Listener Feedback #32

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

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-126.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-126-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 126 for January 11, 2008: Listener Feedback #32. This show and the entire TWiT broadcast network is brought to you by donations from listeners like you. Thanks.

This is Security Now!. I am back. We are late. And actually it's my fault. I apologize. I got back from Egypt with a raging cold - which I didn't catch in Egypt, I caught on the plane back, of course.

Steve Gibson: You're sounding a little, I mean, like, you know, okay, but like a little more throaty, I guess.

Leo: A little Barry White going on there. That's probably all the drugs I'm on right now. I'm feeling better, but we couldn't do it at our normal time. Then you had an important business occasion yesterday.

Steve: Yup, yesterday was burned up for me, so.

Leo: Well, that's fine. We're a day late, but not a dollar short. We've got lots to talk about. Hello, Steve Gibson.

Steve: Leo, it's great to be back with you. And officially now, Happy New Year. This is the first time we've spoken since '08 began.

Leo: Yes. And it's been a very good year so far.

Steve: Your trip to Egypt was a good one?

Leo: Fascinating. Fascinating. I'm going to - I've posted all the pictures. I posted, you know, I took over 2,000. But I had the good sense to narrow them down a lot. So if you go to my blog, they're all there in the photo section. About 125. Maybe not even that many. 80. Something like that. And we had just - it's fascinating. You know, it's a very challenging country. And, you know, the ancient ruins are interesting. But we also, you know, we're part, I mean, it's the world's largest Islamic city, 16 million people. Population's growing 100,000 a month.

Steve: Whoa.

Leo: Got huge population crisis. And so it's, you know, it's very - it was very interesting. I think we learned a lot about the Islamic world, as well. I'm going to write a longer essay, I think, for the blog.

Steve: Is population growing through immigration or birth?

Leo: No. It's an authoritarian state. There is neither immigration nor emigration. It's birth. And it's mostly birth in the rural areas. But what happens is they move into the city. So Cairo is just a sprawling, teeming metropolis. Which, you know, one block from the hotel there are dirt roads. It is a very interesting place. I have a lot to say about it, but I'll...

Steve: And the kids also enjoyed their...

Leo: Yeah. I mean, they learned a lot. And they saw, I mean, we saw every, you know, important monument. And they're incredible. These ancient Egyptians 4,000 years ago, they were doing art that is modern and beautiful. We saw the world's oldest sculpture, 4 or 5,000-year-old wooden sculpture. And it's perfect, it's beautiful. And so it's really stunning what they accomplished all those years ago. And I think that for the Egyptian people that is a little bit frustrating because one of the questions they seem to be asking themselves is, what happened? You know, we were this dominant culture for so long, and now we're essentially a Third World nation. So it's tough times for them. But there were some very interesting people on the trip, as well. Met a lot of great people. So I had a blast.

Steve: Very cool.

Leo: Yeah. But I missed you guys, I really did, and I'm glad to be back at work, cold or not. This is a Q&A segment. We've got lots of listener questions.

Steve: Yes. We've got, well, and it's a bizarre coincidence that we had promised to get more focused on sort of current...

Leo: News, yeah.

Steve: ...events in security because there's just a whopper, a whopper in Windows.

Leo: You know, it's funny, I haven't seen word one about this. All the coverage on CES, which by the way there was nothing worth covering, and not one word about the security flaw.

Steve: This is - I've seen people saying that this is the worst Internet security flaw in history.

[Talking simultaneously]

Steve: ...all about it.

Leo: Is it Vista? Is it Vista, or is it...

Steve: Even XP.

Leo: XP and Vista?

Steve: And Vista. It's a remote code execution flaw. We'll get to that in a second, but I had some errata that I wanted to share. First of all, I wanted to thank our listeners for, one way or another, finding the vote buttons on my Kindle review.

Leo: I did.

Steve: Yes. Even though the link that I had posted took people to a buttonless, you know, voting button-free instance, enough people did that my review is now the featured review, the number one review.

Leo: Oh, good. As it should be.

Steve: Like by a large margin, like by two to one, I think. When I last looked there were 6,755 votes, and 98.4 percent of them said that they found the review helpful.

Leo: That's great.

Steve: I know that some of our listeners are among them because I've looked through the comments on my review. And some people have talked about you or me or the podcast or, hey, you know, there were no buttons on the link you said, but I found the buttons, blah blah blah. So but I think now it's in the first-place position it's got a life of its own. So I absolutely wanted to thank our listeners for making another one of my wishes come true.

Leo: Happy New Year.

Steve: And actually there are - it's very clear, too, that a lot of people are reading it who are not Security Now! listeners but really do honestly find it very helpful to them, so...

Leo: I brought my Kindle along, and that was how I read, and it was great. Even though I didn't have wireless in Egypt, obviously. And I showed it to a number of people who were thinking about the Kindle. And to a man they said, oh, okay, that's it, I'm going to get one. You know, it's not perfect, as you say. It's got warts. But it's pretty darn useful.

Steve: Yeah. Well, it's funny, too, because, on one of the other little errata notes I had, I wanted to mention that Kindle hacking is underway.

Leo: Oh, boy.

Steve: There are - people have not only physically taken it apart, but they've gone into the software. And there's, for example, pages of all kinds of hidden features. Apparently Google Maps is built in somewhere. And there's things where you can change justification options, which is one of the things I really wanted because I like ragged right more than flush left.

Leo: Yes, I'd love to turn that off, flush left.

Steve: Yeah, I just - I don't know. Just the programmer in me, I see like a long word on a line which prevents wrapping where it would be nice if it could. And this is like, oh, it just bugs me. So, but there's all kinds of undocumented things, so that's all beginning to surface. So Kindle hacking is happening. Also I wanted to mention Jungle Disk, which we talked about several weeks ago. I asked the author whether he'd had any effect from our mentioning it because I was curious about that. And he says, oh, yeah, he said, I mean, even over the holidays things were much busier. And a lot of Security Now! listeners had apparently posted in a blog that they learned about it from our podcast and were excited to check it out and try it out.

So, and the reason I actually wrote to him was I was curious if it was really effective in a situation where, for example, you want to keep a laptop backed up as you're coming and going, as connectivity is changing, you're visiting WiFi hotspots, you're just basically not even thinking about keeping your laptop backed up. And of course I wouldn't ever back up an entire laptop. But, for example, the My Documents folder, where most of your documents - and, for example, your desktop. And he said absolutely, it's designed with that I mind. So I have been using it in that fashion experimentally. And I've set up a couple friends who are not very tech savvy with their own Amazon accounts and am using it as sort of just a background constant, just out of your way, don't even think about it, it's just going to - it's going to keep your laptop backed up in case it should ever - the hard drive should get damaged or the laptop be stolen.

Leo: Yeah, yeah.

Steve: Well, anyway, Jungle Disk, I'm just - and it's been working that way really well for me now for a couple weeks. So I'm really pleased.

Leo: I think online backup is really the way to go.

Steve: I think that's the right thing to do, yes, for nave users. And I wanted to share, as I always do, a fun SpinRite story. This one is a little longer, but it was really well written. So it was from a guy named David Bins who sent email to GRC with the subject "SpinRite Success Story." And he says, "I'm an IT consultant who has just bought a copy of SpinRite. I've been," he says, "I've been listening to Security Now! podcasts since around Episode 40 and have meant to buy a copy of SpinRite as I got a lot from the podcast and felt guilty that they were free." He says...

Leo: [Indiscernible] guilty they're free. That's okay.

Steve: No, no. He says, "I was all prepared" - well, see, but we got him anyway. He says, "I was all prepared to buy a copy when I got to discussing Security Now! with a work colleague who also listens every week. He had purchased a copy and ran it on his machines regularly. He hadn't had any problems, and SpinRite hadn't ever found any problems with his drive. He suspected either his hardware was perfect or SpinRite wasn't all it was cracked up to be. So I held off buying a copy for myself."

Now, of course, we know that what's going on is the act of running SpinRite is maintaining his drives. Which is not to say that his drive wouldn't be fine without SpinRite. But by running SpinRite on a drive that is even perfect, it has the option or the opportunity of showing the drive that it's got problems it wasn't aware of because a drive only knows it has a problem when it tries to read a sector. It doesn't know it has a problem until it tries to read a sector. So that's how SpinRite functions good from a maintenance standpoint. And because this sector relocation is, literally, it's hidden, there isn't any way for me to say, oh, look, we relocated X number of sectors. I mean, that information isn't published by the drive. So you kind of have to take it on faith. On the other hand, this guy's using SpinRite, and his drives are not failing. So there you go. Anyway...

Leo: Wouldn't you rather have it be that way.

Steve: Yeah. David continues, saying, "I'm the sort of person" - and this is where it gets interesting - "who backs things up from one computer to another and also to a USB-attached hard drive at regular intervals, but even a couple of days of lost work can mean a lot to me. So I was particularly annoyed when my USB hard drive failed. It's a Hitachi drive, and all it does is play a tune at me when I plug it in. I've attached it to a power supply, but it doesn't spin up. I wasn't too concerned, as most of the data I needed was on my laptop. It was the first hard disk that I have had fail on me in probably the last 18 years." He says, parens, "(outside of work). So I felt I was due to experience some sort of data loss. My USB drive failed on 12/31/2007." Okay, so New Year's Eve. "And last night at around 11:00 p.m., January 7, my laptop drive failed, and my laptop would not boot, giving an unreadable drive error just after the BIOS screen."

Leo: Oh, I hate that.

Steve: Uh-huh. He says, "I recognized this as a failed drive and assumed I had lost a whole lot of data, including the accounts and invoices for my company. I did have a copy of the data on the failed USB drive, and also a copy from around three months ago on another laptop. But I had lost a lot of information and last night had a sleepless night, thinking about how I would

reconstruct what I had lost. I know I should have tried SpinRite immediately, but it just didn't occur to me at first. I purchased a copy this evening." Oh, so he said, "I purchased a copy this evening at about 9:45. After downloading the software I created a boot CD and ran it on my laptop. SpinRite was running through the recovery process by about 10:00 p.m." So 15 minutes later.

He says, "I was impressed at how quick it was for me to purchase the software and have it running. I can't tell you how happy I am after seeing SpinRite recover data from bad sectors. I'm writing this email on the laptop I was about to throw away, which is now completely up and running. It is 11:46 p.m., about two hours after buying your software, and everything is back to normal and running perfectly. If you ever need anything - advice about database administration or service management, which are my areas of expertise - give me a shout, as you've helped me enormously, and I feel I need to return the favor. Many thanks."

Well, of course he already has helped us by purchasing a copy of SpinRite, which is all I would ever ask. And I thank you, David, for the tremendous testimonial. I really appreciate it.

Leo: GRC.com, that's where you can get your copy of SpinRite. A must-have. Are you ready for some questions? Do we have anymore...

Steve: Uh-oh.

Leo: Wait a minute, we want to talk about this Microsoft problem.

Steve: Oh, we haven't gotten started yet.

Leo: Oh, man. Okay.

Steve: Okay. Now, okay. The takeaway message is absolutely everyone needs to run Windows Update. Last Tuesday was the second Tuesday of the year. Since the first Tuesday, of course, was January 1st, it was January 8 was the so-called "Patch Tuesday," when Microsoft releases their security updates. This week we had a doozy. It turns out that private parties informed Microsoft of a very low-level buffer overflow which permits remote code execution in the Windows TCP/IP stack itself. So, and this is...

Leo: Oh, is it the new stack that they wrote, or is this the old stack?

Steve: Well, this is XP's stack. And it's believed that there is a problem in Windows 2000 as well. So this has been around for a long time. What makes this specifically bad is this is down - it's in what's called the IGMP, the Internet Group Management Protocol. That's a multicast protocol. And but the point is, this is not like a service running with an open port where the service has the problem. This is the core of the stack itself.

Leo: So if you've got it running, which everybody does, you're vulnerable.

Steve: If Windows is running, exactly. And, Leo, none of the built-in firewalls block this.

Leo: Really.

Steve: Yes. Because this is, I mean, this is not something that, like, at the application level, where firewalls are blocking ports. This is more like, you know, like, well, it's very related to the ICMP, the Internet Control Management Protocol.

Leo: So how would this exploit be triggered?

Steve: Okay. One packet hits an unprotected machine, and it's taken over.

Leo: Oh, well, that's fast.

Steve: So, okay. What this means is, as far as we know...

Leo: It's a packet on what port?

Steve: It would be, well, it's not port based, and that's the problem.

Leo: So it could be anywhere.

Steve: It's more like - well, there are no ports for IGMP. It's a multicast protocol. So, see, ports are related to applications. So, like, port 80 is web services, port 22 is FTP, and so forth. So this is more like ICMP. You know, people are familiar, for example, with pinging and tracerouting, where that's - for example, the ICMP protocol is built into the stack as part of its plumbing. So that by definition of the RFCs, any stack which - any IP stack will have ICMP support, so that you're able to sort of, like, manage the connectivity of the stack on the Internet at a low level. Well...

Leo: But how does a computer know that that traffic is aimed at it?

Steve: Well, oh, because there is an IP address.

Leo: I see.

Steve: So it's an IP address, but not an IP and port. So, okay. So first of all, the good news is, NAT routers do block this. So here's another reason why it would have always been good to be behind a NAT router.

Leo: So a software router will not stop this, but a NAT router, a hardware router would.

Steve: Well, it's certainly possible that a software firewall could stop it. But Microsoft's built-in

firewalls do not. So, and this is still very new. Essentially what it means is there is virtual certainty that we are going to see a worm because this is a perfect opportunity for a worm to propagate across the Internet, finding unpatched XP and Vista machines that...

Leo: Now, wait a minute now. Vista has a whole new stack. Remember we talked about Vista's virgin stack.

Steve: Yes. Except it, you know, no doubt someone took a blob of source code...

Leo: The IGMP implementation's the same.

Steve: Exactly. They took a blob of source code...

Leo: So this affects XP, it affects Vista, affects Windows 2000.

Steve: Yes, yes.

Leo: Now, Microsoft patched it on Tuesday.

Steve: And that's my point, yes. So first of all, you know, the takeaway is make sure you run Windows Update, I mean, critically, on any computer that would be on the Internet not behind a NAT router. Now, when you're in like a T-Mobile hotspot at Starbucks or a FedEx Kinko's, you know, you're behind that location's NAT router, looking at - and you're in a 10. space most likely.

Leo: So you're safe there.

Steve: Yes. And again, and most people are going to have a NAT router at home because, you know, and we've been promoting the inherent firewall virtue of a NAT router. So a NAT router, even if it were vulnerable, and it's not because it probably doesn't support IGMP protocol, which is a multicast protocol, it's going to drop that packet dead right there. There's no way it can gobble through to a specific machine because a router would have no way to know where it should send the packet because it really shouldn't send it anywhere.

But the danger is any Windows 2000 with Service Pack 4, the latest service pack, XP or Vista machine which is placed on the Internet unpatched and not behind a NAT router, that is, with an actual public IP, what is virtually foreseeable is that the patch will be reverse engineered. We've already seen Microsoft's patches reverse engineered where the bad guys look at what changed and then figure out what the vulnerability was from the change that was made. So it's virtually certain that this patch will be reverse engineered. Bad guys are going to figure out where the buffer overflow is in IGMP. And maybe they wouldn't even have to reverse engineer. Knowing that there is one there, they could just find it the same way the good guys, the good hackers did, who informed Microsoft about this some time ago. So what'll happen then is a worm will be planted on...

Leo: Did you say some time ago? We're talking June of 2006.

Steve: I know.

Leo: I'm looking back at Microsoft's report. This fixes - this is something really a long time ago.

Steve: Yeah.

Leo: Wow. So, now, Microsoft says that it is only a denial-of-service attack on 2000. But on XP and Vista it's a...

Steve: Yes, exactly. So in 2000 it will basically crash your stack. There isn't a vector that they have found that allows a remote code to be injected as part of the packet. But as you said, under XP and Vista the incoming packet can carry executable code payload, which in classic buffer overrun mode will be run when this specially malformed single packet hits your computer. So if, I mean, it has to be, based on all the experience we have, that we're going to see a worm, that there will be enough unpatched Windows XP and Vista machines, even with their firewalls running, this thing cuts through their firewall. And they're going to get infected, and then they're going to turn around and start spraying random packets out at random IPs, as worms do, looking for other vulnerable machines. So you absolutely want to make sure your machine is patched, especially if you're someone who, you know, roams around with a laptop and might ever get a public IP address. It won't be long before the Internet will be sprayed with this IGMP protocol carrying one or more malicious types of code.

Leo: Microsoft gives credit to the IBM security people for discovering this.

Steve: Yeah, the X-Force people.

Leo: X-Force, yeah. Wow, that's - boy, that's pretty scary.

Steve: Yeah. There was another one, an MLD protocol, Multicast Listener Discovery, which is part of the IPv6 protocol. It's a little bit less severe. It can cause a denial of service, and there's less vulnerability associated with it. But it was also patched. The big one, though, is this remote code execution. And again, the thing that makes it special is it is not blocked by the built-in firewalls. It is basically a function of the stack running in the kernel. So it runs, not for example with the security credentials of a service which could be sandboxed or could be running with restricted credentials. It runs with the full power of the kernel when the buffer overflow - because, I mean, this is too tasty for the bad guys not to jump on and try to exploit.

Leo: And I don't want to be self-serving here, but it's one reason why it's important to listen to this show. Because I'm looking through all the major tech news sources, and they were so busy covering CNET, I don't see anything about this. And partly they were covering - CES. I just said CNET. CES. But partly, I think, it's, oh, another Windows security flaw. We're tired of that.

Steve: Right. Well, it'll certainly get some news when this thing turns into a worm. And we don't know when it's going to happen. We'll certainly let our listeners know. But just it can't help but to happen. Now, people might say, oh, well, but wait a minute, you know, won't all of these machines be patched? Well, that brings me to another interesting bit of news from the

beginning of this year. Due to another flaw in Microsoft's database for their SQL, there is an SQL injection attack which was launched, and more than 70,000 Microsoft-based websites have been infected in a way that refers people to a malicious website which attempts to install keystroke loggers, using a number of various browser-based vulnerabilities. So, okay, and get this. This was fixed in April of 2006. This was fixed then. So almost two years ago this was fixed. Yet there are 70,000 websites that are not patched current.

So this demonstrates conclusively that for whatever reason there are web servers, for example, which are not keeping up with Microsoft patches. Because if they had patched once in the last almost two years, this would have been fixed. Yet they have SQL exposed in a way that a malicious agent was able to scan these websites and alter - and we talked about SQL injection attacks - basically scan all of the SQL tables, adding their own JavaScript into the tables, which are then being used to present web pages. So it's a way of them injecting - they use an SQL injection attack to put their own JavaScript onto the 70,000-plus websites, which then of course browsers execute. And you know how I feel about JavaScript. But, you know, I understand you have to have it most of the time in order for contemporary websites to work. So your browser then executes this malicious JavaScript which was injected into a benign server that, you know, was innocent except it hadn't been patched in the last two years. So...

Leo: That I understand a little bit better than, you know, servers, people don't want to patch them a lot of times. A, they're not paying much attention; B, you're always nervous what the patch is going to do. But I would hope that all desktop computers are kind of running automatic updates.

Steve: Yes, let's hope. Another news item that has gotten a lot of attention lately, Slashdot picked it up, relates to the so-called "stealth MBR rootkit." MBR is the so-called Master Boot Record, which is to say it's the first sector on our hard disks. And what's not understood, most people think of that as the partition table because it does contain the partition table. But that first sector is actually executable and executed code. So the way this works is when any - excuse me, a little hiccup. When any PC is first booted, the contents of the first sector are copied into memory, and the computer starts executing code from the very first instruction, which actually contains a - typically contains a jump instruction. But basically it means that that first sector is executed.

Well, this has been used to good advantage, for example, by people doing multiboot managers where they'll put additional code in other sectors on the first track of the hard drive, which is normally not used. A contemporary hard drive track contains 63 sectors. So the first one is the so-called partition sector, which is also the master boot record. Then the other 62 sectors are typically unused, and the first partition starts at the end of the first track with the beginning of the second track.

Leo: This was a very common technique for viruses years ago, Michelangelo and others. You'd put an infected master boot record on a floppy, and it would spread itself that way.

Steve: Well, what's now been done is there is an MBR rootkit that patches the Windows kernel on the fly in order to install a rootkit into the Windows kernel whenever it boots. So, you know, we've talked about who's on first as being the competition with a rootkit versus the operating system. And if something is able to run before the OS is able to protect itself from anything that could run subsequently, well, it's compromised. And this, you know, this does now exist. It has been found in the wild.

Leo: Wow. So how would you get - I guess you'd have to have maybe a boot CD?

Steve: Oh, no. I mean, it would - it turns out that Windows protects everything except that first track of the hard drive. So you could get this installed in the standard spyware...

Leo: Just running an application...

Steve: Yes, exactly.

Leo: Oh, dear.

Steve: Exactly. So it is, it's worth mentioning that this exists. It's not a good thing. But it was even foreseen some number of years ago. It's like, okay, well, this is theoretically possible. And that theory has now been turned into reality. So, and I did note that the iPhone has its first trojan. There is now a trojan for the iPhone which, you know, people go browse a malicious website, and unfortunately it uses some vulnerabilities in the iPhone browser in order to install a trojan which causes a great deal of grief, apparently, when it's removed because it's difficult to remove it cleanly, and people end up messing up their iPhones as a consequence.

Leo: All right.

Steve: So anyway, lots of interesting security concerns here at the beginning of '08. And we'll be keeping our listeners informed week by week and blow by blow.

Leo: Yeah. I'm glad we're going to start doing that. And clearly this is the week to start. Boy.

Steve: Absolutely.

Leo: Are you ready to do some questions now, Mr. G.?

Steve: Let's do some Q&A.

Leo: 12 great questions from 12 wonderful listeners, starting with J.P. in Sydney, Australia. He wanted some VeriSign token clarification. You and I both have used that football and now use that VeriSign card, which I love, my VIP card. He said: If somebody were to get a hold of the VIP event-based credit card token for a short time, couldn't they push the button a few times, write those numbers down to be used later, assuming they also know your username and password? I'm guessing this wouldn't work with the football since it's time based.

Steve: Yeah. This was a great point that was raised. And it raises a few sort of interesting points. First of all he says, well, if someone gets your credit card, your VIP credit card, which is event based, meaning that when you press the button you get the next number, you press it again, you get another number. But that's entirely deterministic. That is there's a counter which is being incremented. The counter is being encrypted and hashed through a well-understood and well-known algorithm to produce that unpredictable number, well I should say unpredictable to someone who doesn't know the key which that card also contains, the

cryptographic key.

Leo: Which is everybody except the site you're going to.

Steve: Well, actually it's everybody but VeriSign. VeriSign knows, and the site you're going to then checks in with VeriSign to say is this the proper next number for this guy. And of course as we've discussed there's a window of those. Remember we had a listener who liked to push his button on his card a lot, and he'd moved that counter so far ahead it was outside of the tolerance window that VeriSign maintains, and so he had to go through the extra resynchronization process because he was just having too much fun...

Leo: It's not a big deal. You just enter the number a couple more times so it can figure out where you are.

Steve: Exactly.

Leo: So once you use a number, it can't be used again. But this guy's saying, well, you could get a couple of numbers, you could stack them up. But it's only going to be good until I use it, though; right?

Steve: Well, but what he's saying is he's saying, okay, if I borrowed your or sneakily got a hold of your VIP event-based card and wrote down a few numbers, and I also knew your username and password - well, okay, stop. What he's saying is, if I have all of your multifactors, then I could log in.

Leo: Right, yes, you could.

Steve: I'm like, yes, absolutely. You know? If you've got something I know and something I have...

Leo: But here's the question. Okay, so he gets a couple of those numbers and gets it back in my wallet. But as soon as I use that card again and give the current number, all of those older numbers are invalidated anyway; right?

Steve: That's absolutely true.

Leo: Okay. So it's for a limited time those are going to be any good.

Steve: Well, it's not limited time except in the sense of limit he'd spend.

Leo: Eventually I'd use my card, yeah.

Steve: I would say a limited event. Now, the point is, though, he says, that won't work with

the so-called football because we know that it changes every 30 seconds. And there's a window of what is it, plus or minus three minutes or something. I mean, it's 30 seconds, and it's plus or minus five. So, yes, three minutes. So if he didn't use the number on the football, and he did have your username and password, then that wouldn't work. And so first of all, he's correct. Secondly, I wish our VeriSign cards were time based rather than event based, for exactly this reason. Except I don't think you can do, practically at this point, a time-based system...

Leo: They're not smart enough.

Steve: Well, these things are thin. I mean, you'd have to have a crystal time base somehow in something that is literally no thicker than a credit card, and also have plastic on both sides and some goo in the middle of the sandwich. I mean, it just - I don't know how you'd do that. And time is a problem because that would be consuming much more power. The beauty of the event-based card, which uses the eInk display, as we know, eInk requires zero power to keep it displayed. So literally, when you press the button, there's a moment of power usage from a no doubt very small and low-capacity battery which has also somehow been sandwiched into this thing, I mean, the reason we love the card is it's in our wallets. The reason we're not so crazy about the football is if you stick that in your wallet, you're going to need to go to a chiropractor.

Leo: And as you point out, they'd have to get physical access to your wallet, your card, write that down without your knowing. I think...

Steve: Well, yes. I mean, his question is, if I had all the factors of your multifactor, you know, wouldn't I be able to log in? It's like, yes, you would. But the point is, you know, you're not supposed to get all the factors of our multifactor. And he's right, though, that the football, because it's changing constantly, one of the factors, that is to say, that coming from the football, it gets stale. It's going to be stale in three minutes. And so if you got it you'd have to use it quickly. You couldn't write a bunch down and then be logging in until, as you said, Leo, until you used yours, which would immediately obsolete all previous numbers.

Leo: Robin in Langley, BC - British Columbia - has been thinking about Matthew's Mega Hash login dilemma. I loved that. If you don't remember that, listen back to Episode 120. It's funny. Anyway, he says, Robin says: I realize the problem with Matthew's login scheme which you described in Episode 120 - multiple secure hashing, then just capturing the results since the connection is not encrypted. However, it occurs to me that there may be a simple way to fix the problem. I was wondering what you think of this solution. Since Matthew is creating both the client and server sides of his web app - the whole idea of this, though, was that Matthew didn't have to use SSL, he could create his own kind of security system. Couldn't he simply have the server generate and supply a unique login ID, a serial number of some sort, to the client, with the original login form generated by the server? Then, using the client script, hash - I'm trying to follow this. You can follow it. I won't try to follow it. Then using the client script would hash both this ID and the user's password data using Matthew's Mega Magic encryption scheme, using an encrypted blob that is sent back to the server for authentication. A man-in-the-middle replay attack would be useless then since the encrypted blob would be unique for each login. Obviously this won't work if Matthew is trying to specifically not use encryption/decryption on the server side. Your thoughts?

Steve: Well, this is a great idea. And several listeners, astute listeners, who are clearly enjoying coming up to speed about crypto technology, responded about this. The idea being that what Matthew described to us was he would have an algorithm in JavaScript code that was being delivered from the server, that would run on the browser client. So that basically when

the user logged in, it would obscure the user's login name and password by hashing it down into a cryptographic blob, which would then be sent back to the server. The server would know what the proper cryptographic blob was.

People, and we had discussed this before, recognized, wait a minute, all you have to do if you're monitoring and have the opportunity of being the man in the middle, all you have to do is capture the blob and then send the blob yourself. You don't have to know what the original username and password is because the whole point of this is that Matthew is not going to be encrypting his connection. And so of course that's absolutely right.

So what Robin has suggested, as well as some other listeners, the server could send something to the client which is different every time. And in cryptography it's called the "nonce," that is, something which is just used once and never again, the idea being so it would send, like, a serial number to the client. The client would add that to the hash. And what that would do is, that would mean that this blob would be different every time because this nonce coming from the server would never be duplicated. And that would prevent a replay attack. That is, essentially the blob could only be used for authentication once and never again.

Now, that's certainly the case that you could do this. But if you were an active man in the middle, that is, if you had the ability to filter the traffic going in each direction, you could simply log in by capturing these nonces and essentially using the client to solve this puzzle for you, intercept its response, and then use that to log yourself in. So again, it's still, I mean, it is a real problem with not having a secured, cryptographically strong connection because there are all kinds of games that can be played, depending upon what level of access an attacker is able to get to your connection. But certainly it was another interesting application of all the crypto that we've been talking about in prior weeks.

Leo: It's a thought exercise.

Steve: Yeah.

Leo: Tyler Menezes in Redmond, Washington had an ALT-ernate password idea: While setting type for a brochure, I thought of an interesting idea. If you were to use a nonbreaking space, instead of hitting the spacebar you hit ALT+160, actually 0160 on the number pad for Windows, in your password, wouldn't this make it much harder for keyloggers to get your password? The attacker would see a normal space, which is not the same as a nonbreaking space bit for bit. So the application would reject it. If I'm worried about someone intercepting my keystrokes or perhaps looking at my saved Firefox passwords, if I didn't save a master password, would this be a good solution? I'm a big fan of the show. Keep the awesome shows coming. P.S.: SpinRite rocks.

Steve: Well, I thought this was sort of an interesting idea. I use the ALT key myself. I think ALT+249, or maybe it's 0249, is a bullet. And so even in my source code, where I wanted, like, to put in some bulleted points, I'll just use that. It's a PC-only sort of feature, which has always been around, where you're able to hold down the ALT key, and then on the number pad you're able to sort of manually dial in a code which doesn't have to be within the normal 256 ASCII characters, but it allows you to access a much wider range of characters. The problem is that it is - it tends to be application specific, that is, some applications understand that keyboard sequence, and others don't. So I would say, well, your mileage may vary. It's certainly an interesting idea.

And so, for example, if Firefox did allow you to put in the so-called High ASCII or Unicode characters, that is, characters outside of the normal character set, so if Firefox allowed that, then you could certainly obscure the fact that a space in a password was not really a space, it

just looked like a space, while not actually being that. So, I mean, that's certainly a possibility.

Leo: The problem is a keystroke logger is logging the actual keys you type. So it knows exactly what you type.

Steve: Exactly.

Leo: It's not fooled by how it looks because it's not looking at it.

Steve: The problem is there is a complex interaction of different layers of keyboard handling. So at some level the output from the keyboard is going to just be ASCII, or it might be 16-bit Unicode. Somewhere else it's actually the individual events of keys going up and down.

Leo: The scan codes, yeah.

Steve: Yes. So, exactly, scan codes from the keyboard. So if you had a keystroke logger that was intercepting at the scan code level, and it was able to interpret those scan codes into their equivalent Unicode, I mean, I guess...

Leo: Well, if it's ASCII, the nonbreaking space is a different ASCII symbol. So even at ASCII level it's going to catch it.

Steve: Correct.

Leo: It only works if somebody's looking. I guess if they're looking at your Firefox passwords, they might not notice that.

Steve: That's exactly what I was going to say, was that if you had passwords that had obvious spaces in them, and if Firefox treated these alternate characters in a compatible way, then it could work. So it's an interesting idea. And I would say the reason I think your mileage may vary is, well, try it, but don't count on it because you might find out, first of all, that a keystroke logger could be smart enough to track the individual scan code events and figure that out; or you might have applications that are incompatible with the whole concept.

Leo: John Campbell in chilly Bozeman, Montana, looks up DNS without the help of his ISP: In a past episode you talked about having your Internet service provider's DNS server track your movements. I found a solution, not for the faint of heart: TreeWalkDNS.com. This is a DNS server you set up on your local machine that bypasses your ISP and does the DNS lookups directly. It can also be used as an ad blocker and to block access to hostile sites. I use it on my laptop and at my house. The DNS software is simple to install. Setting it up to do ad and hostile site blocking is not so simple. It also has the advantage that you are not depending on your overloaded ISP's DNS servers, and it caches DNS lookups locally. I wish they had a Donate button on their site.

Steve: Well, this is an interesting piece of email for me because the two guys who are behind TreeWalk DNS are longtime GRC newsgroup participators; and a whole bunch of GRC

newsgroup users, because there's been lots of dialogue in our newsgroups about this, do use and love TreeWalk DNS. So a couple comments. First of all, I should have mentioned at the top of this, but I'll say it now, the show notes for this episode, Episode 126, has a whole page of URLs. So TreeWalkDNS.com is spelled exactly as it sounds, but there's also a link to it on our show notes page. And we've got a bunch of other URLs we'll be coming to in subsequent questions, which there's no way to pronounce them or spell them out. So I wanted to make sure that people listening to this know that the show notes for this episode contain all the links that we're talking about.

Leo: Okay. And we do that in the show notes on the site, too.

Steve: Right. One of the things that ISPs are sort of notorious for is referred to in this email, and that is the overloaded DNS servers. Many ISPs sort of regard DNS as the unwanted stepchild service that they have to offer their users. They often have servers that are small or old, sort of dusty things in the corner that never get much attention. DNS is not a very glamorous service. ISPs have to offer it. But it can be the case that the DNS servers are old and slow and in fact overloaded. Because the ISP grows and grows and grows, increases the number of users they've got. All users' computers are generally aimed at their ISP's DNS servers to perform the recursive DNS lookup on behalf of their requests. So those DNS servers end up being slow. And as we know, anytime you put a URL into your browser, unless your local machine already has it cached from being used before or if that URL exists in your hosts file, which is a substitute for the whole DNS process, if neither of those is the case, your computer then asks typically your ISP's DNS server to look up the IP. So everything comes to a grinding halt until you get a response affirmatively or negatively from that DNS server about what's going on.

So the idea here that John is talking about is to, instead of using your ISP's DNS servers at all, run your own. That is, have a DNS server running in your computer, and have it do exactly what the ISP server would have done on your behalf. And so the potential advantage is, first of all, presumably your own server would never be overloaded because it's not doing any serving for anyone but you. The downside is that an ISP's servers might already have, for example, and would probably have, for example, www.aol.com in its local cache, and Microsoft and MSN and Amazon and, I mean, all of the common URLs might already be there. In which case you're taking advantage of all the other users who have asked that common DNS server, because all of the ISP's customers are sharing that DNS server, you're taking advantage of the fact that the popularity of popular sites would have caused the result, the IP already to be known by that local server. On the other hand, it's just as likely if you're browsing around that that's not going to be the case. And so having a server running in your own machine would allow you to get more performance.

Now, John also mentioned the privacy aspect, which is what we had touched on a couple weeks ago. And that is that, you know, if anyone cared, if an ISP cared, they're able to determine where your computer goes because your computer is always asking for it, that is, the ISP's DNS server to look up the IP of any domains you want. So from a privacy standpoint there is some compromise there in that your ISP could be tracking that. If you run your own DNS server, you're not asking your ISP, but you are asking other servers, other DNS servers on the 'Net, and the request is coming from you. So it's sort of a privacy tradeoff. On one hand, your ISP would know if you were asking for a specific site and know who you were. But the site and the servers that it has to ask would only see the request coming from the ISP and not from you. So it does, by using your own DNS, it cuts out the middleman, which can be a benefit for performance. But it does mean that your own IP is the one which is now making these requests, rather than your ISP's on behalf of you.

Still, from a performance standpoint, what I hear is that it's a win. And I should say I'm running my own DNS servers. I have a DNS server that we run, GRC runs at our facility at Level 3. And I ever have one here at home on a UNIX box. And it was these guys who did TreeWalk DNS

that helped me through the initial hurdles of getting my own local DNS server set up correctly and running. And they really do know what they're talking about with DNS.

Leo: I have to say that there's a potential, as you point out, for it to be much slower that you don't have - the lookups haven't been done on a lot of the sites you're going to go to. So you're actually going out to a more distant server than your own ISP's server to get that information. You've got to get it looked up somewhere. I just, you know, if you're worried about, you know, if your ISP has a lousy server, you don't have to use your ISP's server. I know a lot of people use, believe it or not, Verizon's ISP servers because they have - or DNS servers, I should say. They have notoriously fast DNS servers. I don't really recommend that. I use a company called OpenDNS, absolutely free, it's OpenDNS.com. And they're faster servers, and they have some additional features, including filtering. Doesn't solve the problems of privacy issue, but I'm not sure you've solved the privacy issue by running your own server anyway.

Steve: Right. And I guess OpenDNS also supports an additional set of sort of off-the-beaten-path, top-level domain names.

Leo: Right. Also if you mistype .com, they'll fix that. And they have a very helpful, instead of a 404 they'll give you a search results, which is how they've monetized themselves. There's some advertising on the right there. But actually if you create a free account with OpenDNS.com, you can turn on filtering, which I use at home. And unless your kids are smart enough to change, manually change the DNS server on their computers, you're set. You just set it on the router, and you're good. And it works very well. They have no idea. So, I mean, I guess there's some advantages of having, running your own DNS server. But it seems like a lot of work.

Steve: Well, our more techie users do enjoy it. And the TreeWalk DNS server is well packaged. Basically it is a Windows port, or a Windows build, of the Internet standard BIND.

Leo: Oh, it's BIND, oh, okay.

Steve: It's BIND. Yeah, basically it's BIND running on Windows.

Leo: Well, then there's another issue you should be aware of, because BIND just has a notorious - notorious for security flaws. So remember, you're running a server now. And that means you're opening yourself up to possible attack. I'm not sure - all right. If you want to do this, go ahead. Be aware of what you're doing. OpenDNS would probably be better for most people.

Athol Wilson in Auckland, New Zealand had a thought about the Romper Room Cipher: Your Symmetric Ciphers podcast did the trick. Finally the penny dropped. With 2^128 having so many billions of zeroes, it's now obvious that even a 5GHz processor could drop 10 zeroes off the end, and there would still be an awful lot of seconds, hours, days, millennia left. I did expect you to cover double encryption using Double RRC - Double Romper Cipher, though. I have never heard of this. Where of course to succeed in a brute force attack one would have to retry each of the 256 keys 256 times to find any plaintext. I also - maybe you can explain that. I also discovered a great Flash animation of the Rijndael - is that how you say it?

Steve: Rijndael.

Leo: Oh, Rijndael. I never saw it spelled out. I've heard you say it many times - Rijndael encryption process. It would be great to share it with other listeners. We'll put, again, that link in our show notes: http://www.iaik.tu-graz.ac.at/research/krypto/aes/old/~rijmen/rijndael.

Steve: Yeah. It's funny because I started out talking about symmetric ciphers in the context of explaining why it was clear that for a symmetric cipher like we've been talking about, double encrypting with different keys would give you fabulously more strength, essentially, you know, a lot more strength. And at the same time, I developed this notion, because I also wanted to clearly explain how that was and why, we developed the so-called Romper Room Cipher, which you will remember was a trivial little cipher that used I think a 4-bit key...

Leo: Oh, this was your explanation of how they worked, okay. Yeah, yeah, yeah.

Steve: Exactly. It was a 4-bit key and an 8-bit, that is, a 1-byte cipher, 1-byte block length. So he was saying that, gee, you know, you went to the trouble of developing this concept of a simple Romper Room Cipher, but then never used that to explain how, in that context, how double encrypting would work and, as he said, how you would have to try all of the keys twice, that is, each key, and then try all the other keys against that, basically, you know, reverse engineering or doing a brute-force attack on double encryption. So I thought, yeah, he made a really great point. He also found, or actually reminded me of an animation I had once seen a long time ago. And we've got the link in the show notes. It's a Shockwave Flash animation. It is so good that I grabbed a copy of it to keep local in case it ever goes away. And also it gives the credits to its authors in the Flash, so I didn't think anyone would mind if I, you know, pulled a copy off of the original website.

Leo: Oh, good. So he gave us an Austrian website which is just, you know, long. But you're going to host it?

Steve: Yes, I am hosting it. And the link, I think, is GRC.com/miscfiles/RijndaelAnimation.zip. I zipped it because it got half the size. And it is really nice. It's a little on the techie side. I mean, so you'd have to sort of be comfortable with binary and so forth. But the guys that put this together, I think if you coupled that with listening to Episode 125, last week's episode on symmetric ciphers, where I describe how Rijndael works, together, these are like the diagrams I never drew for this description. And it's beautifully animated, showing things XORing and the code zipping around in loops, going through the multiple rounds and all that. So anyway, for our listeners who are interested in seeing something happen with animation, check out the link in the show notes. It's really worth taking a look at.

Leo: Cool. Rijndael is, for those like me who've only heard Steve mention it, R-i-j-n-d-a-e-l.

Steve: It's actually sort of a contraction of parts of the last names of the two designers of the cipher. So it's sort of a synthetic word.

Leo: It's Dutch. Andrew Ayre in Perth, Australia, wants to make use of Rijndael: Hi, Steve.

I work for a small software development company where I and all of my colleagues are regular listeners to Security Now!, as well as some of Leo's other podcasts. So an interesting debate arose within our office after last week's episode, Symmetric Ciphers #125. We're wondering if you could settle it for us. Does this happen to you a lot? Can you settle a bet for us?

We're in the process of implementing a .NET web service application that makes use of Rijndael 256-bit symmetric encryption to encrypt data that is then passed to the web service rather than in the clear. We originally intended to hard code a Rijndael key and initialization vector on both ends, client and server. But we are now thinking that might not be such a good idea as we'll need to reuse this hard-coded key and IV over and over again. I think this might be equivalent to reusing a one-time pad over and over again, a big no-no. I don't think we'd be susceptible to a brute-force attack, but we might be susceptible to some kind of, I don't know, statistical attack? One approach suggested to get around this possible program is to somehow use the current date/time as a way of salting the Rijndael key or IV in some way. That way the one-time use pad is never reused, and we can still hard code the Rijndael key and IV on both the client and server. Just hash it, I guess. We'd all be very interested in your opinion of our dilemma.

Steve: Well, this was a really great application question for security. There are a couple ways of using a symmetric cipher like Rijndael to encrypt communication. The simplest way is called the Electronic Code Book, or ECB, which is really just a way of saying you simply take 128 bits at a time, or that is to say the block length of the cipher, that is, you know, the number of bits that you feed into the cipher to get that same number of bits mapped to a different combination out. And in the case of Rijndael, no matter how long the key is, Rijndael is, as we know, a 128-bit cipher. So you would take 128 bits, like the first 128 bits of your message, encrypt it into a different 128 bits, and that's your cipher text. Then you take the next 128 bits, encrypt it, and that's the next block of cipher text. And so you go along. And that's a so-called Electronic Code Book, which is just a shorthand for saying, you know, the cipher is the so-called code book, and all you do is take blocks of plaintext, run it through the code book cipher, turn it into cipher text, and then take the next block.

That makes people uncomfortable, even though Rijndael is very strong, even though we know with 256 bits it's infeasible to do a brute-force attack. The reason it makes cryptologists uncomfortable is that every time the same 128 bits appeared in the message, it would encipher to the same resulting 128 bits, given the same key. So if the key didn't change, it's very clear that some information is sort of leaking out, you might call it "inferential leakage," because given enough analysis of the communication, it might be possible for someone to infer meaning just from the repetition of certain parts of certain types of repetition. I mean, there is some leakage of something. Which makes the cryptographers say, uh, you know, can't we avoid that, too?

In order to do so, they came up with this notion of block chaining. Cipher Block Chaining, CBC, is the most popular of these. With Cipher Block Chaining, you take a so-called initialization vector. Now, notice in the standard Electronic Code Book there was no initialization vector. That is, you simply took the message, the first 128 bits, you enciphered it, and that was your 128-bit result, then you go to the next block. With what's called Cipher Block Chaining, you start off with a so-called initialization vector, which is the same size, the same width in bits as the cipher. And so in the case of Rijndael it would be an 128-bit IV, or initialization vector. You XOR the first block that you're encrypting with this initialization vector. And as we know, what an XOR operation does is it conditionally inverts the bits. So where the initialization vector contains 1 bits, the bits being input will be inverted. And that's nice because it's a reversible process. Remember, anything we encrypt is only useful to us if we're able to decrypt it at the other end.

So we take this initialization vector, XOR it with the first block of our plaintext, and then that's what we encrypt. We encrypt that XORed result as our first block of cipher text. Then we take

that first block of cipher text and XOR it with the second block of plaintext that we're encrypting. And we then encrypt that to create the second block of encrypted results. And similarly, we take that result, XOR it with the third block of our input, and encrypt it to produce our third block of encrypted output.

What that does is, first of all, it introduces another 128 bits of uncertainty into the process. We already had, for example, a 256-bit Rijndael key. Now we're adding another 128 bits, which is unknown to an attacker. If we weren't doing that, then the attacker's problem would only be the 256 bits. On the other hand, we know that's a big problem. So that's not, you know, making it even more impossible, you know, when it was already impossible. It's like, okay, well, I guess we're even safer than we were, even though we were safe enough.

But more importantly, what it means is that this initialization vector is propagated through the entire message. Because we take the output of the first encrypted block and XOR it with the input of the second encrypted block, everything about the message, the initialization vector and all the preceding bytes affect, that is to say, influence everything that comes afterwards. So no longer do we have the situation we did before with the so-called codebook approach, where each block of 128 bits stands by itself. Now the entire history of the message affects the next byte's result. Which means that even if the plaintext had repetition in it to any degree, all of that would be masked in the result. And the beauty of this chaining approach, where the result from one is mixed into the input of the second, is it's reversible. You're able to decrypt this knowing both the original key and the original initialization vector and undo all of this, which of course is required for decryption.

So now that we've got this sort of background, all these guys have to do is include with their encrypted data a randomly chosen initialization vector. That's all they have to do. Essentially what that will do - and this initialization vector is also - it would be called a nonce, a one-time token which is not going to be reused again for security. And even though it would be in the clear, that is, it itself would not be encrypted, that doesn't matter because it's changing every time. And you could have it added to a secret initialization vector, that is to say like XORed, so that even the true initialization vector that was being mixed into the plaintext was never known.

And so the idea is, in this client and server mode, the system would have a 256-bit symmetric encryption key which would be secret for the client and for the server, as Andrew described it. When you wanted to send a message what was securely encrypted in either direction, you would send a first 128 bits which is the initialization vector, followed by the encrypted result. And the receiver would receive the initialization vector and use that to start the process of decryption. And so since the initialization vector is changing every time, but it is obscuring all of the patterns in the cipher, it's entirely secure to let the initialization vector be known in the communication channel. You don't have to, for example, secretly use the time of day or date or somehow otherwise synchronize the encryption and decryption of the endpoints. You have the originator who's doing the encryption just choose a random number or an incrementing number, whatever they want, stick it on the beginning of the message, send it to the other end. And even though that would be done unencrypted, in fact, that channel would be in the clear, an attacker could see the IV, that 128-bit initialization vector, and it wouldn't help them in any way to decipher the rest of the message because it's solving the problem of statistical patterns which would otherwise be present. And then we're relying on the strength of Rijndael's 256-bit key, which we already know is massive strength.

Leo: Massive strength.

Steve: Massive.

Leo: Massive. Well, there you go, Andrew. Talk about free consultation. David Eckard in Durham has been counting his toes: Steve, he says, I have listened to all 125 Security Now! programs. It is possible to pronounce the number of digits in 2^128, which you were talking about in Security Now 125. The exact value of just 2^128, not its factorial, is - and by the way, you are now going to understand why scientists use scientific notation instead of actual numbers - 340 undecillion, 282 decillion, 366 nonillion, 920 octillion, 938 septillion, 463 sextillion, 463 quintillion, 374 quadrillion, 607 trillion, 431 billion, 768 million, 211 thousand, 456. And now you want that number factorial? Sheesh. Just the number of digits in 2^128 factorial is 1 duodecillion, 296 undecillion. Do we go above - I guess we could. We could keep going on.

Steve: There apparently is a web page, and I think David referred to it in his message. But it seemed a little bit superfluous. There's a web page which explains how you can keep going forever.

Leo: Oh, that's funny.

Steve: I mean, there is a - it's a well-known...

Leo: I'd have to be. It'd have to be.

Steve: Yeah, well, yeah, okay, I guess you're right, there would have to be. But there's a well-known discipline for just continuing out as many, as far out as you wanted to go. But he spelled all this out for us, so I thought our listeners would get a kick out of that. And of course we're doing cryptography, so we're all about big numbers.

Leo: Big numbers. But really, scientific notation is just fine. Keith Stein in College Station, Texas is feeling disconnected: Steve, I've become increasingly dependent on Skype over the past couple of years. I know how you feel, Keith. I work for a company of 50 employees. We use Skype as a major communications tool, mostly for the texting feature, of all things. We were notified today by our IT department that the security group has identified Skype as a security risk - oh, please - and they'll now be removing it from all our systems. I know you use Skype for your podcasts with Leo. How much, if any, is it a security risk? I haven't heard anything about a security risk in Skype.

Steve: No. Well, okay. So let's discuss theoretical security risk because that's really the only thing we can discuss meaningfully.

Leo: Now, it is an open source. So there could be all sorts of holes in Skype nobody knows about.

Steve: Well, precisely. So, for example, most typical Skype users are behind NAT routers, our ever-loving NAT router that we preach about here. I mean, I'd be behind a NAT router even if I only had one computer on the Internet because they only cost $49 now, and you can even find them for less than that. And as we know, when we were talking about this nightmarish new security vulnerability in the Windows stack, you really want to be behind something other than your computer's own local personal firewall. So most people behind - I'm sorry. Most people using Skype are going to be behind NAT routers, which is going to protect them.

On the other hand, as we know and we've discussed before, there's the problem with the so-called Skype Supernode, which is inherently an exposed and accessible Skype client, meaning it's someone running Skype whose machine has a publicly accessible Internet address, you know, 70.326. whatever, I mean, something, you know, or 224. or any of those that you typically see your router's IP being, your router's public IP. If someone actually has their computer on the 'Net, and they've got Skype running in a supernode node, meaning that it's able to accept incoming packets, and if there were, as you were just saying, Leo, some sort of buffer overrun, not widely known or unpatched problem, then potentially people could look around for those Skype clients. And as is the case with any server, because that's what a supernode is, a supernode is a server, could use that in order to attack them.

Leo: So the risk is because Skype bypasses firewalls. It's bringing stuff in that you can't control.

Steve: I would say the risk is that because Skype would - Skype is always trying to be a supernode.

Leo: Right, they're offering a service, in other words.

Steve: Right, well, it - yeah. Because it would like to be a traffic relay in order to help those who are behind unfriendly NAT routers, where it's unable to do the NAT penetration for you. In that case, if both users are behind unfriendly NAT routers, then there is no way for Skype Central to negotiate the connection between two clients both behind unfriendly NAT routers. Skype does not themselves offer a relay service, where for example Google Talk does offer a relay service. Instead, the Skype system looks around for exposed Skype clients, that is, that are not behind any kind of NAT router, and uses them, without their owners' explicit permission or knowledge, uses them to relay traffic to clients that are behind unfriendly NAT routers. So if Skype had a security problem, and if it were operating successfully as a supernode, then it would be accessible for attack. Also, even if normal Skype were behind NAT routers, I mean, it is a peer-to-peer network. And I think frankly that's probably what the IT...

Leo: That's the real problem, yeah.

Steve: Yes. I think that, you know, peer-to-peer has gotten such a bad reputation that it may just be the fact that it's a peer-to-peer network that has got the IT department spooked. They just don't like the idea that you are connecting to another peer. On the other hand, as we read, or as we know from the beginning of this podcast, 70,000 web servers, actually many of them were belonging to .gov and .edu and .mil networks, they were just taken over because they hadn't been patched in almost two years. So it's not like a client-server relationship is necessarily any safer to use than a peer-to-peer network.

Leo: So I guess I'll take it back. I guess there are risks, and you should be aware of them. And if your company says don't, you can't. I wonder if there is, well, of course there's web solutions that don't - they're not peer-to-peer, they don't do NAT traversal.

Steve: Well, yeah, in fact I was just wondering whether - I couldn't really tell from Keith's note whether they're using Skype's texting...

Leo: Sounds like they're using it for an IM client.

Steve: Well, that's exactly what I was going to say. And it sounds like they're using it within their corporate perimeter, in which case there's lots of solutions for just texting the guy two offices down.

Leo: Well, there's encrypted chat, too, which you probably should use.

Steve: Right.

Leo: Rob in Pennsylvania wants to watch his cores closely: I thought I heard Leo say on an episode he had a program to monitor each individual core on his quad core machine. I do. I just built my first quad core monster. I thought it would be cool to see how much each core is being used. But I'm on a Mac, so I'm not going to be able to help you.

Steve: Well, so I thought I would ask you to remind us what you use for monitoring your quad cores, and then I will tell everybody what I use for monitoring mine on my Windows machine.

Leo: Yeah. So on the Mac I have a little thing called MenuMeters that is a really handy little menu bar item, and it lets you do a lot of things, including CPU. But you can also watch disks, memory, network usage, and more. But I just keep the CPUs up. And it's very helpful, you know, if your system's starting to get bogged down, you can look up and say, oh, I see why. Something's going on. Now, you know, Apple just announced its eight core. So I presume this would work with eight cores, too. Now, I'm sure there are a lot of choices on the Windows side.

Steve: Well, actually the one I use is just the one that's built in, the good old Windows Task Manager. If you click to the processors tab, it will show you however many windows you've got threads of execution. That is, even on a single-core, hyperthreaded chip, it'll show you two windows. And on my quad core machine I get four. So it's just - it's built into Windows. It's just the Task Manager, which is easy to bring up. And one of those tabs will allow you to see what each of your different cores is doing, showing you a little graph, you know, in real time. And, for example, I recently updated my favorite MPEG compressor to take advantage of all the cores. And my goodness, does it run fast on that machine.

Leo: Oh, that makes a big difference.

Steve: I mean, it just saturates all four of them, and it just cruises through media for compression purposes. I mean, that's an example of something that really can take advantage of quad cores. Most of the time all four of those are sitting around looking at me, saying, okay, you know...

Leo: Give me something to do.

Steve: You got four of us here.

Leo: Give me something to do.

Steve: Yeah.

Leo: You know, the Windows Vista gadget bar has a CPU monitor. I imagine it would look at all four processors. And you said that it comes with Windows. Of course it comes with the Mac, too. There's an activity monitor does exactly the same thing. There are lots of little gadgets that do this. It is nice. It's kind of fun. I was worried that these things are using up cycles, and maybe I shouldn't be running them in the background, but...

Steve: I did discover one thing, speaking of that, Leo. Running it on my laptop, I mean, traditionally I have had the Task Manager, a shortcut to Task Manager, in my startup folder so that it just always runs. I have it set up to start minimized. And there's an option to minimize to the tray. And that way I just have a little cute little rectangle down in my tray that shows me how much the computer's working, which is just, you know, as a techie I find it useful. The problem is, running it on a laptop, every single time it updates it pings the hard drive. And it keeps the hard drive from ever shutting down to conserve battery life and to keep the laptop running cool. So I've learned, oopsie, and that's a habit I need to break for my laptops because I like them to be able to spin down, instead of having some dumb Task Manager sitting there going ping, ping, ping.

Leo: Wonder why it hits the hard drive? It doesn't really need to hit the hard drive.

Steve: Eh, Windows, you know, what are you going to do?

Leo: It could keep SpeedStep from kicking in, though. It might say, oh, I'm never idle. I'm always busy looking at how busy I am.

Steve: I wouldn't be surprised.

Leo: I'm not idle, I'm busy. Ryan Couture of Enfield, Connecticut is wondering about more iron in his diet: Steve and Leo, I've been listening to your show since Episode 1, and I must say you've taught me more about security than I could ever have learned on my own. Keep up the amazing work. Thank you. I had a question about a product that I saw while looking for a flash drive online. It's the IronKey thumb drive, IronKey.com. It claims that it uses hardware encryption to protect your files and has a crypto chip that will self-destruct the encryption keys if a physical attack occurs. It looks good, and I was hoping to have a second opinion to the technology. I was wondering if you could do a mini review of it, or at least clarify if it's worth the premium. Thanks again, keep up the good work.

Steve: Well, I know what he means when he says "worth the premium" because I bought two of them. It was - the 4-gig one from Amazon was $138.

Leo: That's outrageous, yeah.

Steve: However, I put Ryan's question here, and I went to Amazon and clicked the "yeah, send

it to me" button because I have to find out. Many, many, many listeners have asked about IronKey.

Leo: We've shown them on the TV show. I mean, they were nicely made.

Steve: Well, in fact, I clicked - you and I are recording this podcast two days late because you were under the weather when you got back from your vacation. So I had done all of the production work for the podcast and ordered these IronKeys, the first of which came yesterday. Then I looked at it, and it's gorgeous looking. And it's like, okay. And the reason I bought two is I'm going to open one up.

Leo: Oh, you did?

Steve: Oh, yeah. Yeah, yeah. I've got to find out if it, you know, it's like some weird smoke and some - and if I hear "Mission Impossible" music, you know...

Leo: Right [vocalizing]. This key will self-destruct.

Steve: And I'm a little annoyed that the box says "military grade encryption." It's like, okay, well, what encryption? Tell me. So I'm going to do as Ryan and a huge number of listeners asked and check out the IronKey flash drive as thoroughly as I can, and I will report back.

Leo: Good, good. Anand K. in Detroit, Michigan discovered something worrisome about Opera's Mini Browser. Mini Me. I use it. He says: I use a Blackberry Curve and dislike the default browser that comes with it, so I downloaded Opera Mini. I have, too. Got it right here on my Curve.

Steve: Keep listening, Leo.

Leo: Tried to run it. It won't connect to the Internet. So I had to do some debugging what was going on before I could get it to work. In this process I realized that Opera Mini actually talks to a transcoder server, which I assume is like a proxy to get its data. All requests go to this transcoder server. After searching for documentation on this behavior, I found that it's documented on the Opera Help site.

Steve: And we've got the URL also in the show notes.

Leo: OperaMini.com. In a nutshell, the mandatory use of this transcoder server makes it impossible to provide end-to-end SSL security for client connections. Oh.

Steve: Uh-huh.

Leo: So all of my cookies, userIDs, passwords, and other sensitive information I had so far assumed was secure going over SSL was actually going through this proxy server and

getting decrypted there. Even though it's documented, I'm not convinced a browser should do this. I'm not, either. Hmm. Opera's site explains why they need to do this at the URL I referenced above. But I'm not convinced. They should have left the SSL connection alone, direct, with end-to-end security, and used this optimization for plaintext connections. Secondly, there's no indication given by the software for the user to know clearly that this is what's happening behind the scenes. Is this reasonable in your book? Thoughts on if/how they could have done it differently. Wow.

Steve: Well, this is a perfect example of something we have touched on many times in the last two and a half years, and that is the idea of a proxy server that is terminating the SSL connections itself. That is, essentially decrypting connections that you thought were encrypted in order to have access to the nonencrypted data that is inside the SSL tunnel. Now, the reason they're doing this is that this server that the Opera Mini browser connects to is really doing a lot of good work for the user. It is rewriting pages, web pages on the fly, rewriting JavaScript on the fly, essentially turning web pages that were never designed to be seen on a very small screen on a very lightweight and lower powered browser, making them work.

And so if they didn't do that, that is, if they did pass SSL through end to end, first of all, your browser, that is, that you're holding in your hand, running on presumably a lower power chip, it would need to be able to do SSL, which is a little compute intensive, although I would argue these days that could be handled easily enough. And they would then no longer be able to perform this filtering which apparently the Opera Mini Browser depends upon. On their security page where they address this, they're not quite as upfront as I wish they were. I mean, Anand K., who's a Security Now! listener, he's obviously astute enough to sort of read between the lines.

Leo: I know. I didn't. I didn't know, and I've been using this.

Steve: Yeah, you have to read between the lines to get what it is they're doing.

Leo: I'm mad.

Steve: And, yes, I know, I mean, this is not good for it to be less clear for people. Apparently they're providing some sort of tunnel encryption of their own, not SSL. But that, you know, so your data is protected itself going to them. But then it's completely open. I mean, it's as though you're trusting the Opera Mini server, proxy server. Everything you do, your passwords, your secure login, I mean, literally your username and login that you thought was over SSL...

Leo: Unbelievable.

Steve: ...is unencrypted. And finally, at the end of this FAQ page, someone asks the hypothetical question, well, what if I don't like that? And their answer is, well, then, you can't use Opera Mini. Go use, you know, the regular Opera non-mini browser, sorry. And so, I mean, I don't really have an opinion one way or the other, although I don't think I'm going to use it.

Leo: I just deleted it. I'm kind of stunned.

Steve: So that's annoying. And I really thank Anand for the...

Leo: Yeah. I would not have known. I'm looking at their website right now. It doesn't say that it's doing that.

Steve: No. I mean, again, in their FAQ it says, is there any end-to-end security between my handset and, for example, PayPal.com or my bank? Okay, first word, no.

Leo: First word, bye.

Steve: If you need full end-to-end encryption, you should use a full web browser such as Opera Mobile. Opera Mini users a transcoder server, as they call it, to translate HTML, CSS, JavaScript into a more compact format. It will also shrink any images to fit the screen of your handset. This translation step makes Opera Mini fast, small, and also very cheap to use. To be able to do this translation the Opera Mini server needs to have access to the unencrypted version of the web page. Therefore, no end-to-end encryption between the client and the remote web server is possible.

Leo: You know, I understand why they're doing that. But they really should say - that should be very clear on the front page. Wow. I haven't used it much, so I feel all right. But...

Steve: For what it's worth, I mean, they say - another of their made-up questions. Can Opera software, Opera Software Company, see my passwords and credit card numbers in cleartext? What is the encryption good for, then? The answer, the encryption is introduced to protect the communication from any third party between the client, the browser on your handset, and the Opera Mini transcoder server, meaning - so they're talking about the encryption between your handset and Opera's server. If you do not trust Opera software, make sure - and I'll say, and everyone who works for Opera software - make sure you do not use our application to enter any kind of sensitive information. It's like, okay. As you said, Leo, bye bye.

Leo: I deleted it. Ron Bailey in Dallas, Texas has got to have his options outlined: Steve, in Episode 123 of Security Now!, during the Jungle Disk discussion, you mentioned that you keep everything in outlines. I'm an avid outliner, too, and I'm always on the lookout for new, better ways to work these things. How do you manage your outlines? Do you use software? Is it a text? I gots to know, says Ron.

Steve: Well, ever since - what was the very - was the very first one, ThinkTank, I think.

Leo: ThinkTank, yeah.

Steve: I'm sure you remember that, Leo.

Leo: Oh, yeah, great program.

Steve: I think it was on the Apple II. And it survived a while. It got ported to the PC.

Leo: I think that was Dave Winer's original company.

Steve: I believe that that's exactly right. And then there was someone named John Friend, he did one called GrandView. And Symantec bought them at one point. GrandView was another great - and these were text-based outliners. That is to say they ran, you know, back in the DOS era. And I actually still have GrandView here, and it works, although it's a little funky, and I've pretty much moved over into GUI land. Then there was something really wacky for a while called Ecco, E-c-c-o.

Leo: Oh, I remember Ecco, yeah.

Steve: Yeah, which was - I don't think anybody except the authors knew how that program worked.

Leo: That was really freeform, shall we say.

Steve: It was so powerful that you'd just say, okay, I guess I'm much more stupid than I thought I was. I mean, it was intimidating. But it had some really neat groupware features, well before I think that term was even coined. Anyway, today I'm using something called ThoughtManager Desktop. ThoughtManager began on the Palm. It was a Palm app. And I do use it there, although I find that my Palm is much more useful for reference, for like reading from, than trying to type into. And of course that's the advantage, the Palm is mostly like if you have your contacts, your contact book there, you just give it a few keystrokes, and it finds what you're looking for. So it's used more as a reference device. But they did ThoughtManager Desktop in order to create a Windows-based companion, much like the Palm Desktop, where you're able to synchronize your contacts and the data in your Palm. And this works. Although ThoughtManager Desktop is so good, I just use it standalone.

And every maybe year or so I go out, and I look around at all the outliners that are available. And in fact outlining is so popular there are sites that are just about outlining. And I always find the same site when I Google "Windows-based outlining" or "outlining" or something. And there are different ways for outliners to work. Some of them have two panes; some of them have three panes. Anyway, what ends up happening is I always am reminded how glad I am about ThoughtManager Desktop. So I would encourage people who are interested in outlining, I mean, I'm literally sitting here, Leo, looking at an outline for Security Now! 126. And I have outlines going all the way back to #1.

Leo: You're kidding. Wow.

Steve: No, I mean, and I have an outline on antispam email, article ideas, Bam-Bam work, I mean, literally I have 43 outlines that I'm - this is the way I run my life.

Leo: That sounds like you use it as a List Manager more than anything else.

Steve: I guess I do, except that I've got indentation. And so like I've got an Errata line, and then Weekly Security Update, and then Q&A. And then underneath those are the sub-subjects. And underneath them are little things I don't want to forget to mention. And I'll tell you, for example, when I'm brainstorming a new product, I think I've talked about Crypto Link is the

next thing I'm going to do, a cryptographic communications product. I've got outlines, I mean, everything about Crypto Link is in a ThoughtManager outline because, as I think of things, I just put it in. And I just want to know that I'm not going to forget it. It'll be there. I've captured it.

And then I literally use, well, the thing that's cool about computer outlining is you can move things around. You use the ALT key and the arrows to, like, grab whole chunks and all their subsidiaries and, like, move them around to somewhere else. So you're able to take, you know, just random brainstorming stuff. And as you begin to see, oh, wait, this is like this, and that's kind of like that, we'll put them together, and we'll give them a heading. And so, I mean, I just - I'm a - oh, and of course the ability to open and collapse the levels of outline that are being shown. So you're able to easily see an overview and then go, okay, now, what's under this? And so you click it to open it, very much like a - sort of like a Windows folder tree. Anyway, it's just a spectacularly useful way, or something useful for a PC to do, you know, much like word processing and database and spreadsheets. Outlining is an application that's always been near the top of my list of what I'm glad someone figured out they could do on a computer.

Leo: Wow. I'm so impressed. I thought I knew everything about you. I haven't seen these little lists that you have all around you.

Steve: Next time we're together in Vancouver I'll show you. I mean, I carry them with me. And in fact, that's what I'm using through Jungle Disk because - so I've got my whole outline folder up on Amazon, accessible from any machine where I happen to be, using Jungle Disk to make the connection to Amazon in the sky.

Leo: Now, that's a good idea. So you always have it available.

Steve: Right.

Leo: Well, Steve, we have done 12. We've wrapped them up. It's been a long episode. We had a lot of catching up to do, some big security news. And I think you've earned yourself a break for the week.

Steve: Well, we'll be back with another episode...

Leo: On time.

Steve: ...next Thursday.

Leo: Next Thursday.

Steve: On time.

Leo: Although we have a little, you know, Macworld Expo, we're up against the clock on that one. So we'll have to figure out how to do that. I might have to do that earlier in the morning with you. We'll figure it out.

Steve: We always figure it out, Leo. I'm glad you're feeling better. And I told Elaine not to worry about the timing of the transcript. I imagine we'll put this up as soon as you...

Leo: I'll put it up right away.

Steve: As you're able to do it. And then so the transcripts will follow, you know, whenever Elaine is able to catch up with us.

Leo: That's where you go for the transcripts, GRC.com. That's Steve's site. GRC.com/securitynow has show notes. As he said, he'll have links there to all the stuff we've mentioned. I also will have the links in the RSS feed for the show and on the website. And you can also go to GRC.com for 16KB versions, if you want the small little versions of the file. And the transcripts, when Elaine gets around to this one she'll have that one up. But all 125 episodes are up there already. And of course go there for GRC's fabulous - Steve's fabulous SpinRite, the world's best hard drive maintenance and recovery utility. SpinRite at GRC.com. Steve, have a great week.

Steve: Thank you so much, Leo. It's great to talk to you again. Welcome back from your vacation. And we'll kick off next week's episode, as we're going to from now on, with any important security events since we last talked to our listeners.

Leo: Excellent news. All right, Steve, take care.