Listener Feedback #25

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-112.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-112-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 112 for October 4, 2007: Your questions, Steve's answers #25.

It's time for Security Now!, everybody's favorite security podcast, now officially the best security podcast of the year...

Steve Gibson: [Trumpeting]

Leo: Thanks to the folks at the Podcast Expo and the - I don't know, who does the voting? Is it...

Steve: Well, thanks to our listeners, Leo.

Leo: Oh, you're right, the voters are you, the listeners.

Steve: Yeah.

Leo: And we only asked once. Which I'm very proud of.

Steve: Very cool. Very cool.

Leo: We did very well considering we only asked you to vote once. So you're officially now the best technology and science podcast in 2007; is that right?

Steve: I think that's the case. And...

Leo: And I think you get some little doohickey and...

Steve: I don't know what I'm going to get.

Leo: You'll get a plaque or something, I'm sure.

Steve: Well, Elaine is going to be there with camera, so I may actually be able to put some photos up on the show notes.

Leo: Oh, good. Elaine is the wonderful woman who transcribes all these Security Nows. Yeah, we have to say that because she's typing even as we speak.

Steve: And she's smiling right now.

Leo: She's the greatest.

Steve: Actually she was laughing so hard, she told me, during the end of the Listener Feedback 24...

Leo: Oh, yeah, the tattoo, yeah.

Steve: ...with the wacky UV tattoo guy that her husband came into her office and said what in the world is so funny?

Leo: Well, her bread and butter is court transcriptions, or court reporter. And so I don't think you get a lot of laughs out of that.

Steve: No, it's pretty dry.

Leo: Probably rare that he hears her giggling. Maybe not, though, who knows. So this is a listener feedback day, as are all our even podcasts these days, Listener Feedback 25. And we've got a good set of questions. No wacky 13th this time.

Steve: No.

Leo: Speaking of nerds, Steve Gibson and I are here. We are ready to nerd out with some of your questions. Do you want to mention SpinRite real quick before we move onto that?

Steve: Oh, I thought I would give it another vacation since we're...

Leo: Steve...

Steve: ...on a bit of a tight schedule, and I don't...

Leo: Steve.

Steve: All right. I don't want to take up...

Leo: Make it simple. Yeah, I understand, but if you've got a hard drive in trouble, SpinRite. GRC.com. It's the best money you'll ever spend. It recovers hard drives. It helps you maintain your hard drive so you don't have these problems. I've got a guy who was ready to spend, I think, a couple thousand dollars on one of those companies where they take your drive apart and replace the platters. And I said, whoa, whoa, whoa, you know, it could just be a software problem. You should try SpinRite first. And I think it was a software problem because the drive would boot up, but it would say "No NT loader found," right?

Steve: Right.

Leo: And that's where SpinRite can really help you. And I haven't heard back from him, but I'm sure that he's been very happy with it. GRC.com.

Steve: It probably is worth mentioning something that's been on my mind. It's not a testimonial, but I should mention that we keep hearing from people who have been brought back from the brink of disaster. Their computers wouldn't boot, they had lost their stuff, whatever. I mean, those are pretty much what the testimonials are every week. But had SpinRite been run on those drives prior to passing off the cliff, then the drives would have never had this problem. I mean, Greg, in fact, my tech support guy, he says, "You know, Steve, you never mention to Security Now! listeners that it is preventative maintenance. It's not just data recovery." And I go, "I know, Greg, but I just don't think people are going to go out and spend $89 because they have extra money."

Leo: Well, they do if it's a crisis. But you notice I always say "the world's best drive maintenance and recovery utility."

Steve: Yeah. So I did want to mention that SpinRite does prevent these kinds of problems. It is able to, way more often than not, bring a drive back into use that has already passed out of use. But, boy, if you run it every couple months, it will never get into that trouble in the first place. I mean, it's really the case.

Leo: Right. All right, let's get to our questions, my friend, starting with No. 1. Numero uno, Keith in New Jersey. He says: How do you do credit card transactions? I just want to know what service allows you to process credit card transactions. He's talking about your eCommerce solution. I understand how you were able to take all the required information, including the credit card. But then what? Does VeriSign offer a service to charge credit cards?

Steve: That's a great question because I remember when I was first researching this I dug around to, like, find the right electronic funds transfer, EFTS, Electronic Funds Transfer Service, or System, that would do that. I ended up settling on a system called CyberCash that was in the game and had a very nice API. Essentially, I downloaded a kit from them and had - in this case for Windows, although they supported multiple platforms. And they had a DLL, a Dynamic Link Library, that I just added to the server.

Leo: Oh, that's nice.

Steve: And then my software would interact with that DLL. That contained SSL technology, certificates, and all the stuff it needed for connecting to their back-end credit card processing system. And as it happens, a couple years ago VeriSign bought CyberCash, and the system is called Payflow Pro. I can't speak for any other facilities because I've never used anything other than Payflow Pro, which used to be CyberCash and is now VeriSign. One of the reasons is we have never had a single problem with this from day one. I mean, it...

Leo: It works.

Steve: Yes. And I was worried when VeriSign bought them. It was like, oh, no, are they going to change the protocols or make me update something or mess with something that's already working perfectly? And they have done apparently nothing, at least nothing bad, to it. I mean, it just - it works beautifully. And so if we have listeners who have an interest in doing eCommerce, for what it's worth, again, I have no relative comparison except to say that nothing has ever not worked perfectly with the Payflow Pro system, originally from CyberCash and now from VeriSign. I mean, for what it's worth I can vouch for it absolutely. And I would only caution people to be very careful about all the other aspects of doing an eCommerce site. I mean, you really want to protect your users' information and just do everything you can to prevent there from being any sort of way of exploiting the site other than the back-end credit card charging process, which is really pretty much nailed down.

Leo: Yeah, I mean, Payflow Pro is for somebody who's rolling their own solution.

Steve: Right

Leo: So, I mean, if you're not a programmer like Steve, you're probably going to go to Yahoo! Stores or something like that and let them do the merchant banking and the transactions and all that, or...

Steve: Or just PayPal.

Leo: Or PayPal, or eBay, or there's lots of other ways to do it. Because, I mean, this sounds like a pretty low-level solution in the sense that it's for somebody who's writing an eCommerce system or has an eCommerce system he can plug it into.

Steve: Yes, that's exactly right. And I objected, since I have the skill to talk to the back-end credit card processing system directly myself, I didn't want to give anybody else a piece of my action. A lot of these other services, they're not free. And Payflow Pro is not free, either. I pay some small percentage, a transaction fee and a percentage. But it's not then marked up again by somebody else who's basically using them and putting their own wrapper around it.

Leo: Yeah, you know, PayPal's fairly expensive. I'm trying to remember, I think it's - when you do a $2 donation to us, they take 37 cents of it. So...

Steve: Yeah, PayPal itself.

Leo: Yeah. It's not insignificant.

Steve: Yes.

Leo: That's the total charges. It may be PayPal takes some and the credit card company or whatever. But that's...

Steve: As opposed to, like, 2 cents, which would be 1 percent.

Leo: Right. Yeah, that's a lot. Question No. 2, Russell Johnson in MinneSnowta - not MinneSnowta yet, but it's going to be MinneSnowta soon - wonders about the double-edged sword of SSL proxying. What is your opinion of deploying an SSL proxy, similar to something offered like Finjan? What are the privacy and legal issues of this technology? I helped set up a system using this technology in a public school's network. Also, what disclosures should be made so that users are aware the info they thought was encrypted from the browser to the server is actually being decrypted by the proxy? A CEO of a large company may be unaware that an IT employee or outside contractor can now access his previously impervious encrypted data, like credit card numbers and passwords. I know the intention of the software is to prevent using SSL to hide web surfing activity. But is this going too far? I wonder if ISPs would employ this kind of technology. Now, that would scare me.

Steve: Yes. They can't. Okay. Just to sort of give our listeners a little bit of background, an SSL proxy means that, rather than your browser connecting to the actual remote web server of a site you are visiting, like BankofAmerica.com, with a secure connection, instead this proxy is intercepting the connection, and then it is turning around and creating the secure connection to the remote site. The reason this was installed, for example, in this public school's network is they wanted to prevent, specifically to prevent what SSL is for, which is encrypting the communication from endpoint to endpoint.

So essentially what they did was they inserted their own intermediary endpoint so that the user or the student, in this case, in the case of a public high school, or a public school, but if the student is connecting to the proxy server, that's one SSL connection. And then a second SSL

connection is made from the proxy server to the remote website. The reason this is done is that between those two connections in the proxy server, the encrypted data is decrypted, it's in the clear, and can then be inspected by filtering software. So the school did this because they wanted to be able to insert web filtering software into, not only normal HTTP connections, where it's easy to do, but also into HTTPS connections.

In order to do that, though, the browsers in the school have to have a certificate from the proxy, and the proxy has to create - it's a function of how transparent the proxy wants to be. The proxy has to essentially create certificates on the fly that apparently belong to the third-party remote server, and it signs them. Normally, as we know, server certificates are signed by trust certificate authorities - Equifax, VeriSign, or whoever. But in order to transparently proxy in the way the school district is, there's no way for a man-in-the-middle attack, which is really what an SSL proxy is, if you think about it. It's trying to be a man in the middle to decrypt that traffic.

Well, the beauty of SSL with certificates is, as long as the certificates are signed by a trusted certificate authority, nobody else can sign, like, fake certificate with that authority because nobody else has that authority's private key. The public key allows you to verify the certificate, but you need the private key that would never be released in order to yourself sign it.

So what happens is, within this school network, in order to make a secure connection, all the browsers would have to accept a certificate from the proxy, or just not have HTTPS connections. But apparently they do. And in fact this the way SSL proxies work is that you, you know, all of the browsers have a certificate from the proxy, so that essentially means I trust the proxy to do anything it wants. And in so trusting it, one of the things it wants to do is to decrypt your traffic for inspection.

Leo: Wow.

Steve: So relative to privacy and legal issues, we immediately can see what they would be. First of all, the assumed privacy of HTTPS is gone because it is being - the traffic is being decrypted at the border, inspected, filtered, logged, I mean, anything...

Leo: But it's okay if you trust the intermediate.

Steve: Correct. So, for example, the CEO of the large company presumably knows...

Leo: It's his company.

Steve: Yeah, exactly. It's his company. Especially if he's the CTO of the large company.

Leo: He's the one who said it could be all right.

Steve: Certainly I would say that it would be worth - and probably for, like, school district policy, or corporate policy in a corporate setting, in the policy manual it says you should know that you're using corporate property. These are company computers. They are subject to monitoring, logging, surveillance, filtering. Basically behave yourself.

Leo: I'd like to point out, though, while that's good policy, it's not required by law.

Steve: Right.

Leo: The law says it's a corporate computer. They can do anything they want with it.

Steve: Right, from the get-go, without having to specifically notify you.

Leo: Don't have to tell you nothing. Although I think it's a very good policy to do so. I mean, that's just good citizenship.

Steve: Well, especially if some damage were to occur as a consequence of that proxying, like data escaped from the organization's control.

Leo: Why can't an ISP do that to me?

Steve: Well, an ISP could if you accepted a certificate and treated them...

Leo: They'd have to modify my system.

Steve: Exactly. They would have to add a certificate to your browser. And, I mean, I'm already nervous about the Hong Kong Post Office doing that.

Leo: You keep bringing the Hong Kong - now, the truth is, though, many ISPs now give you a CD that has software to install. Usually it's a PPOE dialer for DSL. But there's no reason they couldn't be, I mean, they modify IE, and they often say "Provided by Verizon."

Steve: And they convert it to their home page and do all that. But you're right, Leo, I mean, from a technical standpoint they would not be prevented from doing this. And this is why every time we talk about SSL and anti-phishing, I talk about looking up the chain of trust because what you would see is you thought you were on eBay, https://www.ebay.com, and you thought you were on a secure eBay page. If this were being done, you would right-click on your browser's page, and you would see eBay, that the eBay certificate had been signed by Cox.net. Now...

Leo: Ah, okay. So it'd still be an eBay certificate, but it would be signed by your Internet service provider or your boss or the school district. It would be clear that it wasn't eBay's.

Steve: Correct.

Leo: Okay. So there is a way to tell. There's no way they can hide that.

Steve: Right. And in fact, well, for users who are smart enough and aware enough to perform this test. I mean, I think it is the case - and this is a point I was going to mention last week when we were talking about the problem of phishing with OpenID service. In general I think we're going to see motion towards preventing phishing attacks in the future. We've already seen the green field in your URL that some browsers will now provide if you have an extended authentication certificate which, you know, you have to pay more money for, which bugs me. But we're beginning to see this. I imagine a system where web browsers preferentially connected over SSL if it was available. So even though you go http://eBay.com, imagine that the web browser also tries to make an SSL connection, and does so, and prefers it over the non-secure connection if it's available.

It used to be the case that SSL was very expensive in terms of computing to establish due to the computation overhead of secure sockets and the public key crypto that has to be done once during the connection. But machines have gotten so powerful now, and servers are so powerful, that it just wasn't - it's still the case that SSL is, you know, doing a secure connection is a little more computationally burdensome, but not so that it's significant in this day and age.

So I would say the other reason an ISP would be not motivated to insert their own certificate is it would really open them up to liability if anything ever happened and it was found that they were inserting themselves in secure connections without really clear, making it obvious that they were doing so. Like, for example, adding their own line to every secure page that's displayed, saying this page has been filtered by your ISP. And if they did that, who would use these people?

Leo: Yeah. For your protection. Filtered for your protection.

Steve: So it's exactly like the old spyware argument. It's like, oh, well, we told our people that we were installing this new search system on their browsers. Like, no, you didn't. And if you did, they would hate you, so...

Leo: So don't.

Steve: Right.

Leo: Eric, anonymously somewhere near the Hong Kong Post Office...

Steve: That's actually what he said in his posting, Leo. I didn't make that up. He was being funny.

Leo: Is there any way to use web-based email to send email while remaining completely anonymous? I tried using TOR, but every web-based email I used asked me to turn on cookies or Java. This, as we know, can defeat TOR and give away your local public IP. I suppose that's why they require cookies and Java?

Steve: Well, now, it's an interesting question because it's the case that your computer can only give away the information it has. And if you're behind a residential or local NAT router of some kind, you always certainly have a private IP, not a public IP. That is, your computer is 192.168. something or other, or 10., or 172. something. So it might be that your computer doesn't know what your true local public IP is. It only has your private IP. So once that's the case, and you're running a TOR client on your machine so that your traffic is encrypted to the first TOR node, the

second TOR node, and the third TOR node, after which it goes out, then you really are anonymous. That is, even a web-based email service that was putting the client IP in the email headers, as we've talked about before, as far as I know they all do except Google. For whatever reason Google Mail doesn't. It uses probably something that they can decrypt into that, but at least it's not there; whereas Hotmail and Yahoo! Mail, the last time I looked, both were putting the client IP in there.

Still, what those web-based systems would see as the client would be the IP of the last TOR node in the chain. So again, your browser, through which you are using this web-based email through TOR, it can't give away what it doesn't know. And it doesn't know your local public IP. So in fact I think you could be, if you're careful, you could be anonymous using TOR, as long as your local machine did not have your public IP, as long as you were behind a NAT router that was giving your machine a private 192.168 that absolutely does not identify you on the Internet.

Leo: There are, I should point out, web-based email services designed around privacy. Phil Zimmermann, the creator of PGP, worked with a company called HushMail. And Hush Mail uses PGP encryption and presumably has lots of other privacy protections. They have secure free email. They also have - by the way, they do certificates now, I see. That's something new. Something to look into, HushMail.com.

[Talking simultaneously]

Steve: And it's a very good service, and well respected.

Leo: Yeah. I think they're absolutely reliable in that regard. Moving on to our next question. David Farrell in London raises another point about TOR, The Onion Router, which we again underscore is used for privacy, for anonymity, not for encryption or security. Dear Steve and Leo, I'm an avid listener of Security Now! and the other netcasts. I benefit greatly from the insight you have to offer, so thanks. You're welcome, David, thanks for the kind words. One thought that I kept having as you two explored some of the issues of TOR recently was that it seems that using TOR is in fact less secure than normal browsing if you're performing ordinary web surfing. And here's what he means.

Without TOR I'm trackable by IP address, of course. But my data, assuming secure WiFi or LAN to router, goes from my PC through my ISP's machine and then is routed to the actual page I'm visiting. The servers with access to my data, such as emails, passwords, and the like, all belong to web hosting companies - well, okay - in whom we place some degree of trust. I assume that Be Broadband or BT or NTL will not be sniffing my HTTP stream for username/password form posts. But contrast that with the use of TOR, especially now that we're seeing that there are TOR servers owned by governmental agencies, the Hong Kong Post - oh, no. Sure, I'm now no longer trackable by IP address, but I place a significant degree of trust in the random person whose TOR node I'm using. As evidenced by the comments in the last show, it is possible and not uncommon for there to be a sniffer at the TOR node. So assuming that I'm engaging in normal web browsing without SSL or a VPN connection, it seems to me the risks of data sniffing vastly outweigh the benefits of anonymity. What do you guys think?

Steve: I think he raises a really good point. Essentially, clearly there's this cluster of interest around TOR's exit traffic because people are doing things through TOR that require anonymity. So you might argue that TOR is drawing people to itself, that is, spies or security researchers or governments or whatever, who are using TOR as potential high-value sources for questionable web traffic. And David's point is that, by contrast, when he's not using TOR, his data goes to his ISP and immediately scatters to the winds. It's going, you know, there's no central, except for

the ISP machine where his data does egress onto the Internet, there's no central concentration of his data as there is in the case of a TOR node. Nor, and I think more importantly, is there any reason for someone to suspect that that data might be interesting to them. Whereas the data exiting TOR nodes, it's probably a little more risqué.

Leo: Yeah. Well, again, let's emphasize, it's for anonymity, not security. And it's good to know that.

Steve: Right, and people who are wanting anonymity, I would argue - and I know you would, too, Leo - that absolutely that's a right. It's something that the Internet provides, and it's something worth having. So it's not the case that people who want to be anonymous are doing bad things. But analysis of traffic has shown that TOR tends to attract that, too.

Leo: Joe Graf, Sacramento, California. He's got an idea for making your feedback form screen reader friendly. We talked about this a couple episodes ago, a blind listener said, I understand what you're doing to prevent bots, but it makes it hard for me to see with my screen reader.

Steve: Right.

Leo: Steve, all you have to do is add some alt text to the images. So just use the alt tag in your images to something that will make sense to...

Steve: I know, Leo.

Leo: ...the user when the screen readers read it. Also, if you want to be XHTML compliant, you have to have alt text anyway. Yeah, of course, alt text makes it accessible.

Steve: And the bots love it.

Leo: The bot can read it just as any screen reader. I think I mentioned that when we were talking. I think I said you can't use an alt text. So that's the problem is anything that a screen reader can read, a bot can read.

Steve: Exactly. And as you said, that's the problem. And I just wanted to close the loop with Joe and for any other users who are thinking that that would work. The problem is that that's text in the HTML, which is exactly what the bots are out there sniffing for. So, I mean, much as I have absolutely no interest or intent to make things difficult for visually impaired users of our forum, and in fact we enumerated the fields on the form in order with Leo typing the tab key so that we could let people know who were listening how to fill out the form and do so successfully.

Leo: I think that's a good way to do it. That's your cheat sheet. No robots listen, as far as we know, no robots listen to Security Now!. Let us know if you're a robot and you're listening.

Laura Cooksey in - is Laura next? I want to make sure I didn't skip anybody. Yeah, Laura Cooksey in Burke, VA, a suburb of Washington, D.C., says: I've used Steve's free Unplug N' Pray widget to disable the Universal Plug and Play service on all my computers. Also - I like you, Laura - disabled UPnP on my Linksys WRT54G router. However, Microsoft strongly recommends that UPnP be turned on in routers used with Xbox Live to ensure best performance in multiplayer games. Since I've disabled UPnP on my computers, is it safe to enable UPnP in my router full-time, or should I only enable it when I'm planning to game and then disable it when I'm done? And while we're talking about that, maybe I could ask you about UPnP2, which Microsoft says solves the security issues of Universal Plug and Play.

Steve: I have not yet looked at UPnP2, but I definitely will.

Leo: That's what their Windows Home Server uses. And since a lot of people I think are going to use Windows Home Server as a router and a bridge to the Internet, I think we should look into this.

Steve: Oh, I absolutely will for sure. Laura's question was great because the answer is, well, first of all, no, that it is not safe to enable UPnP in the router. And in fact, if you had to disable UPnP facility anywhere in your network, Laura, the one place you want to disable it is the router. You could even leave it turned on on all the machines that are behind the router. The reason I did Unplug N' Pray widget was that, as was the case with probably every server Microsoft ever created - and of course Universal Plug and Play service is a server, meaning that it had an open port. So anybody not behind a router had this open port exposed to the Internet, and there was a buffer overflow that allowed people to take over your machines remotely. So at the time I immediately created Unplug N' Pray just to turn it off because also at the time almost no one needed it.

Now, for example, we're seeing the case that, with the Xbox, it would like to have Universal Plug and Play enabled on the router so that it's able to essentially open incoming ports back through the router to your LAN. Thus the danger of having Universal Plug and Play first edition, that is, v1 of Universal Plug and Play, enabled on your router is that, even with the services disabled, your Windows Universal Plug and Play service disabled, it's still possible, just using some UDP and TCP traffic that any trojan could easily generate, it's still possible for them to query your LAN, find any Universal Plug and Play-equipped devices, determine it's a gateway, i.e., your router, and then talk to it behind your back in order to enable incoming unsolicited traffic, which is exactly why we've got the router there as one of the substantial security benefits, for example, if Windows, even in a vulnerable state, were behind the router, that Universal Plug and Play vulnerability would have never been a problem. It was only for people whose machines were directly on the Internet, which even now just saying that just sort of makes me shudder because having a NAT router is just such good security. But if you enable Universal Plug and Play on a router, and something did get into your machine that could talk to the router, then you're in trouble.

Now, the good news is - I did some research on this - there are only two ports that the Xbox actually has to have open. Microsoft documents this in a Knowledge Base article, 908874. And so if you just go to Microsoft.com and put in KB - for knowledge base - 908874, you can go there. But I can also tell you what it is because it's very simple.

The Xbox Live system needs the UDP port 88 and both UDP and TC ports 3074 mapped into your network. So what you can do is - and obviously Laura is listening to this and is very tech savvy or security savvy, you would like to give your Xbox a static IP within your LAN. Normally when you turn a computer on it gets the next address available, 192.168.0.1, .0.2., .0.3, .0.4, and just sort of goes up that way. What you'd like to do, though, is you'd like your Xbox to

always receive the same IP so that it's not floating around, its IP is not changing. That's because you want to forward those ports I just named, UDP 88 and TCP and UDP 3074, you'd like to forward them to that fixed IP where the Xbox will always reside.

All routers now allow you to define a fixed IP based on the MAC address of the LAN adapter on the network. So you first look at the client list to find your Xbox, and it'll be listed there based on its IP. Then you can figure DHCP to always give that MAC address the same IP. You probably want to pull it up out of your normal range, like 192.168.0.40 or something. Doesn't want to be too high because some NAT routers won't allow the numbers to go all the way up to 255. But 40 would be safe unless you've got 42 computers in your LAN, and I think few users probably do. So that would always give the Xbox the same IP.

Then you configure what's called "static port forwarding" to statically map those ports to the Xbox. The beauty of this is that you do not need Universal Plug and Play enabled. Essentially you've done what the Xbox would have done anyway with Universal Plug and Play, but you've done so by manually configuring it rather than having it automatically configured. And it's the automatic configuration that is so worrisome with Universal Plug and Play. And in that case you've got the highest level of connectivity for your Xbox. Microsoft uses some terms; there's, like, three degrees of connectivity. There's moderate and strict and something else. I think it's open. And you get the open degree of connectivity, which means anybody else is able to connect into you, even if they're behind a NAT router which has got strict connectivity because they haven't gone through what you have to get yourself configured.

Leo: So you've just described something called port forwarding. And basically UPnP makes it easy for people who don't want to go through this trouble or understand it to get this kind of connectivity. But the risk is that a bad guy can get this kind of connectivity, too.

Steve: Exactly.

Leo: So do it by hand. Learn - and obviously Laura is smart enough to know this - learn how to do it by hand. It's not that complicated once you...

Steve: And then she can tell all of her Xbox friends and spread the word. Because it won't introduce significant insecurity because those ports, even though they're now open, they're only going to go to that IP address where you've put your Xbox. So nothing else can get it. And we hope that the Xbox doesn't have any unknown vulnerabilities that would allow people to crawl into it.

Leo: You're relying on Xbox Live being secure and the Xbox being secure. But, you know, you've got to do that.

Steve: Got to trust somebody, Leo.

Leo: Got to trust somebody, I don't know who. Kalman Dee in Australia's capital, Canberra...

Steve: And we now know the capital of Australia.

Leo: ...has monkeys on the brain. Leo mentioned that an infinite number of monkeys typing for an infinite time would produce, not just a work of Shakespeare, all the works of Shakespeare. I thought this odd assumption over. I think, if there are an infinite number of monkeys, they wouldn't need infinite time. One of them - well, I'm just quoting the old saw. One of them out of the infinite number would type the work up in a finite time, in a few hours, depending on his typing speed. If there were only one monkey, he would need infinite time. You're right So somewhere in the middle of the allocated infinite time the monkey would type up the work, i.e., it would actually - it would be a finite

time. So he could stop and get on with his life. The old paradigm could be modified to "one or more monkeys typing for an unknowably long but finite period of time would produce a work of Shakespeare." I misstated it. It's actually an infinite number of monkeys typing for an infinite number of time would produce all written works of all kinds. Plus a lot of gibberish.

Steve: Yeah, I liked...

Leo: Got a good point, though.

Steve: I liked Kalman's, well, I liked this also from the standpoint of we're often dealing, as we talk about security, with really big numbers and issues of, oh, it's unbreakable; or, oh, it'll only take a few minutes to break it. Or we've got this many bits gives us this much crypto protection and all that. So we're dealing with sort of scales of size and things less than infinity. And so are his monkeys.

Leo: Well, that's the thing about monkeys. That's the thing about infinity. Infinity is a special number.

Steve: Oh, it's big, Leo.

Leo: It's not just a really big number, it's infinity. So infinity times infinity is actually no bigger than infinity by itself. Or is it?

Steve: I don't know. There are classes of infinity, I think. There are mathematicians who, like, spend their days thinking about this.

Leo: Exceeds my meager brain capacity.

Steve: Fortunately we don't have to, no.

Leo: Adam in Ottumwa, Iowa says: I've been dutifully listening along with Security Now! since its inception. Yay. 112 episodes ago. I was thinking about the recent episodes where you and Leo refer to saving the browser's state when hosting eCommerce sites. My question is, how could this work with Safari and private browsing? I haven't used this feature, don't claim to know anything about it; but it's my understanding it does not save any history cookies, session data whatsoever. Is this really true? Wouldn't you have

problems with private browsing and eCommerce sites? Well, that was the whole point of what you were doing is it bypasses this issue.

Steve: Well, it didn't depend upon that. I did not know what Safari's private browsing was until I got this posting and took a look around. And I think it is extremely cool, Leo. It's right there on the Safari main menu. And you turn on private browsing, and it pops up a notice warning you about all the things that are deliberately not going to save your state.

Leo: When private browsing is turned on, web pages are not added to the history. Items are automatically removed from the downloads window. Information isn't saved for autofill. Searches are not added to the pop-up menu. Until you close the window, you still can click the back and forward buttons to return to pages you've opened. And that's where it stops. But just to go on, no cookies are saved; no state is saved.

Steve: Isn't that cool? I didn't even know that was there. And it's funny because, as I was doing some research, I found some blogs that refer to it as "porn mode."

Leo: Yeah. That's what it's for, of course. Everybody knows that. Well, what else are you trying to hide?

Steve: Because, you know, so much effort has gone into eliminating the state of where people surf, with all kinds of third-party tools. And I thought, wow, this is a cool feature for a browser to have, and all browsers ought to have it.

Leo: I think IE7 has something similar. It basically clears history when you close it. The point is, your system was designed specifically to get around that by not using cookies.

Steve: Yes. And in fact my system would work perfectly, even with private browsing enabled in Safari and any other browser, because essentially it's the page itself, the page the user receives carries the state information, not cookies or URLs or history or any other mechanism. So when you submit the form, that page that contained the data, a piece of it goes back with the form to the server. So it is the case that at least GRC's eCommerce system functions without scripting or cookies of any sort because the actual page is where that state is saved. But I just, for users who hadn't run across that private browsing feature, I wanted to put this question in because it's like, hey, I'm glad to know that's there.

Leo: Yeah, that is a nice feature, yeah. I've been getting a lot of emails saying, "Steve claimed he invented this, and it's been going on for years." I don't think you - we said this last time, but I just want to reiterate. You didn't claim you invented it.

Steve: No. I just came up with the solution that worked for me.

Leo: Independently you came up with that solution. But of course you're not the only one who thought of it, and you weren't even the first one who thought of it. And I don't think you claimed that.

Chris Noble of Wellington, New Zealand gets the Gold Star Award. He writes: Nice work putting some hidden fields into your feedback form to trip up the bots, Steve. However, rather than actually specifying type=hidden in the input tag, which bots can easily see and of course could be smart enough to ignore, you could do this via CSS, removing the "hidden" alert from inside the form. In the input tag, use something like class="abc", and then in your stylesheet include .abc {visibility:hidden; display:none;}. Still not bulletproof - in fact it's not because the bot also sees the CSS, but okay. At least it's one step harder for a bot to figure out it's a hidden field that should be left alone. I think most spiders and bots ignore style info. They may well dip into these - you have to include the CSS if you don't put it in line. And they could load that CSS and look at it. So you're right, depends how sophisticated they want to be.

Steve: Yeah, I just liked it because it sort of fits my model of something simple that's providing some additional resistance to bots taking advantage of our technology. And while it's not galactically powerful, it's like, okay, I like that because having the hidden tag right there couldn't make it any more obvious to a bot that this is not a field that a user would fill in. And if you didn't have it, you might assume that the bot writer didn't ever consider that they ought to parse the CSS file and then have to do a match-up of CSS class...

Leo: Right, makes it a little more complicated, yeah.

Steve: Yeah, I mean, it's substantially more complicated. So it's like, yeah, I gave him the gold star.

Leo: Right. Gold star. Clever. Andy in Iowa - another Iowan - had a common question about SpinRite and RAID: I love SpinRite. I love RAID. I'd like to use SpinRite on RAID. Is that okay?

Steve: We get the question a lot. And so I don't want to take a lot of time about this, but I want to explain that there are, in our mind, sort of two types of RAID controllers, what we call "thin RAID" and full-on industrial strength "thick RAID," the distinction being that a thick RAID controller is a controller with a coprocessor, probably has got some caching memory on it, and it's really decoupled the drives in the RAID array from the machine. So the machine dumps a bunch of data on the controller, which the controller caches. And then the controller independently turns around and writes that data in so-called "lazy writing" to the drives of the RAID array.

Now, this is as distinct from motherboards that now often will have a RAID controller on them, but it's a little Promise Technology chip. And all it's really doing is basically allowing that to be a bootable RAID. So there's some BIOS support that allows a couple drives to be booted. And then you still need a software driver in your OS in order to essentially implement the RAID in software.

Leo: People often think that these motherboard RAIDs are hardware RAIDs. They're not. They're software RAIDs.

Steve: Yes, they are. They're just sort of hardware assists that just gets the RAID and allows it to be bootable. But once it gets going, the OS has a driver which does this in software. So, relative to SpinRite, it is generally okay to run SpinRite on a RAID 0, which is...

Leo: A hardware RAID. Or even a BIOS RAID.

Steve: No, in fact you would not want to run it on a so-called thick RAID controller ever because, well, at least not on the controller. That is, what we tell people to do is just temporarily take the drive off of the RAID...

Leo: Ah, do individual drives.

Steve: Exactly. Stick it onto a regular motherboard connection, which you probably have right there on the motherboard, and SpinRite will see it and run on it just fine.

Leo: And that works because you don't need a file system; you don't need to know that the files are all there. You're looking at such a low level, you're not looking at how it's written or anything like that.

Steve: At just the raw physical sector level, right. And so we won't break - SpinRite will never break the RAID. It won't cause it to be nonfunctional. It doesn't matter if you've got RAID 5 or 6 or 27 or whatever you're using. SpinRite will work on it just fine as long as it's talking just to the bare drive, not through the controller. But in the case of the thin RAID, because when you're running SpinRite there's no operating system with its own software drivers, it will see the drives separately in the normal case. However, you still, in a mirroring configuration where you are writing to both drives - but you're only reading from one typically in a mirror. You're not redundantly reading from the drives. So the point is, it does not make sense to run SpinRite on mirrored drives behind a thin RAID controller. Even there you would want to unplug them from that connector and plug them into a regular motherboard controller.

But in the case of striping, where you've RAIDed them to expand the size - so, for example, you've got two 100-gig drives, and you're running in RAID 0, which is where essentially you've created a virtual 200-gig drive, there you could run SpinRite in place with the drives just like they are because essentially the queries are being split between drives, but there's no redundancy of data. It's the redundancy of data, or in the case of a thick RAID it's the caching, which is sort of decoupling SpinRite from the drive. And that's what you want to avoid. You want SpinRite to actually have the full and undivided attention of the drive.

Leo: Makes sense. And actually I'm glad you addressed this notion of software RAID because I've said it for a long time, but nobody believes me. You they believe.

Eliezer Martinez, listening from Puerto Rico, wonders about OpenDNS. He says: I recently started using OpenDNS because it supposedly speeds up the loading of pages. Seems to, he says, mostly because they block known phishing sites. After listening to your latest feedback episode in which you explain the pros and mostly the cons of using TOR, it made me question whether OpenDNS is worth using. Maybe I'm paranoid - thank you, Steve - but I stopped using it in fear that sensitive information like Internet banking transactions could be monitored by the OpenDNS people. What are your thoughts about it? Am I mixing up two different things, or should the same precautions be taken when redirecting web traffic through a third party with OpenDNS?

Steve: Well, that's a really great question. OpenDNS, of course, is essentially an independent domain name server system. And it is popular with many security-conscious users.

Leo: I use it all the time.

Steve: Yes. The idea...

Leo: I put it on my router, so every computer on my system uses OpenDNS.

Steve: Exactly. So the idea being that, first of all, it is generally high performance. Often it's higher performance than your ISP's own DNS servers, that sort of tend to be unwanted stepchildren of ISPs. Just like, oh, DNS is not a very sexy thing for them to be offering, so they sometimes don't get faster servers, or they're overloaded. And the idea, of course, is that any time we're going out on the Internet and surfing somewhere, we're using a URL, generally with a domain name.

So, for example, eBay.com. The first time you use eBay.com, and even periodically, your system needs to go and have that eBay.com converted into the actual Internet IP address. That, as we know, is what DNS does. So if your ISP's servers, your DNS servers, which would be the default when you set up an account with your ISP, if those servers are slow, then it takes longer for you to get to eBay.com, and potentially everything else. So, I mean, the speed of DNS is sort of a not-often-discussed but very important aspect of the overall performance you feel in doing things on the Internet. So OpenDNS has very fast and often faster servers than your ISP. So your browser will get the IP that it's looking for more quickly using the OpenDNS servers.

Leo: But what about the security issues?

Steve: Well, the security is a concern from a standpoint of there have been problems with so-called "DNS poisoning," that is, if you were going to eBay.com, your browser is going to inherently trust the information that it gets from DNS. So if the DNS server lied about Microsoft's IP when you were going to Microsoft.com, your browser would go to the wrong IP. Now, once again, secure sockets solves this problem because SSL connections cannot be spoofed and fooled. But lots of people would just go to Microsoft, you know, http://microsoft.com, not the secure version. So it would be possible to spoof Microsoft and really confuse someone and allow them to be misled because even their web browser would show www.microsoft.com. Oftentimes phishing sites will obscure the URL and use, like, hex notation or decimal notation for URLs, or some fancy way of obscuring what's actually up in the URL, assuming that most people don't glance up there to verify that they're at Microsoft.com. The potential potency of DNS poisoning or some sort of DNS man-in-the-middle attack is that your browser would think it was really at Microsoft.com even though it wasn't.

But aside from security, the other issue here is privacy. And we're going to be talking about this in our - explicitly more in our episode about third parties because here again we have a third-party phenomenon. There's us, and there's eBay or Microsoft, and there's our DNS server. The DNS server does know every site we visit because we're having to ask the DNS server for the IP address of every site we visit. So it's the case that the DNS server really has nothing but our IP address. On the other hand, that's all any other third party generally has about us, although it's a little more potent in the OpenID case because we're authenticating with them, so presumably there's some sort of an account relationship there where there isn't one with a DNS server.

But again, there is this third-party phenomenon that is an issue when you're using OpenDNS, although the presumption is these are good guys; they are not tracking people. They're even going further by - and this is what he talked about blocking phishing sites. They work not to

carry the domain names of bad sites so that your browser can't go to a bad site even if it wanted to. It's trying to look up that DNS name which is deliberately not carried by the OpenDNS servers. It's sort of like - it's very much like what people, some do, as we've talked about with the hosts file. In the hosts file we're blocking DNS queries locally by preventing the query from ever leaving our machine. Instead, the hosts file provides typically just 127.0.0.1 in order to prevent our browser from going outside of our machine. OpenDNS does the same sort of thing, but gives you the advantage of sort of centralized monitoring and management of that.

Leo: And now, Steve, the last question.

Steve: I'm ready.

Leo: Can you still hear me?

Steve: I'm ready.

Leo: You ready? This is going to be a tough one. No, I don't know. Steve and Leo, my question is about utilizing PayPal's virtual debit card. Prior to downloading the software you're asked to verify your sys- this is the card that gives you a new number every time you use it.

Steve: Yeah, it's very cool.

Leo: Yeah. I have a debit card with them, but I'm really tempted to use this system. Prior to downloading the software, you're asked to verify your system requirements. If everything checks out, you can then download and install the software. As I clicked on the "Download Now" button, my system, which is guarded by HostMon, a host file manager - we were just talking about hosts files - blocked the site, and I was redirected to the default safe site. I was able to discover that the PayPal download link actually points to a DoubleClick.net site. Huh? What is the purpose for this? Is it safe? Should I remove DoubleClick.net from my host file manager and disregard future alarms? Wow. Is that an ad that's on there? What's going on?

Steve: No, it's very disturbing. I was sad to hear this. What it means is that PayPal has some sort of relationship with DoubleClick, and that DoubleClick is essentially redirecting people to the actual download. So PayPal has a link to DoubleClick. And in the URL tail is the URL that PayPal wants. So essentially it's a way of allowing DoubleClick to play cookie games with people's web browsers.

Leo: Oh, that makes me mad. Have you verified that they're doing this?

Steve: No, I have not verified. And I will by the next show. I'll close this loop to make sure...

Leo: That is infuriating.

Steve: Isn't that really annoying?

Leo: I hope that's not true.

Steve: Yeah, well, from what he's described, that's exactly what he's describing is that, in downloading the software, his browser is being routed through DoubleClick.net, which creates a first-party relationship with DoubleClick.net, allowing them to play cookie games with his browser, knowing that he's a PayPal user, knowing that he's downloading their virtual debit card. And then DoubleClick.net is then bouncing them back to PayPal in order for him to get his downloaded software. And his host file manager blocked that nonsense because it's got DoubleClick.net nulled out, essentially, so that his system...

Leo: Rightly so.

Steve: Absolutely rightly so. So I wanted to tell our listener, who I guess was anonymous, that he absolutely probably wants to keep HostMon and his host file manager working just like it is because it's doing the right thing. And if what we assume is true, PayPal is up to some shenanigans that are compromising his privacy.

Leo: Well, we'll look into it. I don't want to hang PayPal yet. We should find - we'll figure this out.

Steve: Yes. I will verify to be sure, and it'll be number one in next week's errata.

Leo: Okay. And that was Sidney in Jacksonville, Florida.

Steve: Ah, great. Oh, yeah, there it is.

Leo: Very interesting. Does DoubleClick provide a downloading bandwidth service or something?

Steve: No no no. See, the idea is, all they have to do, if PayPal sends his browser to them and with PayPal's website in the URL tail, that allows DoubleClick to receive the link, to find out where it came from, play first-party cookie games with the browser, and then redirect the user back to the data in the URL tail, which will cause the browser to download from PayPal, but having made a little quick visit through DoubleClick in the process, which is really annoying. Anyway, we don't know that for sure yet, folks. It'll be top of the errata list in next week's Security Now!.

Leo: I'll be sure to tune in. We hope you all tune in. And remember, you can get Security Now! in a 16KB version at Steve's site, GRC.com/securitynow. That's where the show notes live. That's where Elaine's great transcriptions of each and every show are. And of course Steve's got a security form there where you can post your questions so you can get into the next episode of Security Now!. Or actually the next, well, you might be in the next episode or the next question-and-answer episode. We do them every other episode. And that's where you can also get SpinRite, everybody's favorite disk maintenance and

recovery utility. Maintenance and recovery utility.

Steve: It does that, too.

Leo: And of course all his free, useful security tools like Unplug M' Pray, Shoot The Messenger, DCOMbobulator, and ShieldsUP!, which is now at, what, over 50 million uses.

Steve: Oh, it's 54 or 55. And in fact, Leo, I can confirm and announce that next week's podcast will be discussing a new fun free thing that GRC will be unveiling.

Leo: Well, ain't you the man.

Steve: It'll be neat.

Leo: That's great. Thank you, Steve, for all that you do. And thank you all for listening to Security Now!. We'll see you next week. Bye bye, Steve.

Steve: Bye, Leo.