SECURITY NOW!

Transcript of Episode #116

Listener Feedback Q&A #27

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-116.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-116-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 116 for November 1, 2007: Your questions, Steve's answers.

It's time for Security Now! Episode 116, a brand new month. And Steve Gibson, our security guru, is here.

Steve Gibson: Hey, Leo.

Leo: Hello, Steve. Good to talk to you.

Steve: Great to be with you.

Leo: So we are going to do a little Q&A thing in just a little bit. Number 27. We've been doing quite a few of these. I didn't realize we'd gotten so many in. That's exciting.

Steve: And I'll tell you, as I read through the email, I just - we're getting such great feedback from our listeners. I just - it's just delightful to read...

Leo: Well, we have smart listeners, which is kind of neat.

Steve: Yeah, with definitely involved listeners, for sure.

Leo: Yeah, yeah. So we'll get to that in just a second. Do you have any errata or things you want to talk about?

Steve: Yes. I've got two things. There was a lot of commentary that I received about our talking about the Firefox Master Password. And our listeners brought two things to my attention. One is that there is a very frightening piece of freeware called IE PassView. You and I did a spot on your radio show last weekend, Leo. And you'll remember that we were talking about the importance of setting the master password in Firefox, as we also discussed on the podcast, because without that anyone could come along and look at all of the sites, usernames, and passwords that Firefox had stored. Setting a master password protects it and encrypts it. But on Firefox it turns out that there is a brute force attack, and several open source pieces of software that will successfully brute-force reverse-engineer your master password. So I wanted for our listeners to know that all the rules about a really hard-to-guess password must be applied to Firefox.

Leo: Now, people would have to get access to your physical system to use these attacks.

Steve: Correct, correct. Although it's an offline attack, so you could grab the files and then attack them separately. On IE, things are a little bit worse because most users are not familiar with this notion. You asked me on your radio show whether IE had the same sort of facility that Firefox does. And when I said no, what I meant was there was no way to display them, although IE certainly does offer to remember them. When you're typing in a username and password it'll say IE can remember this for you, would you like IE to do so. Many people say yes. Well, this freeware that I mentioned called IE PassView, you put it into Google, IE space PassView, it's the first bunch of links that come up from a company called NirSoft, NirSoft.net, that's a company I know, and they're good guys. They've got a piece of freeware that shows the same thing as Firefox.

Leo: Great.

Steve: But because IE has no protection at all, it's wide open. And so I, when I learned about this, I ran it. And it's like, oh, look at that. I mean, it was like a little bit of a blast from the past. I've been looking at websites...

Leo: Every site you ever logged into, yeah.

Steve: Yes, no, I'm not kidding. It's all there. And it's like, oh my god. I mean, so...

Leo: As I remember, these were stored in the registry. That's - so you could even go into the registry and look at them. It's not like they're encrypted.

Steve: They're wide open in the registry?

Leo: Yeah, I think so, yeah.

Steve: Well, there's something called password-protected web space or something. I think Microsoft has been attempting to do a better job. One thing this utility does allow you to do is delete them. So that's nice.

Leo: That's good, yeah.

Steve: So anyway, I just wanted to tell our users, make them aware that there is this tool for IE. So they need to know that their username and passwords - oh, and this thing will even dump it out into HTML or into a file. So...

Leo: Oh, great.

Steve: So, I mean, it's dangerous.

Leo: Yeah. All right, well, that's good to know. And once again, I think you probably should not be using IE to remember your passwords, since you can't protect it, and use something like AI RoboForm, which is encrypted and will remember the passwords and, I think, replace the IE password mechanism so you don't have to worry about IE...

[Talking simultaneously]

Leo: ...as well, yeah.

Steve: That's great.

Leo: Good.

Steve: And then my other last little blurb here is I had another fun, and always different, SpinRite story. This is from Robert in Aberdeen, Scotland, who says, "I've been using SpinRite for a while now, ever since it saved my data." And he says fortnightly backups, 13 busy days, then his hard drive fails half an hour into a more recent backup. So he was in trouble. Anyway, he says, "So one day my dad's laptop refuses to boot. The problem is, he's about 400 miles away from me, and neither of us had the time to go to see the other for a few days. And I couldn't mail the trusty SpinRite boot CD to him because there was a postal strike. So email and SpinRite's small ISO file came to the rescue. I talked him through getting his machine to boot from a SpinRite CD, which he burned from the SpinRite ISO, and getting SpinRite going. I heard nothing for about an hour. Then I got the 'thanks' phone call. Needless to say, that's another sale you're getting from the, 'Here, try this,' method of advertising. Thanks for such a great product and a great podcast. I really feel from the podcast that I know you, and as a fellow geek, enough to know that all of the above would be fine with you."

Leo: Oh, that's neat.

Steve: Meaning, you know, did I mind him sending his dad a copy of the ISO? Of course not. And I appreciate that his dad bought a copy as a consequence.

Leo: See, that's the way to do it. I mean, you know, people will pay for stuff if it benefits them, if they use it and they love it. And you only get the negative by locking it down so tight. Then people don't - they go, ah, screw 'em. So I'm proud of you, Steve. And I think that's a great story. That's really a nice story. Are you ready, Steve, for 12 questions?

Steve: Let's go.

Leo: And 12 answers from Mr. Steve Gibson, starting with number one, Dave F. in San Francisco. He's asking about what we talked about last week, the Perfect Paper Passwords: Steve, it seems to me as though your policy of no passcode reuse for any given four-character key in the PPP system weakens the system. An attacker knowing this can watch the keyspace for a long time and gradually weaken the system by crossing off potential keys. Well, now, we talked last week about how many keys there are. Sure, it would take a long time to use the system enough, but every login would reduce the future choice space. If instead you have a practical constraint on duplication, say no duplication in the span of 12 cards, and your PRNG, or Pseudorandom Number Generator, is good, then you get the benefits of no local replays being viable, but the keyspace not being reduced. That's not a bad point. What's the flaw, he says.

Steve: Well, okay. I think it was a bit of a terminology problem. When I talked about never reusing a passcode, I meant never reusing it by its position in the passcode sequence. Essentially what...

Leo: So reused by - purely randomly.

Steve: That's exactly right. Every single time there is a one in 16.77 blah blah blah million chance of any one of those passcodes appearing.

Leo: So you do reuse them.

Steve: Well, yes. What I'm not reusing is I'm not reusing the sequence. I'm never going back and saying, oh, well, let's start...

Leo: Let's use it again.

Steve: Let's start at number one and go forward again. It's like, no, we've got so many of them, I mean, a gazillion gazillion of them, there's no reason not to just keep going forward, or choose a new sequence key that will generate a new sequence of them. So it's not that I'm not reusing individual codes; it's very possible that, well, not very, somewhat possible, okay, it's very possible but very unlikely that two codes will occur very close to each other that are the same. And certainly there's no benefit in guessing because any one of them could be any of those 16 million, 770-some thousand possible passcodes.

Leo: Okay. And by the way, because it's 16.7 million, it's not like crossing them off is going to be much use to a hacker. I mean, it's only a theoretical reduction in security.

Ron Goodbin of Clifton, New Jersey needs some IP spoofing clarification. Steve, you've talked about how when a client establishes a TCI/IP connection to a server, there's no way the client can spoof their IP. When a client establishes a connection to a server, there's no way the client can spoof their IP. If so, what is an IP spoofing attack? Is there absolutely no way someone can fake their IP when you've established a TCP/IP connection? Some clarity on this would be much appreciated. Well, he raises a good issue. I thought you could spoof an IP.

Steve: Nope, not with a TCP connection. The reason is, the way a connection is made is...

Leo: Oh, it has to get back to you.

Steve: Exactly. It's that three-way handshake. It requires two roundtrips, that is, the so-called SYN packet, short for "synchronized," that goes from the client that's initiating the connection to the server that has the open port which is waiting for the connection. The server receives that, and it sends back its SYN/ACK, which is to say its own SYN packet combined with an ACK, an acknowledgement of the receipt of the client SYN. Well, it sends it back to the IP that was the source IP on the packet coming in is now the destination IP on that SYN/ACK going back out. If that were a spoofed IP from the original sender, then the SYN/ACK would be sent to that spoofed IP, not back to the sender. So, while, sure, you're able to spoof incoming SYN packets, and that is in fact what a spoofed IP attack is, is just flooding a server with random, made-up...

Leo: Because you don't care about the return.

Steve: Exactly. You're not trying - there you're trying to do an attack, a bandwidth attack on the server. You're not trying to actually establish connections. So in order to establish a connection you have to be sending the packet from a valid IP. And then the SYN/ACK comes back to that IP, and that's the second leg of the three-way handshake. And finally, the client sends its acknowledgement packet back to the server. And the beautiful thing about that is that, from the original designers of the Internet, that requires two roundtrips, one from the client to the server and back, one from the server to the client and back. And that verifies that the routing between those two endpoints is in place for packets traveling in both directions. So it makes sure that everything is intact, and it does validate and verify the IP addresses of each endpoint.

Leo: That's, you know, we didn't talk about this last week. But when the Rockies put their World Series tickets on sale for the first time last week, they decided not to send - you remember this story. They decided not to sell them at the ticket booths. They sold it all online. They got 8.5 million requests in the first hour and a half, so many that they were only able to sell 500 tickets. That was probably a SYN flood; right?

Steve: That's also question number five.

Leo: Oh, well, we'll get to it in a second. Sorry, I didn't read ahead. Let's move on. Ferruccio, writing from Sharon, Mass., raises a good point: Steve, after the discussion of why no one would write or should write their own encryption routines for production use, which I wholeheartedly agree with, I was surprised to hear you say you wrote your own random number generator. Well, no one should be writing custom random number generators for exactly the same reason they shouldn't be writing their own encryption code. It's a solved problem. There are many excellent algorithms for generating pseudorandom numbers. It may well be that your algorithm works perfectly, but I'd never use a custom random number generator without having a really good reason to do so. And I would at least run it by someone who has a lot more math/crypto experience than me. So what's the deal, Steve?

Steve: Well, I got a kick out of his point because he was essentially turning the point I was making back around on me.

Leo: Yeah.

Steve: He's completely correct, although remember that last week I did mention that the Perfect Passwords page on GRC was using a pseudorandom number generator that I assumed would be super high quality because I got it from...

Leo: It came from VeriSign.

Steve: Yeah, actually, Leo, it was from RSA Labs.

Leo: RSA, that's right, yeah.

Steve: The major crypto people. And it turned out it was a pretty crappy random number generator. We had some guys that did an analysis by sucking down a gazillion of those Perfect Password pages and then analyzing it and determining that its entropy was not as high as it could be. I mean, it was good, but it wasn't fantastic. So I wrote my own, and now it's fantastic.

Leo: So it was validated by the entropy test?

Steve: Oh, yes. It's got, like, 7.99997 or something bits of entropy out of a possible eight. Because it's, you know, it's doing bytes. And so the maximum possible entropy would be 8.0. None of them do 8.0. Mine's as good as anyone. And it's the fastest one that has been tested among any random number generators because I based it on the Rijndael crypto, which is an implementation I wrote myself in Assembly language.

Leo: So you didn't invent the algorithm. You just wrote your own implementation of an existing algorithm.

Steve: Well, I wrote my own implementation of Rijndael. And remember that, if you put anything into a really good, state-of-the-art cipher, what comes out is pseudorandom. And so I just run a counter on the input. And what comes out is really good pseudorandom.

Leo: Okay. So you really didn't create a whole new algorithm for pseudo number generation, random number generation. You used Rijndael.

Steve: No. I didn't do something like that wacko with the virtual matrix encryption last time.

Leo: Right, right, right, that makes sense. Mike Gray, who listens in Tacoma, which bills itself as the world's most wired city, is plugged directly into the Internet. Good for you, Mike. I only have one computer. Do I still need a router, or is my firewall enough? Isn't a router just a hardware firewall?

Steve: Well, that's a good question. I mean, I would feel relatively naked if I had a publicly routable IP, and a Windows machine was just sitting right on it. On the other hand, servers are publicly routable IPs, and they're sitting right on the Internet. So...

Leo: But they're hardened by professionals, too.

Steve: That's generally the case. You absolutely want to make sure, I mean you want to make sure, that your software firewall is running. And that's...

Leo: I'd still use a router.

Steve: I would, too, Leo. I mean, the Achilles heel with Windows is - and you and Paul were talking about it on Windows Weekly in the last couple weeks, is the idea of putting Windows on the Internet with no protection, I mean, it's just like game over. You're just taking - your Windows machine is just taken over almost immediately. Certainly, if you were also in the process of, like, trying to download updates to the original XP build, that has years of exploits wandering around the Internet right now, you would never get a chance to get Windows Update updated and up to speed.

Leo: Well, you've also pointed out that, because software is running on the computer, the same computer that is running other applications, those other - if you get malware on that computer, it can see this firewall and disable it.

Steve: That's exactly the case, yes.

Leo: A hardware router is just a dumb box. And it's a lot harder to hack a dumb box because you can't get software onto it.

Steve: Well, yes. As long as you make sure you change the default administration username and password. One of the things that Mark Roberts did when he was testing all those 66 routers that he has was...

Leo: Mark Thompson.

Steve: Mark Thompson, sorry, Mark Thompson of AnalogX, is that he developed some code that attempted to log into routers just using a bunch of sort of standard logins. And more often than not he was able to. And we know that there is malware now which is attempting to log into your local router in order to change settings. Which means there's malware we know that is trying to shut down your Windows firewall. I just - a firewall is so important that I would no longer trust any software firewall to be up and running all the time. And it has to be up and running all the time. So routers are so inexpensive at $49 that it just - it's really good security to have a second line of defense.

Leo: There's also another advantage to a router if you're a DSL subscriber, and that is the router will do the PPPoE negotiation, so you don't have to have any special software on your computer. Often that software is just junk.

Steve: Right. And remember that we've also talked about that most software can't detect what your public IP address is. So software running behind a router gets a 192.168.*.* address, some nonroutable IP. Software can't give away information it doesn't have. So not letting software know where you're located, even your own software, is a little bit more secure also. It's just a good thing to have a little bit of hardware there.

Leo: It's cheaper than buying a software firewall, frankly.

Steve: Yeah.

Leo: Brian F. in Denver, Colorado - here's a question - wonders, crash or no crash? He says: I'm a long-time listener to Security Now!, a SpinRite user, and my local GRC enthusiast. The other day I saw on the news a story about server problems with the Colorado Rockies website - I have an affiliate in Colorado, KCOL, and did a couple of interviews over there about this.

Steve: Wow.

Leo: Yeah. I thought you might be interested in hearing it. Apparently the Rockies decided to sell all their World Series tickets online only, and had set a time to put them on sale. After only 500 tickets were sold, the servers went down. They claim malicious attackers took the site down. And seeing that there were over 8.5 million connection attempts in 90 minutes, sounds possible. But I wonder if it was just that they were unprepared for the flood of traffic they received. 8.5 million hits in 90 minutes is a whole lot, but could this really have been just an angry sports geek with a bot network and a friendly DDoS button, or could that be a lot of fans hitting reload and trying to get their tickets? I'm not a big sports fan, but I found the story interesting. Thought of you guys, and I'd like to hear your take on it. I'd like to hear your take, too, because that was the question, of course, that these radio stations wanted to know, is the Rockies claim it was a DDoS attack. Was it a DDoS attack?

Steve: No.

Leo: 8.5 million in 90 minutes? Is that too many for it to just be fans?

Steve: No, I don't really think it is too many, given what it is that they were selling and the popularity of it. Also I don't know what 8.5 million hits means. How were they counting hits? In a DDoS attack these days you get 8.5 million packets in a minute.

Leo: So 90 minutes is not a big deal.

Steve: Exactly. What I think is probably happening, and I don't know if you've noticed this, Leo, is that web pages have so much technology behind them now, and that technology is anything but optimized, says Mr. Assembly Language here, that these things are taking a long time to go. It might very well be that they had some sort of a - certainly they're going to have some sort of an active site that's doing things with cookies, and it's got code running behind the ticket site. Maybe they've got a system where fans are able to choose which seating they want. They're able to, you know, who knows how much processor time behind the scenes was going into servicing an individual ticket purchase.

Leo: We know a little bit more now, by the way. They had a CAPTCHA on the site. And the reason they think it was malicious is because they were getting all these bogus CAPTCHA entries. But that doesn't mean it's a DDoS attack. In fact, what it probably is is scalpers using bots to try to buy tickets.

Steve: Wow, interesting.

Leo: So, and by the way, the next day they revamped their system in ways that they didn't specify, and they were able to sell all 50,000 tickets in about two and a half hours.

Steve: Yeah, I think that demonstrates that something was wrong with their system that was causing them some sort of a serious slowdown, and it wasn't an attack.

Leo: Yeah. In fact, McAfee Avert Labs, Dave Marcus there said, you know, it sounds like they didn't configure their software right. They should have kicked off users that tried to trick the system. He said, quote, "I wouldn't call it malicious, it's just somebody trying to buy more tickets than they're allowed to in an automated way."

Steve: Right.

Leo: Alves said it was malicious because it was an attempt to disrupt the ticket distribution method, but that's not a DDoS attack.

Steve: Right.

Leo: Yeah. Really interesting story, an example of, frankly, not being prepared.

Steve: Exactly. I think they'll know better next time.

Leo: Everyone will. Don Ramm in Chula Vista isn't convinced: Steve liked the idea of not entering the digits from the PayPal token when entering the password, to cause PayPal to come back and ask for it. In fact, I've been doing that ever since you said that, Steve.

Steve: Me, too, Leo.

Leo: Since only PayPal knows you have a PayPal token, this would foil a phishing site. However, if one did end up at a phishing site, when you enter your PayPal credentials, couldn't that site immediately send that info to PayPal and react accordingly? That is, you know, because PayPal would say, okay, now give me the second part. And when the phishing site sees that, it could then ask you for the token.

Steve: Yup, he's absolutely right. I didn't intend to indicate that this was foolproof anti-phishing detection. But it's just one additional thing. I mean, we know that security is as good as it can be, but often far from perfect. So we do things which are trying to raise the bar, just because raising the bar is a good thing. I've had occasion several times to purchase things ever since that great idea we got two weeks ago. And I now enter my password separately. I mean, I'm still making sure that I'm really on PayPal, doing my standard anti-phishing tests. But I really thought that that was a good point is that right now phishing sites are probably not turning around and checking PayPal.

Leo: Not enough people use keys, I think, so that it's not worth it for their...

Steve: Exactly.

Leo: Yeah. So it's just, you know, it's a little more security. It's not perfect.

Steve: But Don is clever, and he's absolutely right. You could certainly do more of a proxy attack like he's talking about, in which case it is possible for one site to emulate another, even if it's got that kind of active behavior.

Leo: James Lewis of Colorado Springs has some good news. The PayPal hardware token is washing machine safe. He says: I was very glad to learn of the token security key from PayPal. I ordered mine as soon as I got to a PC. The other day I searched and searched my house for it because I needed access to my PayPal account. When I finally found my key, it was in the washing machine. Oops. After letting it dry out overnight - good thinking.

Steve: I love this. It started generating codes again.

Leo: It just did. However, the codes were no longer valid on PayPal or eBay. My guess is the wash probably caused the internal timer to reset, which put me out of sync. But all I had to do was reactivate the key, and it was good again. Oh, how interesting. I just thought you and your listeners would like to know. That's funny.

Steve: Isn't that great? I love that, you know. So clearly he got it out, you know, he fished around and rolled up his sleeves and pulled it out of the water, but it was still dead. So he shook it out and then waited for it to dry. And then it came back to life, but it was no longer in the proper location in terms of its sequence of time-based keys. So then he resynced it, and he was good to go again.

Leo: So what happened is it probably stopped working for a period of time, and that's how it got out of sequence?

Steve: I would think, yeah.

Leo: Jeffrey Wurzbach in San Diego wants to know more about the games Comcast has been playing lately. We talked about this on TWiT. An Associated Press reporter says that Comcast has a new method for shaping traffic. From what the story suggests, it looks like a man-in-the-middle attack on the subscriber's computer. Is this really a man-in-the-middle attack? Is Comcast's system application specific? In other words, would it block only Torrent/P2P networks? Or does it say, look, there's a lot of upstream traffic, kill it? The MSNBC story says it's not app specific, but I question the accuracy.

It's always hard when you read mainstream media reports of this stuff to know what happened. Just to recap what the AP reporter said, is it looked like Comcast was interrupting the P2P connection and spoofing your software to say, disconnect please, I don't want any more. So it was disconnecting your peer-to-peer connection without your permission, which is kind of an interesting thing to do.

He says: Is it ethical for Comcast to do this? Is it even legal? From my understanding, this kind of denial of a service on somebody else's system is illegal. And certainly there are people who are saying that now, the Electronic Frontier Foundation and others.

Steve: Well, it's interesting. I asked Mark Thompson about this. Mark has written a very popular and successful BitTorrent client.

Leo: Oh, so he would know a lot about this.

Steve: Well, actually he knows everything about it. It turns out, well, actually between the two of us, because he was - apparently some ISPs have been doing something like this for several years. So aspects of this is not new. And there have been some BitTorrent client pushback against this. If you are BitTorrent aware, you are able to look at the protocol. And there's a bitmap showing the segments of the torrent which are available and which are still necessary. And so that's how the torrents are able to assemble themselves in individual chunks. What some ISPs were doing, and I don't know that this is exactly what Comcast is doing, but what they were doing is their intention was to allow someone to download something, but not allow them to upload it. And so what they would do is they would come in at the last moment, when this bitmap was almost complete, and interrupt the connection. Well, now I know what's going on because all that you would need to do, somebody doesn't really have to be a man in the middle. They can just be a passive observer. And if you see the traffic going back and forth over a TCP connection, you simply send that a TCP reset packet. And it will...

Leo: That's what happened, yeah.

Steve: It will drop the connection. And so essentially it is absolutely possible for an ISP that wants to be belligerent against its users to essentially drop connections by sending them end-of-connection packets spoofing the source IP. You also have to have the synchronization number in the packet within a valid range, which is easily done when you're monitoring their traffic.

Leo: But they do have to have your IP address in that reset packet, as well as that synchronization key.

Steve: It's got to be both. It's got to be the source and destination IP. Otherwise the receiving stack will reject it.

Leo: Oh, I see. This apparently is a technology that is used by a Canadian company called Sandvine. Some Internet service providers use Sandvine. Apparently Sandvine is not saying whether they're being used. Comcast won't say what they use. But that's what BitTorrent, Inc., President Ashwin Navin said. He said this is consistent with how Sandvine works. And that would make sense. I don't think Comcast is writing code to do this. As to the legality of it, we'll just have to leave that for somebody else.

Steve: Yeah, I would agree that it's a relatively high-tech attack for your typical ISP to go through. It's entirely believable that some third party would come up with something that any ISP could tack onto their network in order to do this. And I'm glad it's getting some attention. I mean, there's this whole issue of Net Neutrality that we haven't ever really discussed on this podcast, but this notion that ISPs are wanting to treat different classes of traffic in different ways. For example, an ISP that's offering telephone service might be giving its Skype users a lower quality of service for Skype traffic than their own users receive.

Leo: Right, right. Yeah, Skype's, I mean, Sandvine's Intelligent Traffic Management does exactly, exactly what we're talking about. They even bill it that way. So it is a - promises to save bandwidth for Internet service providers by managing and redirecting filesharing traffic. Yeah. Redirecting like, go away.

Steve: Then, yeah, snipping the string that connects the two computers.

Leo: Go away. Bye bye. Matthew Paulson in Madison, San Diego - I'm sorry, Madison, South Dakota - wonders about the security of onscreen keyboards. We've talked about this before. I know that keyloggers are a major way for malicious individuals to steal account information from users. I was wondering, if a bank's website were to implement an onscreen keyboard, where the user clicks on the keys to enter their password, would that be more secure? I know that keyloggers can also measure mouse-click positions. So if the key layouts were randomized at each time, would this be a secure means of authentication that's immune to keyloggers? I think we - he's doing this actually as an undergraduate research project. I think we talked about this recently, didn't we?

Steve: Yeah, we did because it was - in fact, it was in last week or two weeks ago's episode. Someone was asking about the notion of a keyboard jumping around the screen. And the sense was that, well, the reason I included this question was that I want to always reinforce this notion that security is not absolute. It's a relative thing. You can design systems which are absolutely secure as you can make them within the constraints. So, for example, having a keyboard that randomizes its key positions is going to be better than not randomizing its key positions, which is going to be better than not having an onscreen keyboard and using a physical keyboard, which is really not very good because it's really prone to keystroke logging.

So absolutely, the more you can do to confuse things and to slow down the bad guys, the better. You're going to have a lower chance of being compromised. But again, if your model is perfect information, that is, anybody logging in can see what you can see, then even an onscreen keyboard could be mapped and tracked. Which is why, for example, in last week's episode I talked about this Perfect Paper Password system where its key is, so to speak, it never reuses the same login twice. Which means the act of logging in obsoletes that login so that even somebody with perfect knowledge can't use that knowledge. So, I mean, that's substantially stronger than anything that uses a repetitive login and some sort of a puzzle. You might even consider that, like, moving the keys around the keyboard is sort of a puzzle. Lord knows it's going to confuse your users. Wait a minute, where did the "E" go? I saw it over here just a second ago.

Leo: It's going to drive them crazy. That's my only complaint is it's going to make them nuts.

Steve: Right.

Leo: Let's see. Frank S. Werren of Sherman, New York needs random numbers to go: I sometime work on networks within a closed environment - this guy's probably working for the NSA - with no access to the external Internet. It'd be nice to have GRC's Perfect Password page packed into a zip file that could be unzipped at a remote location, or a GRC utility that has a pseudorandom number password generator in a nice, neat file. I trust you, Steve, and you make the right software with no holes in it. And I know I could install your utilities with no fear of compromise. Any thoughts? Yeah, could you make this a standalone?

Steve: Well, it's interesting because one of the offshoots of the Perfect Paper Password system is that little EXE, remember that's 11K, and it depends upon a 17K DLL. So together, what are we, at 28K. Actually 8K of that combined is just Authenticode. But it has a very high-quality pseudorandom sequence generator in it. If you say PPP space and then a null string, just open and closed quotes, that tells the EXE to generate a sequence key, which is hex for a 384-bit extremely random output. So anybody can just grab the PPP EXE and the PPP DLL and have a very portable, high-quality, pseudorandom number generator.

Leo: Very cool. And just run it via command line.

Steve: Exactly.

Leo: Steve. That's a nice side effect.

Steve: Yeah, it's neat.

Leo: I never even thought of that. Fred Zanegood of Orlando, Florida wants some OpenID Delegation clarification: A few episodes ago you briefly mentioned OpenID delegation. Can you explain this further with detailed information on its purpose, implementation - didn't we do a whole series, an episode on this?

Steve: We did one on OpenID. We didn't talk about delegation too much, so I thought I would just answer his question really quickly.

Leo: Okay. He wants to know exactly how it's used, its purpose, its implementation. I understand the concept of creating an easier identity alias for the somewhat cryptic and lengthy one used by VeriSign. But beyond that, I don't see how it actually gets utilized. Unfortunately, there doesn't seem to be much information on this yet. You and Leo also spoke about using SeatBelt for Firefox. How does this fit into the picture? Does using SeatBelt preclude you from having to add the few lines of HTML to your home site? Leo mentioned he uses Leoville.com, I believe - that's correct - for his delegation home base.

So the idea of this is with OpenID, when you log onto a site that supports OpenID, it will then ask you for your domain. You could say - and give it Leoville.com, and it will then go to the right OpenID provider and give you the OpenID login. Why do you need this extra step, I guess is what he's asking.

Steve: Well, yeah. So I just wanted to clarify for Fred's understanding here that you could go to a site that wanted you to use OpenID to authenticate and give them your, for example, steve.gibson.pip.verisignlabs.com URL, essentially. In which case that site would go directly to - basically to that URL in order to pick up the OpenID to begin the whole authentication process. However, delegation allows you to give it a nicer URL that you control because the first thing that the server will do is look on the page, the HTML page that comes up for specific OpenID delegation instructions up in the metatags of the page. So the idea is that instead I could simply give it, just as you do, Leo, giving Leoville.com, I could give it GRC.com. So the site that is asking for my credentials, I give it GRC.com, and the first thing it does is look in the metatags of the default page that comes up at that URL for delegation instructions. And that then contains the gnarly URL pointing it to my actual OpenID delegator.

Leo: So it's just easier to say GRC.com or Leoville.com.

Steve: Way easier, yeah.

Leo: But that's the only - in fact, if you look at the HTML code you embed, it's really pretty much just saying, oh, go over there.

Steve: Exactly.

Leo: It's too long.

Steve: It's just a pointer to another site.

Leo: Right, it's very simple. So unnecessary, a convenience, that all. No additional security. Finally, Tim Knittel of Lexington, Kentucky wins this week's clever idea award. And now, ladies and gentlemen, our winner of the great idea of the week contest, which is not a contest, and he's not a winner.

Steve: But it's a great idea.

Leo: It is a good - we could find some stuff to give away. I'll think about it. It occurred to me while I was listening to your award-winning podcast - did we mention that this is the best science and technology podcast? I don't think we did.

Steve: I don't think I did. But I think everybody knows because they made it happen.

Leo: That's right. That there's a fundamental difference between the way blind website users and spambots interpret alt tags. The difference presents a possible solution for allowing blind users to understand and use image-labeled form fields while still preventing the spambots from understanding them. The difference is spambots read alt tags. Blind users listen to alt tags. Therefore, to a blind user it's equivalent to write "email" or "ee-meyl" because it's pronounced the same. The spambot doesn't know what ee-meyl is, or any variant thereof. So he suggests for "subject" to do "suhb-jikt"; for "your name," "yohr neym" and so on. That's actually a clever idea.

Steve: I think that's why he won the clever idea award.

Leo: And there's no canonical phonetic spelling. So it's not like a spambot can learn all the different ways of doing it.

Steve: Yeah, no spambot's going to start going for phonetic interpretation of - it's going to do a string match on the alt tag to see if it can figure out how to fill out the form. But this is going to be like, you know, a phonetic version. I thought it was very clever because it nicely obscures it from the spambot while still being completely screen reader friendly.

Leo: Very clever. Well, we thank you for your good suggestion. We thank you all for your emails. And if people want to send you questions, it's not really email. There's a form, as we mentioned.

Steve: Yup. In fact, we tried email, and that was a disaster because the spammers found it immediately. So, yes, it's an online form at GRC.com/feedback.

Leo: All right. Just go there, fill it out. And of course there's a great security forum there, as well, where you can talk to other security experts. That's GRC.com. That's where you'll find SpinRite, everybody's favorite disk recovery and maintenance utility. It is the program you must have for all your disk drive needs. And lots of great stuff from Steve for free, including ShieldsUP! and all his free programs and his Perfect Paper Password generator. That's GRC.com/ppp, by the way. And if you go to GRC.com/securitynow, you'll find 16KB versions of this show for your friends with dialup connections, Elaine's great transcriptions, all the show notes, and a lot more. GRC.com. Steve, a great job once again.

Steve: Always a pleasure, Leo, and we'll be talking next week.

Copyright (c) 2006 by Steve Gibson and Leo Laporte. SOME RIGHTS RESERVED

This work is licensed for the good of the Internet Community under the Creative Commons License v2.5. See the following Web page for details:

http://creativecommons.org/licenses/by-nc-sa/2.5/