic, sort of up in the user space. It's sort of a poor man's network interface ability. The bad news is, spyware is using this in order to intercept browser and email traffic and to insert its own and basically play all kinds of games. So people like the Ad-Aware guys have - they're, like, aware of this being another vector of attack for spyware. And they're now keeping an eye on sort of the chain of processes in the so-called Layered Service Provider stack because you're able to have multiple service providers that, like, sort of link to each other, and bad stuff inserts itself in there. So essentially it's sort of one more...

Leo: One more vector.

Steve: ...one more vector of vulnerability that Windows has. And so apparently, in this case, maybe the software was buggy that this guy installed, or he just has a lot of software like Ad-Aware and ZoneAlarm and Microsoft AntiSpyware that are all now watching the Layered Service Provider stack for any changes and alerting him if something changes. He apparently installed something benign, so he should say, oh, okay, this thing is a Layered Service Provider shim-installing piece of software, now I know. So that's what it means.

Leo: Wow. You know, there's a pattern here in the way Microsoft works. I mean, essentially they're enabling, without really thinking about it, just putting things in that let people do anything they want.

Steve: Yes. I mean, it's just the way they operate, the way they implement. Now, in fairness, this is old stuff. It's still around, like so much other of Microsoft's old stuff, for the sake of forward compatibility. So this stuff has been around since before security was really an issue, which we could argue for Microsoft has only been in the last couple of years. So, you know, I don't know that they would do it this way now. One would hope they wouldn't. But we are stuck with it. And the malware has found it, and it's just a simple way for malware to get itself running, basically in every process that the user uses that communicates with the Internet. This LSP stuff is potentially really bad.

Leo: Wow. Anthony from Albuquerque asks: "I recently...." Did you have to look up Albuquerque when you typed that, by the way?

Steve: I think I probably did. I just put it into Google, and Google says, "Did you mean this?" Oh, yeah.

Leo: You mean Albuquerque? That is probably the single hardest city to spell. "I recently relocated, and at my new place I have a DSL connection with an ISP that is issuing private IP addresses." We're going to have to start limiting people to three or four acronyms per message here. "I have a DSL connection with an ISP that is issuing private IP addresses." This is that 192.168.1 to me. "I had to change my router configuration to issue 192.168.0.something in order to get online. Everything's working okay except BitTorrent. I'm thinking the ISP is either actively throttling BitTorrent traffic" - which is very common now - "or preventing a good connection by not having the BitTorrent ports forwarded the way I have to on my own router. I'm using Azureus, but my ISP is not on their list of so-called 'bad ISPs.' I changed my BitTorrent port, as also recommended by Azureus. That hasn't helped. In my troubleshooting I also discovered my dynamic DNS is no longer working properly, as the IP it receives is an external IP in a router somewhere." Probably another 192.168. "Is there another way around this ISP's shenanigans to get DynDNS and BitTorrent working?"

Steve: Okay.

Leo: How does this even work?

Steve: This guy is hosed.

Leo: Because the ISP is using a router to connect these other - they're not giving you a real IP address.

Steve: That's exactly right, Leo. And I've seen this before, and it's beginning to happen. ISPs, maybe who can't get a large enough public IP space for all of their customers, they're beginning to run NAT themselves and then issue private space 192.168.x.y IPs to their customers. Now, this guy was able to get his router to work because he was smart enough to know that you have to have different networks on either side of the NAT router so that it's able to know which traffic to route across. So, for example, his ISP in this case was issuing 192.168.1.something, and that was the default subnet inside of his router. So he had to change that. Then he was able to work. But he still has the problem that his traffic from his own use is not going public until after it crosses his ISP's router.

Leo: I can see a lot of issues coming out of something like that.

Steve: Well, you can't run servers.

Leo: Right.

Steve: You see, his BitTorrent client is trying to be a server.

Leo: Is a server, right.

Steve: Exactly. And there's no way to run a server without explicit port forwarding. And the ISP can't forward ports to everyone. They would only be able to go to one particular user.

Leo: He wouldn't be able to use Videochat either; right? I mean, there's a lot of things that would break.

Steve: Nope. As I said, this guy is seriously hosed. I mean, this is...

Leo: Well, I mean, this is why ISPs don't do this very - I mean, you'd think this would be in their advantage to do it, but it would cause so many problems.

Steve: It's convenient for the ISP. But basically you end up being a client only to the Internet, and all kinds of stuff will no longer work.

Leo: So complain to the ISP. And if they don't fix it, you're going to have to go somewhere else.

Steve: I would go somewhere else. If there's any way he can change ISPs, that's the first - I'd just, you know, don't even consider complaining. Just get out of there.

Leo: I could see a small, rural ISP doing this because, you know, they're small, they don't have a big enough pool. Essentially, you know, they become kind of a, you know, in-house, like, renter of access. But you can't do, I mean, a lot of things break. A lot of things.

Steve: Yeah.

Leo: Yeah, all right. Chris Peterson - our last question. Chris Peterson of Billings, Montana asks: "Can anyone's machine just be hacked? During an argument online the other day a belligerent kid said he was just going to hack into my machine, and he was going to get what he wanted from it, and so there. I assumed he was bluffing. He didn't seem that smart. But could a really talented hacker crack into anyone's machines?" In other words, is any machine vulnerable?

Steve: The good news is, no.

Leo: No.

Steve: It's not like in the movies, where, you know, you just decide, okay, we're going to go hack into this computer system somewhere and get whatever we want to from it. I mean, the bad news is, many systems are so complex that there are ways in. But, you know, in some cases that requires social engineering, where you get the secretary, if the boss is out of town, to flip his keyboard upside down and read you his password on the bottom of his keyboard or something like that. I mean, there can be ways in. But fundamentally, it's not as if all computers have sort of, like, have a soft boundary, and if you just poke at them hard enough you'll be able to, you know, stick your spike all the way through their crust. I mean, it just doesn't work that way.

Leo: Good.

Steve: So that's the good news.

Leo: On the other hand, there are so many holes and exploits in operating systems these days that there's no guarantee that you're safe, either.

Steve: You've been watching, of course, OS X has been having its share of trouble, and Safari and so forth.

Leo: Although, you know, it's funny, they're talking about OS X having problems because it has a lot of old UNIX programs, and Apple hasn't been assiduous about making sure the patches are applied. But I haven't seen any widespread exploits. I mean, it seems more theoretical.

Steve: Yeah.

Leo: But it's something to keep in mind always, that, you know, unless you are actively hardening a system, there's always the chance that you're hackable. And if you're a good hacker, you probably have in your toolkit a few of these well-known exploits that are out there.

Steve: Well, and I would say, if nothing else...

Leo: Or not so well known.

Steve: Yeah. I would say, if nothing else, the notion of needing to keep your software patched and current has now arrived for Mac users the way it arrived years before for Microsoft Windows users.

Leo: Although I think most Mac users, just kind of by default, turned on that there's a software update feature. They did it before Microsoft did it, and it's automatic. And I think most Mac users have it turned on. I think there's a bigger issue - and I was guilty of this. You know, we kind of tend to run as administrator, which, of course, if you've ever used a UNIX system, you know, is verboten. You never run as root. But both Windows and Mac users tend to run as administrator. On Windows you almost have to because there's so many programs that won't run otherwise. On the Mac, you can in fact very conveniently, very easily run as a limited user.

Steve: Well, and in fact from the UNIX world the idea of being "root," as it's called, is so powerful that from day one programs have all assumed they would not be running as root. They would create their own, like a GNUS account or an Apache account or whatever that would run with limited resources. So the heritage of UNIX has that fundamental security awareness advantage...

Leo: Right.

Steve: ...where, you know, things really do work just fine not being root, and you only briefly have to do that from time to time for, like, special, you know, true maintenance things. Whereas, as you say, Leo, under Windows all kinds of stuff is just, you know, broken.

Leo: But heritage is the opposite, assuming that you have all power.

Steve: Exactly.

Leo: And that's exactly the difference. It's just, you know, how people are used to working, or programs are used to working.

Steve, we are through the questions. Well done.

Steve: I think we will resume then next week. We'll come back with our answer to this week's Puzzler/BrainTeaser, why can't you send messages back and forth using that double lock approach with a one-time pad, or can you send them back and forth, does that work, in order to give you a really cool encryption system.

Leo: It seems to me that, if that worked, that it would be more widely used. So I'm thinking there's got to be something wrong with that.

Steve: And so the question is what, you know? People can think about it for a week, and we'll give them the answer at the beginning of the show next week. And I'll talk to you then, Leo. And...

Leo: Don't forget - go ahead.

Steve: No, I was going to say that we're going to talk about the second family of symmetric ciphers. We ran out of time last time, and all we were talking about so far is what's called "stream ciphers," that is, the idea being that you have a message, and, for example, you use a pseudorandom number generator or even a one-time pad to generate a stream of randomness that you mix in with your message to create a stream that cannot be decrypted, and then you reverse the process to get it back. That's one family of encryption. The alternative is called a "symmetric block cipher." And that's really, I mean, there we're going to see, when I quickly run through a list of them, like 2fish and Blowfish and AES and Rijndael and those things, those are things people have heard of. And those are really today's workhorse encryption. So that's for next week.

Leo: That's for Episode 33. Meanwhile, reminding everybody that they can get 16KB versions of this and all our other Security Now! podcasts, plus transcripts thanks to Elaine, on Steve's site, GRC.com/securitynow.htm. This program and all of Steve's labors are brought to you by his fabulous program SpinRite, which is the ultimate disk maintenance and recovery utility. I just recommended this the other day to somebody.

Steve: Oh, cool.

Leo: It's a must-have if you work with hard drives. And you can find out more about that now at SpinRite.info, a brand new site just for you. SpinRite.info. We also want to thank our good friends at AOL for providing us with the bandwidth and for broadcasting this show on their podcast channel on AOL Radio. That's AOL.com/podcasting. And thank our donors who make this show possible. Without you we couldn't produce Security Now! and all the other fine TWiT.tv podcasts. If you're not yet a regular contributor, we encourage you to visit the site, TWiT.tv, and press one of those Donation buttons. A donation as little as $2 a month really makes a big difference in keeping us running. And frankly, now is a good time. Our expenses are high, and I'd sure appreciate a little help here. So that's TWiT.tv, and you'll see those PayPal Donate buttons right there on the front page. Steve Gibson, we'll see you next week.