SECURITY NOW!

Transcript of Episode #298

Listener Feedback #116

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

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-298.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-298-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode 298, recorded April 27, 2011: Your questions, Steve's answers, #116.

It's time for Security Now!, the show that covers your security and privacy online. Boy, there couldn't be a better time to do a show like that. Of course we started this show more than five years ago. Steve Gibson has been the host ever since, from GRC.com. We thought five years ago he might run out of stuff to talk about.

Steve Gibson: Well, when you first proposed it, I was thinking, uh, boy, I wonder, you know, what are we going to talk about? Will we have enough? And here we are, Episode 298. And, yeah, we've got a ton of information this week, really interesting stuff. There was a breach at the Oak Ridge National Lab.

Leo: Oh, boy.

Steve: Another spear phishing attack there. And of course big in the news was 77 million users of the Sony PlayStation Network had all of their personal information lost. So, funny that I have a new Attacks & Breaches section in our podcast. I think that's going to be a busy section. And all kinds of other stuff. So and of course we've got a great Q&A episode this week.

Leo: That's the thing. I think if you invest, if you buy futures in insecurity, you're going to be all right. It ain't going away.

Steve: God. And it just seems that we see over and over and over where it takes a catastrophe for these companies to get a clue. I mean, clearly, based on what Sony has said, and we'll be describing this, they're cautioning people that, if you use the same email address and password anywhere else as you did or do at Sony, you need to change that, too.

Leo: Oh, dear.

Steve: Yes. Which, I mean, all follows the best practices that we've been talking about for password management.

Leo: I've got a problem, because I signed up for PSN probably three or four years ago. I have - I don't remember what my password is, and I can't log into it, obviously, to figure it out.

Steve: Oh, just ask the hackers what your password is, Leo. They have it.

Leo: And I don't know what credit cards are on file with them or anything. I mean, I haven't used it in years. And yet presumably I'm compromised. Holy cow.

Steve: Well, they said everyone, their entire network.

Leo: Well. And you can tell it's bad because they said we're not going to put it back up till we fix it.

Steve: Well, and they didn't even explain what happened for six days. It just went off. It was shut down. And people were saying, hey, what happened to PlayStation Network?

Leo: Finally they admitted that it was a breach, and we can't fix it, and we're going to start from scratch, they say. So I guess, I take it this means that the passwords weren't hashed.

Steve: Precisely what I was going to say, is that not only - the passwords were not hashed. So the bad guys got away with basically unencrypted data. Nor were credit card information, expiration date, billing address, I mean, everything you gave Sony to sign up, they have acknowledged probably got loose.

Leo: Jiminy. That's terrible.

Steve: It's as bad as it gets.

Leo: That's terrible.

Steve: 77 million. So clearly they were running this huge network with really very minimal concern for security. It's like, oh, well, nothing bad has happened so far, so nothing ever will. Anyway, it'll be interesting if we do learn more. Right now they're being very mum about, like, what it is that happened, how it happened, who got in. They said that they've hired a well-known, highly respected security firm to come in and do a forensic analysis and tell them what happened. So I don't know. You've got to wonder who built this network that they aren't monitoring and managing the security themselves. But anyway...

Leo: Unbelievable.

Steve: Yeah.

Leo: All right. Let's talk about Oak Ridge National Laboratories. What do they do at Oak Ridge?

Steve: They're a department of the U.S. Department of Energy. And...

Leo: The DoE is, by the way, a very highly security-conscious enterprise. They have a - their DoE web page has all these recommendations for security. For end-users.

Steve: Well, and here's the problem, is a phishing attack, a spear phishing attack, is extremely effective. I'm going to quote from what the SANS Institute wrote because they had a really nice summary of this. They said, "The U.S. Department of Energy (DoE) Oak Ridge National Laboratory in Tennessee has shut down email systems and employee Internet access following the discovery of a cyber attack last week. The attack, which some have called" - here we go again - "an Advanced Persistent Threat (APT)" - we saw that with RSA. What that meant was somebody opened their email. It's a little less impressive when you look at the man behind the curtain. It says, "...appears to have targeted Oak Ridge and several other national laboratories in the U.S."

Leo: Oh, man.

Steve: Yeah.

Leo: Is this our version of Stuxnet?

Steve: "The protective measures were taken after an investigation indicated that the attackers were trying to steal technical data." You know, like, what, plans for nuclear weapons or something. That's the kind of stuff we have there.

Leo: Yeah. This would be the most secure place in the world.

Steve: "Investigators believe that they stole less than 1GB" - oh, good - "of data before the attack was thwarted. The attack gained its initial foothold on the laboratory system through spear phishing messages that appeared to come from the HR department regarding employee benefit changes." So somebody opened their email. "When the recipients clicked on the provided link, malware was downloaded to their systems. More than 10 percent of the employees who received the message said they clicked on the link." At least we have honest employees at the DoE.

Leo: Well, 10 percent honest. It could have been 50 percent.

Steve: That's a very good point. 10 percent honest. "Just two of those machines became infected with malware that lay dormant for a week..."

Leo: That's all they need.

Steve: "...before it started harvesting and sending data to a remote server. Lab deputy director Thomas Zacharia says that 'one of [the] core competencies at the lab is cyber security research.'"

Leo: It is. If you go to the DoE page, this is a big thing for them.

Steve: Yeah. Yeah. So Eugene Schultz, who's an editor for SANS Institute, he wrote something in reply to this that I really appreciated. He said, "Spear phishing attacks such as the one against ORNL invariably succeed. Users are getting training concerning how to resist such attacks, but the training is not sufficient - it goes in one ear and [out the other]. More radical (and possibly somewhat potentially traumatic) training, such as inoculation training in which users are sent simulated messages and malware in training labs and loud noises go off if they open one of these messages, is needed." Now...

Leo: Wait a minute.

Steve: I think that's a bit extreme.

Leo: Loud noises?

Steve: Okay. But think about it. One of the problems is, I mean, in a way I think Gene makes an extremely good point, which is there isn't any daily encounter that people have in the workplace with their employers testing them to make sure they don't click the link.

Leo: Right. We should do that.

Steve: I mean, that's brilliant, really. I mean, think about it. Large companies ought to - the security departments of large companies ought to be proactive in deliberately sending email which is spear phishing, that is, they know the employees, they know what departments they're in, they know what they're interested in. See if, I mean, send them baiting email tied to an EXE or a script or something which runs that notifies the employee and headquarters that, whoops, a link got clicked on. And obviously it's not malicious. But, I mean, it's a test. And if you even - well, so first of all, that would spread through a company like wildfire, the news that that was being done, and put everyone on guard. And people who did click the links would have that experience. We don't need loud noises going off, but they'd be like, oh, my goodness, that's what they've been talking about.

Leo: I do kind of like the loud noises, though. Here's an info graphic - this is from a website called KindSight.net - on the process of infection. And so what's interesting is that it lay dormant for two weeks. It's also interesting that only two people got bit. Was that because you think they had antivirus software?

Steve: Or it could have been various patch levels. It's that sort of thing.

Leo: Ah, yes, of course, yes.

Steve: People clicked on it, but some people had - their machines were current. And other people were more like me, oh, I'll reboot soon. And they hadn't yet. So, yeah. But, I mean, so here it is. The problem is we've got - and again, how is it that technical data of a protected nature can be accessed by some computer that's receiving email from the Internet? I mean, just saying that, I mean, it's like RSA's problem where somehow the most critical, crucial data they had was available to, I think we heard it was a secretary who clicked on this email.

Leo: The rich irony of it, I mean, this is DoE's website, their national security page, cybersecurity protection, managing operations security, preventing the spread of weapons of mass - these guys specialize in teaching people how to secure their businesses.

Steve: Yeah.

Leo: This is what they do. I mean, this is as bad as RSA.

Steve: So, I mean, the only thing I could imagine, when you think, when you put yourself in the position of someone clicking the link, they received an email that bore - probably had Oak Ridge National Laboratory letterhead stationery logo on it, looked absolutely legitimate, from human resources. And they clicked on a link. So that says that the only way to prevent this is for no one to ever click on a link in email.

Leo: Thank you.

Steve: You just can't.

Leo: I've been saying this so many...

Steve: You can't trust it. It can come from your mom, or it can come from yourself or something. I mean, there's just no - there's no way to trust the contents of email because it's all able to come in from the outside.

Leo: We've been saying this for so long.

Steve: I know. Yeah.

Leo: It's really scary, frankly. Because this is not just trivial. This isn't a trivial website here.

Steve: No, no. So breach number two...

Leo: Oh, there's more.

Steve: ...of our new Attacks & Breaches section...

Leo: Soon to be the longest section in the show.

Steve: ...is Sony's screw-up, as we were talking at the top of the show. Basically they are saying that they had a network, a server, a database, something somewhere in which the detailed personal information of 77 million users, that is, all the people who were present in the Sony PlayStation Network, had a compromise. They posted an FAQ, a Frequently Asked Questions. And I'm going to quote from two questions from that that are most salient here.

Question #6, they're asking themselves, "Does that mean all users' information was compromised? Tell us more details of what personal information leaked." And the response is: "In terms of possibility, yes. We believe that an unauthorized person" - this is Sony speaking - "an unauthorized person has obtained the following information that you provided: name, address (city, state/province, zip or postal code), country, email address, birth date, PlayStation Network password, login, password security answers, and handle/PSN online ID. It is also possible that your profile data may have been obtained, including purchase history and billing address (city, state/province, zip or postal code). If you have authorized a sub-account for your dependent, the same data with respect to your dependent may have been obtained. If you have provided your credit card data through PlayStation Network or" - what is this, Qriocity, I guess, is their audio.

Leo: Qriocity. Yeah, that's, yeah, it's like a podcast.

Steve: It's related. "So it is possible" - continuing from Sony - "that your credit card number and expiration date may also have been obtained." And then Question #9 is: "I want to know if my account has been affected."

Leo: Well, don't log in.

Steve: Yeah. Sony says, "To protect against possible identity theft or other financial loss" - love that opening - "we encourage you to remain vigilant..."

Leo: [Laughing] Unlike us, apparently.

Steve: Yeah, we haven't, but we want you to.

Leo: So now you have to.

Steve: "...to review your account statements and to monitor your credit reports. Additionally, if you use the same username or password for your PlayStation Network or Qriocity service account for other unrelated services or accounts" - elsewhere on the Internet, they mean - "we strongly recommend that you change them. When the PlayStation Network and Qriocity services are back on line, we also strongly recommend that you log on to change your

password" at that time.

Leo: We certainly do. Another fine mess you've got us into.

Steve: And at the bottom was "Get your free credit reports here."

Leo: What?

Steve: They provided the links and phone numbers to the three credit reporting agencies. They're not taking responsibility, they're not paying for them, but they did mention that, as a - and they cited some law which said that, you know, required that these agencies provide free credit reports at least once a year. So Sony said, if you're concerned, or in order to manage any fallout from this, you can, and we advise you to, pull all your credit reports and see if anyone has been making inquiries, trying to open accounts using this personal information. Because they're talking about identity theft, of course. And this is the mother lode of identity theft.

Leo: This is stunning. I mean, I don't know how it could get any worse.

Steve: I know. And nothing encrypted, just here it is, folks. It's probably in a big SQL database, 77 million people. So...

Leo: Boy, I wish we had those one-time-use credit cards still. That would have been a perfect solution for this.

Steve: Yes, yes. I mean, the good news is this gets a lot of press. I mean, these things are generating press. We talked about the Apple tracking, and we're going to come back to that here in a second. But letters were written from Congress to Steve Jobs asking what exactly it is that Apple is doing. And Sony has not stated whether they were storing their data in compliance with the formal regulations that the credit card industry has set up for the way this data is stored, one of them being it needs to be encrypted. So it's hard to imagine that it was. And again, I'll keep an eye on this and share any other information we learn about how this was perpetrated, how long the people were rummaging around in there. But, I mean, I think we're probably, today, at the worst point we're going to be.

Leo: I hope so.

Steve: My sense is, yes, it can't - because it can't get much worse. And we're all becoming extremely dependent upon our connectivity. Here you and I before we began recording, we're talking about cloud services and stuff in the cloud and all of that. So my feeling is these high-profile breaches have to have other CEOs, COOs, CFOs saying to their security personnel, tell me how this cannot happen to us. Is all of this data encrypted? Those questions have clearly not been asked until now, certainly not within Sony. How is it that a secretary can open email and have access to highly confidential secret technical information?

There's architectural problems which underlie breaches of this kind. It shouldn't be possible for this to happen. So I think in time the architecture will change so that it's not. Or so that it's radically more difficult. This is just shooting fish in a barrel, apparently, because, again, this is now a weekly topic, major breaches that are occurring with clearly increasing frequency.

Leo: I just have to think that - boy. I hope it gets better. But it's something that is so well understood already that I just have to think that companies are kind of accepting it. They - just like, well, this is part of the price of doing business. Somebody in the chatroom said, and I thought this was good, I wonder what the ratio of the amount of money Sony spent on lawyers in the last week is to the amount of money Sony spent on security experts to fix the problem. And he says, I suspect they spent a lot more on lawyers.

Steve: A very good point. And you can imagine lawsuits will be flying.

Leo: And I think that Sony and companies like this, and we know this about the banks, they just batten down the hatches, and they just say, well, it's the price of doing business. Rather than have a real security policy.

Steve: And point at everybody else. Well, look, everybody else is having problems.

Leo: I hope it doesn't - my real theory is that it has a chilling effect on people using the Internet and eCommerce and everything else. I mean, people are really going to think, normal people, not us, but normal people are going to really - well, actually, we will, too, think twice about giving people credit cards, things like that. Citibank apparently still does those one-time-use credit card numbers.

Steve: Yeah. I mean, it's almost worth getting an account just to have that.

Leo: Just to do that.

Steve: Yeah, it really is.

Leo: I want the one where you use it once, but it can be reused, as long as it's always the same company.

Steve: Yes, exactly.

Leo: So it would be Sony's number, and only Sony could use it.

Steve: Yes.

Leo: I mean, I don't want to have to give a new credit card number every time I want to buy something, that's okay. All right. Thank you. So, cheered me up.

Steve: So Apple responds to this whole issue that we discussed that was breaking news the morning that we recorded last week's podcast. And I got a kick out of some of their responses. Number 6 in their own Q&A was the question they posed to themselves: "People have identified up to a year's worth of location data..."

Leo: More than that. Not just up to a year. Many years.

Steve: "...being stored on the iPhone. Why does my iPhone need so much data in order to assist it in finding my location today?" And Apple's answer to their own question: "This data is not the iPhone's location data - it is a subset (cache) of the crowd-sourced Wi-Fi hotspot and cell tower database which is downloaded from Apple into the iPhone to assist the iPhone in rapidly and accurately calculating location."

Leo: Oh, that's interesting.

Steve: I don't think it's true, unfortunately, Leo.

Leo: No, because I'll tell you what...

Steve: I know, my own data was me. It wasn't a bunch of other people.

Leo: It wasn't crowd sourced. I mean, it's downloading it I guess from where you are, based on where you are? I don't know.

Steve: So they're saying, "The reason the iPhone stores so much data is a bug we uncovered." It's like the bars; right? Remember the whole nonsense with the bars showing a strong signal when you were barely getting any? Oh, that was a bug, and we're going to fix that. So "The reason the iPhone stores so much data is a bug we uncovered and plan to fix shortly (see Software Update section below). We don't think the iPhone needs to store more than seven days of this data."

Leo: That seems true, yeah.

Steve: Okay. Now, that was Question #6. Question #7: "When I turn off Location Services, why does my iPhone sometimes continue updating its Wi-Fi and cell tower data from Apple's crowd-sourced database?" Answer: "It shouldn't. This is a bug, which we plan to fix shortly (see Software Update section below)." So they're now claiming that this was a bug.

Leo: That I believe. And I also believe that it was a Skyhook-like attempt to collect data about WiFi access spot locations. You know how Skyhook works.

Steve: Yes, which really does make sense. It makes sense that they would turn their users, thus the term "crowd-sourcing," into mobile probes, which would be associating WiFi hotspots to cell tower databases.

Leo: Exactly.

Steve: I mean, cell tower locations.

Leo: They use Skyhook currently, but it's expensive, and they want to just create their own. Just as Google has been creating their own with the street view cars.

Steve: Driving around all over the place, exactly.

Leo: That makes perfect sense. I believe that it was a bug because I don't see a lot of advantage to them saving it. And to be honest, I feel like this is not the biggest privacy issue ever. They should have absolutely let people know. But, I mean, if you carry around a cell phone with a GPS turned on, don't you think a lot of people know where you are at all times?

Steve: Oh, yeah. In fact, there was someone in Congress...

Leo: Ed Markey of Massachusetts asked Apple for clarification.

Steve: Right. There was that. But there was a gal who sued her cell provider for her own GPS data and then posted the results, to demonstrate what it was that cell companies had. And we know, for example, I mean, even if you didn't have GPS, our phone is logged into the...

Leo: They know.

Steve: ...nearest cell tower.

Leo: They triangulate cell towers. They know exactly where you are all the time. And they sell that information to law enforcement. They have portals. You don't even need to sue. This is apparent.

Steve: Right.

Leo: So I think it's, I mean, it's good, I mean, certainly people now know this and are knowledgeable about this. But it shouldn't be a surprise. I don't think - I think this is no more nefarious than Google's acquisition, accidental acquisition of unencrypted WiFi access spot data, and I would treat it the same way. I think it's - however, in future, please let us know that you're doing this. I think Apple says, well, you do, you agree to this when you - by the way, this is another little disingenuous piece from Apple. When you first connect your iPhone, iPad, or iPod Touch, you get a box that says would you be willing to give anonymous data, return anonymous data to Apple to help improve our services, yes or no? They say, well, that's the box. That's when you opted in. And if you don't want that, well, just either don't check that box or turn off location services. You've opted in.

Steve: And I guess there is a question of proactivity. For example, if after a week or maybe a month, when you next synchronized your phone, iTunes popped up and showed you a map of where...

Leo: That's what should happen.

Steve: Yeah. Then they would say, hi, just wanted to make sure that you understood that you've given us permission to send all this data back to Apple. People would go, what? Because we click on Yes now. We've been trained because no one can read the fine print in this stuff. And also they really weren't very clear about what it was that they were sharing. And they always say, oh, it's anonymous, and blah blah blah, which it is, but it's not anonymous apparently when it's on your own computer. I mean, when I looked at my computer, I was seeing what was clearly where I had been. That is, the data on my machine. So some of this doesn't really track with my own personal experience, which I checked on when this first - the story first surfaced last week.

Leo: It doesn't pass the smell test. And they took so long to respond to this, also.

Steve: Yeah, they really did. And so in their software update section they've said, sometime in the next few weeks Apple will release a free - oh, free, that's nice - free iOS software update that: One, reduces the size of the crowd-sourced WiFi hotspot and cell tower database cached on the iPhone; two, ceases backing up this cache; and, three, deletes this cache entirely when Location Services is turned off.

Leo: Good.

Steve: So that's great. That means for people concerned, you don't need to go digging around in your Mac looking for the consolidated.db file. You just turn Location Services off, dock your phone, synchronize, and iTunes, after this free iOS update, will dutifully delete the cache for you. So that's good.

Leo: Okay. I try to keep this blood pressure down here. You've not done much to help. What else?

Steve: In an interesting story, an interview in Politico, Jon Leibowitz, who is the Federal Trade Commission, the FTC chairman, singled out Google for not adopting the do-not-track header. Computerworld reported that "Federal Trade Commission Jon Leibowitz this week singled out Google for not adopting 'Do Not Track,' the privacy feature that lets consumers," as we know, "opt out of online tracking by websites and advertisers." And in their story they said, "Noting that Do Not Track had gained momentum, Leibowitz said, 'Apple just announced they're going to put it in their Safari browser. So that gives you Apple, Microsoft, and Mozilla. Really the only holdout - the only company that hasn't evolved as much as we would like on this - is Google.'" So I just see this as good news because it is so trivial, so simple to add this to any browser that Google just needs to.

Leo: I think they will. But of course remember that their business...

Steve: Uh-huh. They bought DoubleClick.net.

Leo: Yeah. I mean, this is their business is tracking you, frankly. They own one of the worst offenders here, DoubleClick. So, I mean, but I think that, nevertheless, it's going to be pretty obvious that they have to do this. There's no way around it.

Steve: Yeah. No browser will have it defaulted on, so everyone will have to be - have to turn it on. But it'll happen. And by the way, I forgot to mention that I have had for years, there's a page on GRC somewhere, actually it's on the ShieldsUP! menu of things you can do, which lets you look at your browser headers. And it's trivial to go look there and see if the DNT: 1 is actually being sent, because GRC will show you. So that's something that I had meant to mention. Actually I have some other tech that I haven't put online yet - actually it is online, but it's not enabled by default - where I will be showing you up in GRC's menu header on the top of every page some of these privacy settings and alerting people if they're not taking advantage of them. So, I mean, I'm going to get proactive with that.

Leo: Good. That's great.

Steve: Yeah. So there was another interesting story that I got, I learned about through people tweeting me. And I'll say again, thank you for everyone. And believe me, I got deluged with PlayStation Network tweets.

Leo: Oh, I bet.

Steve: And this is really interesting. This is something that popped up on Nmap.org, the famous makers of the TCP scanner. And that is something called a "split handshake attack" for TCP. What was discovered was that it was possible to bypass the operation of many intrusion detection systems by something called a "split handshake." I'll just read from the abstract of the PDF which is posted:

"Many network engineers might presume that the TCP three-way handshake is the one, inviolate method of establishing TCP connections. A smaller percentage of engineers are also familiar with the little-used 'simultaneous-open' connection method of establishing TCP connections." I would be among them. "Researchers have discovered a third means to initiate TCP sessions, dubbed the 'split-handshake' method, which blends features of both the three-way handshake and the simultaneous-open connection. Popular TCP/IP networking stacks respect this novel handshaking method, including Microsoft, Apple, and Linux stacks, with no modification.

"Given the novelty of the split-handshake technique, session-aware devices have had very little formal testing to determine their effectiveness in relation to sessions established in this way. The authors audit a number of intrusion detection devices, NAT gateways, port scanners, and firewalls, and unexpected behavior was observed within each class of device and application. This inconsistent behavior leads to the conclusion that such network-aware devices and applications should undergo more rigorous testing by their respective manufacturers in an effort to reliably detect malicious traffic, handle network address translation more effectively, and detect the presence of servers offering this form of session establishment."

Okay, so what does this all mean? We've talked in the past, back in the dim history of the podcast, and will again when we refresh our series on how the Internet works, about how TCP functions. We all know that a client that wants to open a connection to a remote server sends a so-called TCP SYN packet to the server, which is short for "synchronize." In that packet the client provides a not-quite-random but definitely unpredictable sequence number. Now, contemporary systems have very good unpredictability of sequence numbers. But historically that was something that was not done well, and early attacks which are no longer effective took advantage of the fact that sequence numbers could be predicted.

So the client sends a SYN packet, a TCP SYN to the server, saying I want to establish a two-way TCP connection with you. The server typically sends a SYN ACK back, which is acknowledging the receipt of the client's SYN and also sending its own SYN, that is, its own synchronization, with a 32-bit initial sequence number in that packet. So that's used to establish its numbering of its packets. And then the client finally sends and acknowledges the receipt of the SYN portion of the SYN ACK by sending an ACK back to the server, thus the three-way handshake - the SYN, a SYN ACK, and an ACK.

Now, the geniuses, the original geniuses that invented all this stuff realized there was the possibility of two machines on the Internet wanting to simultaneously establish a conversation with each other. That is, it was possible that SYN packets might cross paths on the Internet, two endpoints, each sending a SYN to the other at the same time. So the original TCP specification handles that gracefully. It's called a "simultaneous open," where two machines send a SYN to each other, and then the other machine receives each other's SYN, and then they send back acknowledgements of each other's SYN packet and have established a connection. So actually that's four packets transiting rather than just three, but it's called a "simultaneous open."

And in fact it is a trick, it's one of the main tricks used for penetrating NAT because in order to connect two clients that are both behind NAT routers, as we know, NAT routers do not allow incoming unsolicited packets. So the trick is, you get each end, each of the end points that you want to connect through where they're each protected by a NAT router, you get them each to send outbound packets at the same time. And what that does is the packets going out through the NAT routers conditions the NAT routers to accept return traffic from the proper IP and port. So the packets cross over through the Internet and enter the NAT routers and allow you to establish a TCP connection. So it's something which is well understood by people who really do understand the way TCP works.

What these guys discovered is that there is a third way, not the normal three-way handshake and not the simultaneous open, but a third way that also works, which is, and this is kind of tricky, the client sends its SYN to the remote server. The server essentially ignores the SYN data, that is the client's sequence data, and does not send a SYN ACK. Instead, the remote server sends a SYN, just a SYN, as if it was opening a connection to the client. Because the client has got a TCP connection in the process of coming up, it accepts the SYN, even though it wasn't accompanied with the ACK, which the normal three-way handshake would. In response to that, it says, oh, hmm, well, I got a SYN, but I didn't get the SYN ACK. So it treats that as a dropped packet, and it resends a SYN, but it's also acknowledging the receipt of the server's SYN. So what it sends is a SYN ACK, exactly as if a three-way handshake was going on, but initiated by the server rather than by the client. And when the server receives a client's SYN ACK, it then acknowledges the receipt of the client's SYN, and the connection is open. But it in every way looks like the server initiated the connection, not the client.

And what was discovered was that there is, there was network protection equipment which doesn't handle this correctly. The IDS systems that are doing intrusion detection, they're set up to monitor and handle outgoing TCP connections. When one comes back in, it's treated differently, and it turns out it allows bad guys to get around some of the protections and networks. So that was just - that ran across my radar, and I thought that was extremely cool.

Leo: The split handshake.

Steve: The split handshake connection. I've got a person who tweets from time to time, who's also a blogger and a listener of the podcast, whose name is Andrew. He has a site called AndrewTechHelp.com, and he wrote a nice article after we covered and announced essentially the introduction to Microsoft's Security Scanner, the so-called MSS, last week and sent me a little note saying, hey, Steve, I put together a nice article sort of explaining it if you want to share it. So I just thought I'd tell our listeners, if anyone is curious for more information, he did a nice little write-up about why it's not an antivirus, how it differs from that. And I was impressed with what he wrote, so I wanted to share that at AndrewTechHelp.com. If you go there you just find the article about Microsoft's Security Scanner. Currently it's right on his home page there.

Leo: Great.

Steve: And then in TWiC - This Week in Clever, Leo - we have a new form of steganography. Steganography is something we've actually never covered before because it's never impressed me very much. I mean, it's the trick of hiding information like, well, sort of in plain sight. For example, there's a - Wikipedia has a nice article where they show you a picture of a forest or something which actually contains a hidden lower-resolution picture of a kitten with a ball of yarn or something.

And the idea is, for example, if you have a 24-bit color photo, then you've got eight bits for R, G, and B. But the fact is, our eyes don't really need all eight bits. We wouldn't notice if the least significant bit in the various color bytes was off or on. We wouldn't detect the difference. So, for example, you could hide a black-and-white photo or a lower-resolution photo or actually digital information, it doesn't have to be a photo you're storing in a photo. You could hide a file inside a photograph by encoding the files bits in the least significant bits of the photograph.

Leo: So it'd be completely invisible.

Steve: And it works. I mean, it absolutely works great. I guess it never impressed me that much because the bandwidth is inherently limited, that is, the photo is going to be a certain size, so it's X by Y. You've got X times Y, number of pixels. And, for example, you only use the least significant bits of a 24-bit photo, then you get three bits times that. So it's not a lot of data you're able to hide. But clearly, I mean, apparently it has been used by spies. And, for example...

Leo: It solves that problem of having to pass a secret key. So you don't have to put a lot of data in it, could just be the secret key data. And then you could then pass data back in another form; right?

Steve: Well, of course it also solves the plausible deniability issue.

Leo: Ah, yes. It's just a picture.

Steve: I mean, exactly. And you just post it on a website. Or you post it to a photo aggregating site. Or you stick it on your Facebook page. Or, I mean, so it's in plain sight. Somebody else gets it, knows how to strip the noise bits, the least significant bits out, and they've got a message. So anyway, get this. And this is why I think it's - that's why I called it This Week in Clever: Disk drive steganography using deliberate fragmentation...

Leo: Oh, clever.

Steve: ...of files.

Leo: Love it.

Steve: And so think about it. I mean, there is information in file fragments which is completely different from the file contents. So we defrag our drives, which makes the files contiguous and packs them down at the front of the disk for maximum speed. But if you then deliberately fragmented the file system in some clever way, that could contain information itself, the way the files were fragmented, while not altering the contents of the file system at all. So the inventors of this have said that their method would make it possible to encode a 20MB message on a 160GB portable hard drive.

Leo: That's quite a bit.

Steve: So that's a lot of data on a hard drive that would not - it would not be obvious to anyone inspecting the drive. You would have to know that it was there. Of course it's fragile because anything that changed any of those files would immediately destroy the fragmentation. So you could imagine, if you needed to store less than, like dramatically less than 20MB, you could put in, like, redundancy and error correction and things to, like, even be resilient in the face of probably not a full defrag, but some change to the file system, your message could survive that. So the fact that it was 20MB that you could store in a 160GB drive, I thought, well, okay, that's worth talking about. So it's very cool and clever.

And I ran across something, I don't remember now where, about a widget for Windows 7. And I now have a Windows 7 up and running. In fact, we're talking through it at the moment. I built it on this fast machine that I built so that we would have something for both this Skype and also for Vidyo, when we switch to that. Anyway, I got a kick out of it, there's a widget, it's the "End of Support" widget for Windows XP.

Leo: Is that like a clock that counts down?

Steve: It is.

Leo: Oh, my god.

Steve: And I got a kick out of the fact that of course it doesn't run on XP. So I'm not really sure what you use it for because in order to run it you've got to have Windows 7 or probably Vista.

Leo: It should say "This system will self-destruct in...."

Steve: And I have to tell you, Leo, I seriously considered playing hooky for a day and writing a GRC end-of-XP-service-life app.

Leo: For XP.

Steve: For XP, so that it could actually be sitting on people's XP machines. Anyway, the good news is today, when I fired up Windows 7, we have 1,076 days remaining. So I'm not going to start worrying yet about...

Leo: It's Service Pack 3.

Steve: Service Pack 3, yes. Service Pack 2 is lost, stopped being supported. But Service Pack 3, 1076 days remaining. So everybody else who's with me still on XP, and I know you're out there, we don't have to worry yet. We have some more time.

Leo: I wonder if they can say the number the days till next exploit.

Steve: Oh. Actually that'd go negative immediately. And red.

Leo: And very fast.

Steve: And finally, this is my favorite anti-bot thing I've ever seen. I don't know if anyone else has seen this, or Leo, if you have. But I was filling - I was joining - oh, I know what it was. I wanted to post - there was an AES encryption tool whose documentation wasn't very clear. And so I wanted to write the author to ask him something about the way he was handling logon information or, like, password management, and also where he was getting his entropy because that wasn't clear. And that's of course very important. So but I needed to post on this forum. And so I had to join in order to do that. But I thought this was worthwhile. So I filled out a form providing information. And then there was an anti-bot measure, something to prevent bots from joining. And I looked at it, I thought, uh-oh. And it was a question. What year was the Battle of Hastings?

Leo: Now, this must be a British guy because every British schoolchild knows that answer, 1066. But I suspect he's British because I doubt American school kids know this.

Steve: And come to think of it, in 10 days that will also be how long Windows XP has left.

Leo: Coincidence? I think not.

Steve: 1066. I think not.

Leo: Now, is it always the same? It must change.

Steve: That's exactly the question. I was just going to say, I've been tempted to go back and, like, pretend to sign up again to see if...

Leo: But that's better than CAPTCHA, I think.

Steve: It's great. Well, it forced me to go to Wikipedia for it.

Leo: To Wikipedia, yeah, very quickly.

Steve: Yeah, I had to go in, look up the answer. But bots don't know how to do that. So I thought that was just very clever.

Leo: I like it.

Steve: Bravo.

Leo: Only a human.

Steve: Yes, exactly.

Leo: Who's buried in Grant's Tomb, they could have asked.

Steve: Hmm.

Leo: I don't think a computer would know that.

Steve: Yeah, maybe not. We do, though.

Leo: We do.

Steve: So very quickly, a listener, Marek, wrote to say that SpinRite fixed a problem without even running. And I thought, well, I haven't shared that little tidbit with our listeners before. He's in Sweden. And he said, "Hi, Steve. Last week I brought over my computer to a friend of mine because we were having a LAN party. When I got home later that weekend and booted up my computer, it was extremely slow. It was practically impossible to work with it. It would start up in about 10 minutes, 10 times longer than before, and it would get stuck while performing tasks like opening an application.

"After rebooting the computer a few times, I decided to use my copy of SpinRite. While booting into SpinRite, SpinRite immediately recognized that the drive's SMART subsystem for some reason had been turned off. So SpinRite automatically turned it on. That surprised me. So before proceeding to run SpinRite, I tried booting normally. Bang. Everything was back to normal. I didn't need to run SpinRite. The computer booted up just fine and worked as before. Thanks for a great product."

Leo: Sometimes the threat is stronger than the execution.

Steve: Oh, that's a good point. The drive saw SpinRite coming and said, "Aaaggghhh. Okay, I give up. I'll behave." Actually, there's a bunch of stuff SpinRite does because drives can get themselves a little tangled up. There are sticky bits in many drives. And among them is the SMART enable or not-ness. And there's caching bits and error correction bits and things. So one of the things SpinRite does is sort of straighten things out as it's sort of getting ready to go. And that's all that was wrong with his drive. So it wasn't actually a surface problem, it was just something in the sticky bits which SpinRite fixed.

Leo: Now, somebody in the chatroom has given me - Stride (sp) in the chatroom has given me another CAPTCHA that you might like. You'll have to turn around and take a look at your screen. Just to prove you are human, please answer the following math challenge. Calculate...

Steve: Oh, my god [laughter].

Leo: I don't even know what the hell that is. I think I'd have to fire up Mathematica.

Steve: Actually that would be a pretty good joke, you know, to have somebody - the thing that always annoys me is when you fill out a huge long form, and then you get to a CAPTCHA that you cannot read. I mean, I'm definitely human. And there are some that are just so nasty-looking, it's like, oh. And sometimes you can say give me another one, but not always. It would really be funny, like, to have some sort of a deal where you, like, you fill out this really long form, and you get down, and you get, like, one of those things that looks like some graduate study in nuclear physics. And it's like...

Leo: What the heck?

Steve: What's the proper answer? And not multiple choice, because that would be cheating. You've got to fill in the...

Leo: Ah, here's one I could do: Minus 3 minus 2 minus 5. So it says, if you don't know the answer, just refresh. You'll probably get an easier question. And then it's a variety of very, I think in most cases, tricky math questions. Find the least real zero of the polynomial P of X equals X squared plus 6X plus 9. Come on, you remember your algebra 2, come on.

Steve: Okay. We would do a - we do a factorization, and...

Leo: Oh, he's going to do it. Not necessary. What is this page? This must be MIT; right? I mean, who could this be that would expect you to know that? It is the Rudjer Boskovic Institute Quantum Random Bit Generator Service. So there you go.

Steve: Okay.

Leo: If you need a quantum random bit generator. Is that a joke? I don't - somehow it doesn't feel like it is a joke. It's the Center for Information in Computing in Zagreb. I don't even know if it's - what the name for - Zagreb used to be in Yugoslavia. I don't even - I guess it's the - I don't know what country it's in. That's a question. I don't even know what country it's in.

Steve: Yeah.

Leo: Where is Zagreb these days? Is it Croatia? I don't know. All right. Let's take a break. Hungary? No, not Hungary. Where is Zagreb? Croatia, all right. Used to be Yugoslavia. Not Ruritania. We're going to take a break, come back with more. Steve has questions and - actually I have questions. Steve has answers. And none of them are...

Steve: Our listeners have questions. And now it's finally their turn to ask.

Leo: Finally. You can ask some CAPTCHA questions of Steve. Somebody said "Elbonia." I don't think so.

Steve: And we have 12 today because some of them are pretty quick.

Leo: Oh, good. All right. Well, that'll be fun. It's a famous institute in Croatia. Okay. Well, then maybe it's not a joke.

Steve: And it's got random quantum bits, apparently.

Leo: Apparently. That would be a - I guess you'd use that for a seed?

Steve: Oh, you'd def- no, you'd use that for, yes, or actually as a source of random numbers. That's really cool. I've been looking into it, too. In fact, I don't know if I mentioned that that's one of the features in the little YubiKey gizmo is it has a true hardware random number generator using electron tunneling noise, which is one of the good ways to get real, I mean, real quantum-level uncertainty stuff. So, yeah, very cool.

Leo: And they do get some easier ones. Here's 7 minus negative 7 plus negative 6. That's, whew.

Steve: So you just click, you just keep clicking till you get one.

Leo: Keep clicking till you get something that's not algebra 2 or calculus. Trigonometry. I don't even know what it is. Twelve questions good and true, starting with Chuong Pham, who wrote a question to your support email address asking: Thanks for providing ShieldsUP!. That's the port testing service that Steve offers on his website, GRC.com. However, I have one question regarding the user specified custom port probe option. That's the USCPP. Your website shows my port number 58529 as being failed. It's not true stealth. It's open, due to the fact that I've opened this port for uploading data. Well, duh. Well, yeah. So that means it's open. This is the nature of peer-to-peer, and Vuze - he's using something called V-u-z-e - uses this port for both downloading and uploading.

Now, if I disable outgoing traffic in my router for this port, then I can't upload any data. Would it be possible for you to reevaluate the rules regarding P2P ports? Other P2P apps use different ports from Vuze, so I assume they'll also fail, according to your website scan. Interested in feedback. Kind regards, Chuong Pham. Well, Steve.

Steve: Yeah. Now, okay, now I'm wondering why...

Leo: I guess the answer's pretty obvious.

Steve: Yeah.

Leo: An open port is an open port.

Steve: I did reply to him because he was confused. And I said, look, maybe you're expecting ShieldsUP! to be somehow aware of the fact that you deliberately open this port, but that's not something it can do.

Leo: Or should do.

Steve: Exactly. There's 65535 possible ports, and he's using 58529. So which I guess is the standard port for that. One thing that I explained to him, though, was that - because he was asking, like, what vulnerability does this represent. So I said, well, ShieldsUP! is confirming that unsolicited packets, because I deliberately send, the ShieldsUP! system sends its probes from a different IP than the one the user is visiting at GRC so that they are unsolicited and just like they were coming from anywhere else on the Internet.

And so I said that what ShieldsUP! is doing is it's demonstrating that unsolicited packets are able to get into his inner sanctum, essentially, through his router. And I said, it doesn't mean that this is unsafe. But if your - and this is the reason I wanted to bring this up for our listeners is just to remind people about changing, if you can, the default ports. Because the point I made in my reply to him was that, if this is Vuze's standard port, which they tell you to open for peer-to-peer file sharing, then if at some point a vulnerability was found in the Vuze peer-to-peer system, that is, you give it a certain packet or you connect to it and then you in some way do something nonstandard which causes a buffer overflow, I mean, given how hard it is to make these things right, it's almost - it strains credibility to imagine that there isn't somewhere lurking in there some sort of vulnerability.

And the point is that, once that became publicly known, bad guys would immediately scan for port 58529. Just as ShieldsUP! is showing that it's open, they would be able to find all the people using Vuze and then exploit that newly discovered vulnerability in order to do malicious things. So really the lesson here is, if there's any way, I mean, it's like not using the password "password," or changing the name from "administrator" to something else on your system. These are small things, but easy to do, which end up being the reason that some people are compromised and other people aren't.

So if you are able to reconfigure your peer-to-peer client, anytime you've got to have a static port open that is like the common port for a given utility, and ShieldsUP! confirms that it is open, that it sees it open, that's a problem because it does mean that if something were found that was wrong with that, and it was running on a standard port, the bad guys would scan. It does not take that long, as we've been saying, to scan the entire Internet in IPv4 space. That's one of the nice things about v6 is it becomes impossible to scan the entire Internet. But we're not there yet.

Leo: You gave a very long answer. I would have just said, "Dufus, that's the point of ShieldsUP!, to tell you what ports are open."

Steve: Right.

Leo: Period.

Steve: And he had to put that number in, also. Because I'm scanning normally. It was the user-specified custom port probe.

Leo: Yes, it's open. You opened it.

Steve: Right. And I did explain that to him.

Leo: I mean, that's - you're very kind. I mean, and he's not a dufus. I mean, that's a reasonable - I don't know if it's a reasonable question. But this is what port testing is, is is this port open?

Steve: I guess what he was saying was, could you tell ShieldsUP! about Vuze...

Leo: No.

Steve: ...so that - I know. Yeah. No.

Leo: No. In fact, maybe you should think about do I want to open this port.

Steve: Right.

Leo: That's the whole point.

Steve: Right.

Leo: I'm sorry. You're very nice.

Steve: And it's the reason that we brought it up, because our listeners...

Leo: I shouldn't - you're a gentle soul. I shouldn't get upset. That's a reasonable question. But now you understand what's going on. Why would you test a port to see if it's open when, I mean, and you're surprised that it says it's open?

Steve: Then you're going to love this next question, Leo.

Leo: Oh, it's getting better. Fortunately, this one's Anonymous in San Diego wondering why are you still using Windows? Hi, Steve. Love the show. I've been listening for a little over a year now. During that time, until now, I've been able to bite my tongue. But I can't hold back any longer. For the love of all that is holy, why don't you use Linux? I think I asked Steve this in day one. In your last Q&A show you mentioned how you would love to use BSD, and I suppose that by using Mac OS you are using BSD. But why are you still on Windows? I understand if you want to be successful in developing software you must test it on the OS that has the greatest market share. And this is the good question: Why use it for personal use? In the same show you mentioned something about not wanting to be on the command line all the time. Well, as I'm sure you know, there are probably close to a hundred different window managers and desktop environments for Linux/Unix: Gnome, KDE, XFCE, Fluxbox, Openbox, Blackbox, and now Unity. Please try it out. This is, I think, an evangelist.

Steve: Yeah, I think so.

Leo: There are great advances - and I love Linux. I use it all the time. There are great - so don't - there are great advances being made every hour - every hour - in Linux and BSD technology, and it's free. I've been using Linux as my main OS since 2004, and I haven't looked back. I was forced to do something on Windows 7 recently and found it very confusing and frustrating to use. I think it would be great if you started a small segment of the show discussing Linux and Unix desktop security vulnerabilities because of course I know there's no perfect OS. Thanks for everything. I think that's appropriate. Why don't you just use Linux?

Steve: I like Windows.

Leo: Oh.

Steve: I don't like Windows 7. I like XP. Maybe someday I'll like Windows 7.

Leo: But Steve, it's a toy operating system. You said it.

Steve: It is a toy. And, I mean, it really is. No, I mean, I wanted to add this question today because we do get this in our mailbag a lot. And it is - I guess it's a number of things. First of all, anything I want is available for Windows. Not everything I want is available for Mac. On the other hand, not everything that's - there are some things for the Mac that are not available on Windows.

You may remember that I switched to using the Mac for some period of time when I was writing all the code for those machines that are over my left shoulder behind me. All of that was on a PDP-8 simulator that was only available for the Mac. So I dusted off a MacBook Pro and used it happily for some length of time. So, and I've got a BSD server where our newsgroups live, and it's the DNS server for GRC. And every time I touch it, I feel good. It just feels right somehow. And so the idea that Mac's got a real good Unix underneath with a very nice UI on top, to me I think that's probably my sweet spot.

But I know Windows inside and out. I'm a Windows developer. Anything that I want, like my little wacky Wizmo, which a surprising number of people like and use for turning their monitors off and rebooting and doing little utility functions, it's easy for me to whip these things out for Windows, much as I said I was considering doing a countdown for days left in XP's life. So I'm a Windows developer and a Windows user. And I've never been, knock on wood, been bitten by any of these problems that do catch out so many people because I'm a very careful Windows user, and I do not click links in email. So it works for me.

Leo: Well, and that's another answer which you've given in the past, which is how am I to talk about Windows security, how am I to be an expert in Windows security, if I don't use Windows?

Steve: Right.

Leo: So you kind of have to. I mean, it's not merely because you want to sell more copies of SpinRite.

Steve: No.

Leo: In fact, it's not that at all because, I mean, it runs in DOS.

Steve: SpinRite boots itself, yeah, exactly.

Leo: So it has nothing to do with that. It has to do really with the fact that, if you want to talk about security...

Steve: Well, and Leo, if I want to affect the most people. Frankly, my DNS Benchmark is incredibly popular. I looked at the DNS page, I think 1,500 people a day look at that. And about 500 of them are downloaded every single day. Well, sorry, I mean, Linux exists, yes. But Windows is where everyone is. Windows, I mean, and the Mac, the Mac to a growing degree. And when I wrote the benchmark, I did make sure that it ran under Wine for Linux and the Mac in acknowledgement that those platforms are growing in strength. But still, I mean, by default, Windows is - it's ubiquitous. So I want the things that I write to be able to help the most people.

Leo: That's an interesting point because you don't make money on those freebies.

Steve: No.

Leo: And you know how to develop for Windows. You're not a Mac developer or a Linux developer. And that's just what you know how to do.

Steve: And frankly it would be a huge learning curve. I mean, it's not small...

Leo: No, it's not trivial.

Steve: ...to switch platforms.

Leo: For instance, SpinRite, which uses INT 13, has to be on BIOS. You're using a BIOS call. You would have to duplicate all that functionality on EFI. And I guess you could do it in Linux, wouldn't be so hard. But it still doesn't run in Linux, it runs perfectly well on a Linux box in DOS. You just put the - you create the boot disk, you stick it in, it's running on FreeDOS, and it just runs, so that's fine.

Steve: Well, yeah. And you often see multiplatform things that just aren't very good. I mean, for example, they make you install Java because they're written in Java. And it's like, okay, they work. But they're just - they don't feel like they're - it's like, because they want it to run anywhere, they don't really run anywhere very well. And it's like, eh, that's not a tradeoff I want to make. I want to make really, really good stuff for Windows. And increasingly acknowledge that it's not the only solution in town, and put some time into supporting other platforms, as well, as I have.

Leo: I begged him to write, I begged him to write, to rewrite SpinRite to work on the Mac. But no.

Steve: Not quite yet.

Leo: No. But that's fine. Because I just take the drive out, put it on a PC, and run it. It works fine.

Steve: Yeah.

Leo: Friedrich H. Burkardsmaier, who lives in Thailand, just to throw you a curve, wonders about virtual keyboards and form grabbing: Steve, one of your recent episodes you recommended the use of a virtual keyboard to enter passwords so they can't be intercepted by keystroke loggers. My concern is that passwords could still be intercepted by something called a "form grabber," once the virtual keyboard has been used to fill in the form. In other words, you are submitting it, frankly.

Steve: Correct.

Leo: I would appreciate it if you could elaborate on this topic. How are form grabbers implemented? Are there effective countermeasures a user can take? Thanks for the excellent software and for the great podcast. I always look forward to listening to every episode. Friedrich Burkardsmaier.

Steve: So, okay. So what he's saying is that he recognizes that a virtual keyboard, like a keyboard on the screen which you click with the mouse, will avoid a hardware keystroke, and actually hardware and probably software keystroke logger; but that, once you've used that to fill out the form, when you submit the form, there's this possibility that the contents of the form could be grabbed by some malware running in your machine. And he's absolutely right.

As we've said before, the web was originally a content delivery system. So the concept of interacting with web servers interactively, that is, sending data back, posting things, this was all something that was sort of an afterthought. And in fact the design sort of demonstrates that. One of the ways that data is sent back, sort of the most ugly way is you make a request of the server, and you add the data to the end of the URL. So it's whatever, blah blah blah blah, dot html, question mark. The reserve symbol question mark specifies that this is the end of the URL and the beginning of additional data, that is to say, form data, which is then tacked on the end. It has uses because it allows you to, for example, save search queries in shortcuts because a shortcut is just able to save the URL. So there are some advantages to it.

But the problem is anything that sees what you're submitting is able to look at your forms. And the alternative way of submitting data basically just has the data after a space line under the submission headers, you just list all of the data in all of the fields in cleartext, and off it goes. It doesn't get encrypted, of course, until it goes into SSL. So the form contents itself is in the clear.

In thinking about the answer to his question, the only thing I could imagine that would solve this would be scripting, which would run in the browser client, which would intercept the Submission button, and that's easy to do in JavaScript, preventing the normal browser behavior. It would then take the data from the form, encrypt it well, and then submit that. So it really needs to be - it needs to be a service provided by the page containing the form that you're submitting.

And so that would be, since it's a service provided by the page of the service you're submitting the form to, it would be, for example, LastPass, they do this. For example, they've got script running in the client that encrypts the stuff at your end before it ever goes over the wire. And so it's certainly possible that that could be done, although today the number of services that do that are - you could probably count them on one hand. So we're a ways away from having security from that kind of exploit.

Leo: Question 4, Andrew in Tucson, Arizona wonders about IPv6 support for your program, ShieldsUP!. Can you add - I feel for you, Steve. Can you add IPv6 support for ShieldsUP!? Most operating systems lack an easy way to view if an IPv6 firewall is enabled or to easily check what rules are applied. For instance, OS X lacks an accessible IPv6 firewall. But ShieldsUP! is not a firewall, first of all.

Steve: No, but of course it does test yours. And I'm getting an increasing flux of questions about IPv6. I don't know if it's because we're talking about it so much on the podcast, or people are increasingly getting ready to use it, and they want something to verify what their ports are that are open. The good news is I'm moving toward that pretty quickly. I've got hardware on order. I've got IP networks are being added. And it looks to me like it's not going to be a really huge problem.

The architecture that I created - ShieldsUP! is, I think it's in its maybe third iteration. And I really had figured out how to do it by the third try. That was that NanoProbe edition that I came out with a few years back. And it's just - its implementation is so clean that I can add the IPv6 layer to it without much trouble. So I don't have an ETA. I don't work with ETAs, as everyone knows. But it's definitely on my radar. And I like the idea of being out there early and maybe even exclusively with a good port tester for IPv6.

Leo: Cool. Joshua Gardner, San Antonio, Texas, wonders about Memtest86. Oh, I remember Memtest86. That's a blast from the past.

Steve: Yep.

Leo: I was fumbling around on the web, found this nifty open source program, Memtest86.com, for testing RAM. It appears to be quite extensive in what it tests and the patterns tested. I was just curious if you've heard of it, and your thoughts. Finding a way to actually test memory and give it a conclusive good or bad status has been a real challenge.

Steve: And you're right. Memtest86 has been around since, not surprisingly, the 8086, which is where it got its name. It turns out it's tricky to test memory. And as you could imagine that, being a hardware-level guy and the author of SpinRite, at one point I sort of thought, hmm, maybe I ought to get into the memory testing business also. The problem is that, if I found a problem in a bank of RAM, the user would want to know which was the bad SIMM. And I mean, anyway, there's really no way to tell. There's no way to accurately, especially these days where SIMMs are paired, and there are DIMMs, and there are quads, and it's, like, very complex. There's no way to be able to say, oh, on your motherboard it's this one over here. So I just decided not to do my own GRC-style memtest. And there is a good one that already exists, and that is this Memtest86. It's not as simple to test RAM as writing some data in and then reading it back out because - and you'll remember this from the old days, Leo - core memory used to have something called a "checkerboard test."

Leo: Right.

Steve: And it was because it turns out that not all data which you would write into core memory was equally easily read back. And so it actually, that introduced the notion of ways you could read and write that induce the maximum amount of noise in the core memory system and best tested the one and zero discriminators. Turns out that RAM bears a strong relationship to that. That is to say that the way memory, the way dynamic memory tends to fail is that adjacent bits improperly influence each other. So, for example, you want to write a one into a given bit, and then write zeroes into all of its neighbors and see if the writing of zero into its neighbors influences the one that you stored. And then you want to write a zero there and write ones into all the neighbors and do that. And you can imagine, it's extremely difficult and time consuming.

So bottom line is, Joshua and all of our listeners, Memtest86 is a terrific program. It runs standalone. It's bootable. It's free, open source. You can download an ISO you burn to a CD, boot a machine, and just let it crank away. And when I've had some strange, hard-to-diagnose problems, that's what I've used in order to see whether all of my RAM is working. And it's also a great exercise if you're working on, like, tweaking your system and wanting to get the maximum speed out of RAM, this is a good way to see whether, without all the other layers of OS and everything confusing things, if the RAM itself is being pushed too hard, based on the wait states and speed that you've told your fancy BIOS to run.

Leo: There is an updated Memtest86. It's called Memtest86+ at Memtest.org. That's probably where I would go to get it, as opposed to the URL that Joshua mentioned.

Steve: Is it Memtest86.org?

Leo: It's Memtest.org. But it is...

Steve: Okay, good.

Leo: It's basically what's happened is that the guy who wrote Memtest86, Chris Brady, stopped adding, developing it. And so there was a team that took the open source and updated it. Its open source, GPL, and so it's free. And you can see everything that they've done. And it just makes it more up to date.

Steve: Yes, good. It's funny because he put Memtest86.org. And I went there first, and I saw, like, an abandoned pseudo search site.

Leo: He was confused, I think.

Steve: Yes.

Leo: So there is Memtest86.com, but that hasn't been updated since 2002. Memtest.org, no 86, just Memtest.org is the fresh version. Same idea. Same source, basically. Just updated for modern OSes and so forth. Actually probably not OSes, modern hardware, I guess.

Steve: Yes.

Leo: Kai Franke in Germany would love to use Seagate HDD hardware encryption. It's built in, of course. Steve and Leo, I'm a long-time listener of your show, very security concerned. My question, is TrueCrypt's whole drive encryption as secure as Seagate's HDD hardware encryption? Ever since your TrueCrypt episode, I've been using TrueCrypt all the time. And right now I'm using whole drive encryption on my Netbook. But encryption/decryption is of course CPU intensive, needs more energy, reduces my battery life. I didn't even think of that, but I guess it would. Because the Netbook includes a 250GB Seagate hard drive, I looked around on their web page to find a bigger, faster drive with lower energy consumption, and I found a drive that supported hardware encryption. Wouldn't that mean I don't need a CPU-hungry TrueCrypt anymore? Best regards, Kai Franke in Germany.

Steve: Well, the problem is, with just swapping an encrypted hard drive for one that isn't, is that you absolutely have to have BIOS support. And it's in some laptops, but I don't know about a random Netbook. You can check your BIOS. If you can get into the

BIOS of your Netbook, see whether it talks about a hard drive password. And it's not clear, I mean, you'll need to see whether it's just a standard BIOS password or a hard drive password or hard drive encryption. Normally they'll make it clear that it's hardware, there's hard drive encryption, because it requires BIOS support because the first thing that happens when the laptop powers up is it has to provide the encryption password to the hard drive that normally comes from the TPM, from the Trusted Platform Module.

So you normally enable the trusted platform module, and then it contains the password for the hard drive's hardware encryption. But unfortunately it is not as simple as dropping a hardware-encrypted drive in anywhere and just having it work. And it definitely needs to have a hard drive password in addition to probably the trusted platform module. And, for example, on most mainstream motherboards, we still don't have a hard drive password. But laptops typically do.

Leo: Oh, that's interesting. By the way, now they're telling me, oh, no, Memtest86.com has been updated. So try one or the other. It's probably exactly the same.

Steve: Yeah, they're free.

Leo: Yeah. Brent in Central Illinois is confused about the RSA attack we talked about last week: Steve, additional details have now been released. CNET had a story. They said that the RSA attack was due to a phishing email that exploited a Flash vulnerability? Flash displayed in an email? Crazy, huh? Anyway, I'm not really clear exactly how they got infected because they said the email had also attached an infected Excel file. So Flash loaded the infected Excel file, which infected the system? Is that how it worked? Please explain.

Steve: Well, I thought this was important enough just to quickly touch on this and explain that Brent is right in being confused, that it's sort of the other way around. And we see this over and over and over in spear phishing sorts of attacks. The vulnerability is in Flash. That's the culprit. But in order to get Flash to run, because typically email won't run it, it needs to get encapsulated in something else. And typically users need to be coaxed into opening something. So it's generally a, for example, a Microsoft Word document or an Excel spreadsheet.

And that was the case in the RSA attack. It was an Excel spreadsheet which was opened by a link in the email. And embedded in the Excel file was the Flash exploit. So it's the document that contains the Flash exploit which is executed by either Word or Excel, they being the carriers. So don't open those.

Leo: Don't open attachments.

Steve: Don't do it.

Leo: Don't click links in email. Don't open attachments. Knock it off. Steve Holmes in Lake Forest, California found a terrific Bitcoin podcast episode. Here's an FYI, we did a whole show on Bitcoin. He said he listened to the April 4 EconTalk podcast interviewing Gavin Andresen, a principal of the Bitcoin Virtual Currency Project. That podcast is at EconTalk.org. And they talk about Bitcoin, the origins of it, how new currency gets created, how you can acquire bitcoins, prospects for Bitcoin's future, could it eventually replace government-sanctioned currency? How can users trust it, what threatens it, and how it might thrive. Thank you, Steve.

Steve: Yeah. I wanted just to share that with our listeners. There's been an enduring interest in Bitcoin...

Leo: No kidding.

Steve: ...back ever since our podcast. So it's EconTalk.org. You'll have to scroll down. They've got a number of podcasts. It's very nice. It's about an hour-long audio podcast. The date is April 4. And it's a not-so-technical interviewer interviewing Gavin, who is a coder, and technical, and really understands this stuff and does a really nice job. So if anyone wants any additional dip into Bitcoin, I wanted to bring it to our listeners' attention.

Leo: A dip into Bitcoin. Michael Dombrowski in Washington, D.C. wonders about Do Not Track and Google Analytics: You were talking on Episode 295 about DNT, the header that tells a browser - actually that a browser uses to tell a site not to track it. You said the U.S. will most likely eventually make it illegal not to honor a do-not-track request. I hope the tracking is clearly defined, however. As someone who runs a few sites, I need to know how many people visit my site and what posts do the best. I hope that tracking is defined as tracking a user as they surf from site to site, not as they browse a single site. Products by Google Analytics are greatly helpful to me as a site administrator. I use Google Analytics, too, actually, among others. I also use Quantcast on our sites.

I can only hope that the bureaucrats - in fact, if you download this podcast, you go through a little track. I can only hope that the bureaucrats do not make this all about political parties and also think about the implications any law they pass will have. Thank you for all you do for the tech and security communities. And he says, can I pimp my site: UnblockedAlways.com.

Hey, let me ask you. I didn't even think of that. We track, in a sense. The only way we can tell advertisers how many people have listened to this show is, if you look at the URL for the show, it filters you through a site real quickly, the Podtrac site, and that's how we count how many downloads we've had. And they do a lot of stuff. I mean, it's not just like, oh, one, two, three. They compare the IP address that you're coming in from to a database of IP addresses. They make sure it's unique, they make sure it's real. We do a lot of stuff with that. Would we be affected by this?

Steve: It's a really good question. And Michael raises a good point. I hope that we don't see overreach in what the Congress does. I have to imagine there'll be hearings, there'll be people really arguing against this being too pervasive or too onerous. And the clear distinction will be made certainly between first-party and third-party tracking. I think we pretty much understand what it is that we don't want. And so we just need to get legislation that does a good job of encoding that into legal jargon.

Leo: It's kind of scary.

Steve: I know. Let's hope we - yeah. Because, I mean, I don't know whether...

Leo: It is third-party tracking for us. You go to our site, or you go to iTunes, and it goes through the Podtrac site briefly.

Steve: Yeah. And they are getting people's IPs. We know that they don't care, that they're not aggregating them and so forth.

Leo: We do. We don't collect them. We don't even aggregate them. We just compare it to a database and say, oh, this is a real IP address. And I guess we'd have to collect them to see if it's unique. So I guess for each show it does make, you're right, it makes a database of who's downloaded it, and we just compare it to - because otherwise, if you download it 20 times, advertisers don't want us to count that as 20 impressions. It counts as one.

Steve: Well, okay. So I think that one of the problems we're going to have is that just a single DNT header with a single on/off switch is going to be too coarse. What could happen, for example, is that, if this legislation were enacted, and it were decided that, for example, Podtrac was - oh, gee, that's what second syllable in their name?

Leo: T-r-a-c, yeah.

Steve: Yeah. If Podtrac was tracking, then what would happen is, if they received a link, which they would normally redirect, which contained the DNT header, they'd have to instead give you a page and say, we're sorry, you've come to Podtrac. Here is our privacy policy. Here is why you've come. This is what we do and why you have visited us. Please make an exception for your DNT header for us so that we are able to forward you to the material you want. Or that page could just show the direct link that you would be forwarded to, and you could click that instead, if you manually didn't want to be tracked in that given instance. So there are workarounds. But I really do think that there will be instances where people essentially need to or want to give permission for some clearly defined delineated reasonable tracking, essentially.

Leo: You can see why this is a complex issue. I mean...

Steve: Yes.

Leo: It would put us out of business. Guillaume in France wonders about the SSL OCSP protocol. Steve, I'm an engineering student in computer science, a GNU/Linux user, and listener since December 2010. I watched your last Security Now! podcast and was concerned by the SSL authority breach. The OCSP protocol is used to check the status of certificates, and the default behavior is, if OCSP fails to get a certificate status, it is assumed to be valid. So it's kind of upside down. However, this default behavior can be changed, at least in Firefox 4. Go to Option > Advanced > Encryption > Validation. There you can check a box to assume that certificates are invalid if the OCSP status check fails. Thanks for the show. Guillaume.

Steve: So, yeah. So I wanted just to bring this up to our listeners. We didn't talk in detail. OCSP came up briefly when we were talking about SSL revocation a couple weeks ago. OCSP stands for Online Certificate Status Protocol. And the spooky thing about it is that it potentially represents a privacy compromise because the way OCSP operates, it's not a revocation list, which is the alternative approach. Instead, when enabled, your browser will manually, I mean, deliberately connect to a verification server every time you visit a site with a certificate. So the certificate can contain a URL saying here's a URL of our OCSP server if you want to verify. And so you can - and then there's another setting, for example, in Firefox, and this also is in prior to version 4. It's in 4 and also in the 3 Firefox chain. And that allows you to override the certificate's statement and always go to a specific OCSP server, which can sometimes be useful.

And the problem, though, is that essentially what that's doing is, everywhere you go that is secure, your browser is sending a beacon out to whatever OCSP server is specified that inherently contains your IP because it's establishing a connection in order to verify that the certificate is currently still valid. So it's a nice technology inasmuch as it does allow for realtime verification of certificates. The problem is that it also allows for this OCSP server to know that you're going to that site.

The reason it currently fails in the wrong direction, that is, if you, for example, if the OCSP server didn't respond, Firefox and all the other browsers that support this fail open. That is, they just go, okay, well, we don't know that it's bad. So since we haven't heard anything, we're going to assume it's good. That's the behavior that can be inverted, and I think it probably makes sense to invert it. One of the concerns with OCSP is that, if an OCSP server were under a denial of service attack, or had a network outage or whatever, then if browsers failed hard, then suddenly all the certificates that were referring to the OCSP server would refuse to connect. And that would be really bad. So again it's like, well, we want the security, but we don't want too much because that might get in our way and bite us.

Leo: Last question for the Gibson, from Li in Houston.

Steve: Oh, and this is good.

Leo: He mentions a Firefox plug-in called Certificate Patrol: For expert users like us and your listeners, it's a great help. It alerts when SSL certificates change by comparing and displaying the old and new certs side by side. My employer probably won't be able to start proxying my communications without my noticing. Ah.

Steve: So this is exactly what I said I wish we had. And Firefox does have it. I did some poking around looking for one and didn't use the right keywords, I guess. But Certificate

Patrol. I have found it and downloaded it. And I'm impressed with what I've seen. For example, it will even - it's built some intelligence in. So the first time you go to a site to which you have never been, that is to say, when it's going to be caching the certificate for the first time, it stops you and shows you the certificate and says, hi, here's a chance, we're about to put this in the cache, so make sure this looks good to you so you don't initially cache a bad one, and so you don't go to a site through, like, a bogus certificate chain the first time.

Then when I talked about it being a little intelligent, it will notice, for example, if a certificate has been reissued after the prior one expired. So it's smart enough to say, oh, just by the way, the certificate for this site you're visiting has been updated because the prior one expired. So it'll see that the one it had in its cache before had expired, see that it's being now given a new certificate with a different serial number, since every certificate has a different serial number, and lets you know what's going on. I think this is a great solution. And if you did go to a site that had a bogus certificate, or there was a man-in-the-middle attack going on, or your employer was setting up a proxy, or any of those things, this thing would warn you. It's exactly what we want.

Leo: So Google "Firefox Certificate Patrol," and you can find that add-on and install it.

Steve: Yeah. And there was one comment from someone who posted, saying that it actually did work. His employer was adding a web proxy, and because he had Certificate Patrol installed, it alerted him. Hah hah.

Leo: Hah hah.

Steve: All we want is control. That's all we're asking for. We just want to know what's going on.

Leo: Please don't not track us. Don't let us not - I don't want to go out of business.

Steve: It'll be okay, Leo. I'm sure our listeners will make an exception. But we're going to have to have - it's very much like NoScript, where scripting is disabled by default, but then you enable it for sites you trust.

Leo: Yeah. That'll basically kill us because most people won't enable it. It'll just - it'll be dead.

Steve: Well, no, but, I mean, if it's dead, then - I don't know what.

Leo: Well, and that's the whole issue about tracking. And I'm sure it's one of the things Google is thinking, is...

Steve: Oh, I know what it is, Leo. They'll come to TWiT, and you will see if their DNT header is on when they...

Leo: I will say, "No podcast for you."

Steve: Then you can explain, you say, hey, sorry about this, I hope this is not an inconvenience, but we need you to make an exception to DNT for the following URLs in order for the podcast to be available. So it'll be possible to intelligently handle it.

Leo: It'll put us out of business. I guarantee. Because people just, they don't do that. They just go, oh, never mind. Bye. Oh, that's okay. We'll worry, we'll cross that bridge when we come to it. There may be other ways to measure without tracking, and I guess we'll have to find those ways.

Steve: Actually there are. I mean, you could do it yourself. It's a convenience for you that you're bouncing through a third party. But...

Leo: Well, advertisers demand it, of course, because advertisers don't want numbers from us because we could lie.

Steve: Yeah. But you could put some script, you could do a Google Analytic-style thing where you put some script on the page that would - it would still be giving control to a third party, so. There are probably ways to do it. Maybe that would still be illegal, I don't know.

Leo: That's the issue. And I think that that's really something to think about. Those are the unintended consequences of not tracking. If you're an ad-supported network, and people use ad blockers, it ultimately puts you out of business. And it's the same thing. That's why it's a larger conversation. But I'm not going to worry about it. I'm a bullish optimist. Or maybe I'm just...

Steve: I think too much of the Internet is running on, I mean, with benign tracking. So it's benign tracking, nobody minds. It's having personal information aggregated that is the biggest problem.

Leo: Oh, I agree. I agree. But I don't think that these tools and laws make much distinction.

Steve: We'll see.

Leo: Yeah, I'm not going to worry about it. If we go out of business, we go out of business. I can get that boat then. Thank you, Steve. A great show, as always. Now, next week we don't - do you know what we're going to cover?

Steve: Believe me, too much happens in a week, Leo.

Leo: It's hard to say.

Steve: Who knows what's going to happen? But I've got a whole list of things to talk about. So if nothing else comes up to preempt, then I'll pull from my list and we'll have another great podcast.

Leo: You bet. And by the way, if you have questions for Steve, we'll be doing questions and answers again in a couple of shows. Go to GRC.com/feedback. While you're there, pick up a copy of SpinRite. Everybody should have that. If you've got a hard drive, you need SpinRite, the world's finest hard drive maintenance and recovery utility. Also lots of other freebies, including, as we mentioned, ShieldsUP!. But there's also Wizmo. There's toys, there's security, there's everything. It's a great site: GRC.com. You'll find show notes there, transcriptions of each and every show, all 298 episodes. We're going to be Episode 300 in a couple of weeks.

Steve: Yeah, we are.

Leo: And of course Steve does 16KB versions for the bandwidth impaired, as well: GRC.com. Steve, thanks, as always, and I'll see you next week.

Steve: Talk to you then, Leo.

Leo: On Security Now!.