SECURITY NOW!

Transcript of Episode #120

Listener Feedback #29

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-120.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-120-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 120 for November 29, 2007: Your questions, Steve's answers.

It's time for Security Now!, everybody's favorite security podcast with Steve Gibson, the king of security. I don't know. You wouldn't style yourself the king of security, I know.

Steve Gibson: No, I wouldn't. I just, you know, I've been thinking about it for a while.

Leo: Thinking about being the king?

Steve: No, thinking about security and messing around with it and wrestling with it and all that kind of good stuff.

Leo: So this week we do our question-and-answer session, so we'll be talking about a lot of different security topics in just a little bit.

Steve: Yup, bunch of good questions from people.

Leo: We are prerecording because of last week's holiday, Thanksgiving, and I'm in Vancouver this week. So we don't have any errata from the previous episode. Not to say that it was perfect, but we just haven't had time to receive...

Steve: Right. We're recording this before anybody else has heard what last week's...

Leo: Right, right. And that'll be interesting. I'm sure we'll hear from PayPal if nobody else. Well, I at least hope we'll hear from PayPal.

Steve: That would be great. I'd love to hear some defense of them. It turns out, when I exchanged email with Elaine, our illustrious transcriptionist, turns out she was one of the people also who - she described herself as being bated breath, waiting to hear the complete analysis of that PayPal/DoubleClick relationship. And her comment was, well, you know, I could give up on PayPal and delete everything; but whatever they have, they already have, and I'm already in the galactic database. So what the hell.

Leo: What the hell. You're kind of stuck. What can you do?

Steve: Right, right. I did get an interesting and fun sort of SpinRite story that made me think of what we were talking about last week because we were talking about RAID and how, for stuff that absolutely, positively can't be lost, what you want is you want redundancy thanks to RAID. And of course backing up is absolutely what you want to do when you don't have the option of RAID, and even when you do have RAID. Because, for example, malware can still infect a RAID-based system, and then you've got much more reliable malware, even though that's not what you want, thanks to RAID. So...

Leo: More reliable malware. I never thought of that as a side effect, but you're absolutely right.

Steve: So it still makes sense to take checkpoints. But I got a really fun testimonial from a guy named Brian Kinder, who - it hit me because he is very computer savvy, and he recognized that this was sort of like the reverse of the RAID success. He starts off saying, "Just another success story of SpinRite." Just one more. Just another one. He says, "My boss has in his home a small workgroup network, and he was using an old Win2K box as data storage. He called me in a panic that he could not get to any of his 11,000 photos that he had taken over the course of six years from all of his many travels. I went over to his home and discovered that, of course, he has no backups."

Leo: Now, why would you have 11,000 photos and no backup?

Steve: Well, you know, I think this is one of the traps that people get into is that, well, it worked yesterday, and it worked today, so why wouldn't it work tomorrow? So anyway, so he says, "The previous 'technician' had set up his 2000 box with disk spanning. It had a 160-gig, a 30-gig, and an 80-gig disk set up in a span to form one single volume that was shared to the rest of his network."

Leo: Now, correct me if I'm wrong, but that makes it four times as likely to fail.

Steve: Well, three times because he's got three drives.

Leo: Three drives. So it's multiplied by each drive; right? Because if any one fails, the whole span fails.

Steve: Exactly. I mean, like the reverse of RAID. And of course it is only as strong as the weakest link, so you're multiplying the failure probabilities. So he says, "Well, you know as well as I..." and clearly Leo, he didn't say that, but I'm putting that in...

Leo: Thank you for including me.

Steve: "...that any drive in that group fails, the whole volume fails. So I run my trusty SpinRite and see what happens. It took about a day and a half, but all three drives finished the process. The 30-gigabyte drive in particular had many, many errors. But with your data recovery procedures, it was able to make it through. So now we boot up the unit with the three recovered drives, and of course the span volume comes up fine, and now he could get to the many thousands of photos he had stored. And of course we have to set up a more sensible drive redundancy and a backup procedure for him to follow. So again, thanks for a great program. I've been using it since Version 2, and it has never failed to get the results I need. Signed, Brian Kinder."

Leo: I hope that guy, that photographer fell on his knees and thanked you, because...

Steve: Oh. Well, and again, it's funny because we've seen hard drive storage just dropping in cost. And we've had some people write and say, hey, you know, I can get a drive for less than SpinRite, so why would I use SpinRite to recover a drive when I could just replace it for less? And it's like, well, okay, two things. First of all, you can only buy one drive for less than SpinRite, yet you get to use SpinRite on all the drives you have for as long as you have SpinRite. And secondly, of course, what we're seeing is, as a consequence of storage getting so cheap and digital media coming on so strong with photos and music and, you know, there are people who have huge servers, and they've ripped all their DVDs because they want to store them on hard drive for instant access or for access on various machines around the house. So we're obviously moving into a digital media world where the contents of the drive is what's valuable now, more than the container itself.

Leo: And, you know, if everybody backed up, they wouldn't need SpinRite.

Steve: True, if they backed up, like, constantly, you're right.

Leo: If they were religious about backups, you could just say, eh, that drive died, oh, well.

Steve: Although SpinRite, of course, does also keep drives alive longer. So it can help the drive to fix itself in order to, I mean, there are people who, after they have a problem, continue using the drives that SpinRite repaired for them.

Leo: I'm doing that right now.

Steve: Yeah, exactly.

Leo: And I feel fairly confident that whatever was the bad sector has been moved off. Although I should be pretty religious about backing up. Not merely backing up, but SpinRite-ing these drives, just to make sure.

Steve: Yeah, it's a good thing. And I did not answer your question last week when you said about how often. And of course - oh, there's Elaine sending something. She just responded to my sending the Q&A stuff.

Leo: She doesn't listen live, does she?

Steve: No, but we have never mentioned the fact that - or are we doing it this time?

Leo: We did last time. And we will probably - what we do is I stream, as I do my production day, my recording day - because we do most of the shows, obviously all the podcasts are recorded ahead of time. And I think people would like to listen live. So what I do is I keep a stream going at Ustream.tv of the recording. And then I have a chatroom going at irc.dslextreme.com in the #townsquare chatroom. So people - it is eventually going to be - my idea is eventually to be a little more live and interactive. But at least this is kind of a poor man's way to listen as we do it live. Not today, I haven't put it on today because I'm having trouble with Verizon. It's just not reliable enough. I have to get a second channel in here, probably get a T1, or I was thinking I could just get a cable modem and have the cable modem and the DSL, and that's kind of a nice set of redundancies.

Steve: Yeah.

Leo: Yeah. Anyway, so, yeah, she could be watching live, but she's not today.

Steve: No, not in this case. So anyway, I never answered your question last week about how often you should run SpinRite. And of course the answer is, well, run it before your drive fails.

Leo: Yeah, okay.

Steve: But my sense is that most people have years of life on their drive before they start having any problems. There are infant mortality problems, of course, that we've talked about in the past as one of the ways that drives can fail. But, you know, maybe three times a year, I would think that's often enough to allow SpinRite to help the drive realize that it's got problems and move any endangered data out of the way. It also works the drive. I mean, when people talk about it running for a day, that's serious data transfer that is occurring that also allows the SMART system in the drive to discover if there are any problems. So it sort of just gives it a real good shaking to make sure that not too many bits fall out.

Leo: Yeah. I like that. Shake your drive regularly to make sure no bits are falling.

Steve: Yeah, find the loose bits.

Leo: Find the loose bits. Before we get to our questions - and we have, let me see, looks like we have 12 excellent questions, including the Clever Observation of the Week Award.

Steve: Exactly.

Leo: That'll be fun. All right. Are you ready for some questions?

Steve: Absolutely.

Leo: Absolutely. I've got a dozen good ones, starting with Erik in Palo Alto. He wants to give his credit union some credit. Dear Steve and Leo, he writes, yesterday I got an email from my credit union, Patelco, telling me that they are now going to require multifactor authentication for customers who want to use online banking. Being the geek I am, I got all excited. Maybe they listen to Security Now! as well.

So I quickly went to their website and started reading about it, then signed up immediately for the beta test. The thing that surprised me was their method of multifactor authentication was a bit different than those I recall hearing you talk about. Their system works by having you authorize specific Internet service providers that you can log onto their system from. So he's with SBC. So when he logs on for the first time it says, ah, you're on SBC, I'm going to add that to your safe providers list. Note, you are not able to manually add ISPs. You actually have to log in from them. So you have to be on an SBC connection, and then they'll ask and allow you to add that to your providers list.

This morning when I got to work, logged onto the bank account again. After I entered my account number and password, I was taken to a screen that said you need to enter a new password, we're going to send it to you because the domain you're connecting from is not on the list. So he said, I could have the password emailed to me or have a text message sent to my cell phone - I like that, that's very convenient - since email addresses and cell phones are on file with them. So he chose email. Within a few seconds he got a four-character alphanumeric password that allowed him to access the account. The password expires after two hours. Kind of like a PPP. Well, not exactly, because yours you just use it and then it's done.

Steve: Right.

Leo: Now that I've logged in from here, my work's domain appears on the list of approved providers, and I don't have to go through that rigmarole again. However, should I want to always be prompted for an additional password - oh, he's asking, should I always ask this? I have the option to set it up that way. I was interested in hearing your comments on how good the system is and what flaws, if any, you see in it. Well, that's a simple way to do multifactor authentication.

Steve: Well, it's - yeah.

Leo: It's kind of weak authentication.

Steve: Yeah, I wouldn't quite call it multifactor. I guess there are - and I should be more up to speed on government regulations. But there are new regulations, I know, that are coming onstream that are requiring "multifactor authentication," that is, something more than just username and password. So I guess this would qualify, where obviously another factor is, in this case, the domain from which you're connecting. So...

Leo: It would work better if you were on a small Internet service provider.

Steve: Exactly.

Leo: If you're on AOL, that means anybody on AOL can attempt to hack you.

Steve: Exactly. What they're clearly doing is they're doing something that we've talked about in the past, a so-called "reverse DNS" lookup. Normal DNS, as we also know, takes you from a human readable style URL, like www.grc.com, and that translates it into GRC.com's IP address, which is the 32-bit destination where our servers live, which is shown as the four groups of numbers. I think GRC.com is 4.79.142.203 for the www.grc.com. And so that's what DNS does. Reverse DNS, like the name sounds, does the reverse. That is, you give it an IP address, and it can tell you what the machine name, the domain name is that's associated with it. So, for example, if you do a reverse DNS lookup on GRC's IP address, you'll get www.grc.com.

Now, we all know that DNS is a good and generally robust system, but not ever designed to be ultra secure. There are various types of DNS spoofs and poisoning attacks and so forth that can be done. Also, clearly this was designed to, for example, allow in your case, Leo, an AOL user, if they're like an older style dialup user, they're going to get a different IP address every time they dial into a large modem pool. And of course, as we know, even cable providers that are using DHCP to distribute their relatively limited IP pool around in their customers, you know, there is re-use of that. So that, for example, if you had your cable modem unplugged for a couple of days and then reconnected it, you'd very likely get a different IP address than you had before. If you keep the power on, and you've got a NAT router which is sort of anchoring that IP, they generally are more static, although Mark Thompson in Phoenix was telling me that he's now seeing, for whatever reason, his Cox networking provider is changing his IP, like, daily now, whereas others tend to stay much more static. But anyway, the point is that clearly they designed this to say, okay, we're not going to concern ourselves with the user's IP, which is changing. We're going to concern ourselves with the ISP's domain name.

Now, when you do a reverse lookup, for example on a Cox cable modem IP, you will see the - typically it's the IP address first, then sometimes they'll have something that's sort of like regional, like I remember sd.sd.cox.net - sd.sd happened to be San Diego in the case of someone who long ago was attacking GRC - or oc.oc for Orange County .cox.net. So the bank or other institution that's using this style would - they would ignore those earlier chunks of the DNS result, which might be and probably are based on the specific IP the user has at this time. But they go back further down in the domain name to say, okay, he's at cox.net. He's at aol.com. He's at...

Leo: I guess on big providers they could know what the range of IP addresses the provider - what Class C domains the provider owns; right? They could just add all of them.

Steve: Well, yeah. And in fact that's the problem is it's not one-to-one authentication.

Leo: Oh, I see, yeah, yeah. It's not your IP address because it can't be, because it changes.

Steve: Exactly. And as you said, anyone who is trying to spoof, for example, the "other factor" of an AOL user, well, if they were on AOL, then when they're spoofed, their spoofing attack comes in, and the reverse DNS is made, they're going to be on AOL, which might match one of the, quote, "safe providers" on that user's safe providers list. So the problem with this, and the reason I was skeptical about calling it really multifactor authentication is, well, yes, it's another factor, but it's not a one-to-one factor. For example, one of the cool things about Perfect Paper Passwords that we've talked about is you're the only one with this printed out little password token list. Nobody else has one. If other people had them, we'd be much less enthusiastic about it.

Leo: Right, right.

Steve: Because there'd be the opportunity for a collision there. So the fact that every one of them is unique is where it gets its strength. Similarly, if many people used the same password, even though their log-on was the same, but their password, like where there's a high incidence of password collision, that would weaken the system substantially. So, I mean, yes, this is better than not having it. But it's really not as secure as we would hope it would be. On the flipside, it offers more security than not having it at all. Some random attacker in another country would have a difficult time spoofing their IP into a specific safe provider. But even that's not impossible because the way reverse DNS works is you ask - essentially reverse DNS goes through a process of narrowing down the location of the ISP that the user is using until they come, essentially, to the ISP's DNS servers. Well, it would be entirely possible for someone located anywhere else to pretend to be SBC or AOL. That is, it's not AOL's servers that uniquely provide that reverse mapping. It's the server associated with that IP. So you could easily have some Russian or Chinese or anyone, essentially, who has control of that IP space could say that they're AOL.com. And there's no way to know that they're not. So again, it's a better thing than not doing this at all. But I'm skeptical about this thing really qualifying as a strong additional factor.

Leo: We should find out what the regulations are. It may be that they're not - I bet you anything the banking lobby said, well, don't make it too hard. You know, allow this.

Steve: Oh, yeah, if we make it too hard, no one will do it, so we want to make it, you know. And again, it's better than not having it. But it would really be better if they adopted a good multifactor solution as another factor.

Leo: I just want everybody to use these PIP cards, these VeriSign cards. It just seems like such a great way to do it.

Steve: Oh, it's a beautiful solution, Leo.

[Talking simultaneously]

Leo: ...carry the card around and...

Steve: Yup, I just signed the agreement, in fact yesterday, with VeriSign, an evaluation agreement that will give me access to their back end so I can actually play with the API. I mentioned a couple weeks ago that I had looked at it, and looked at the spec, and it looked absolutely clean and robust. So I'm going to enjoy writing some code to actually make it jump around and be able to authenticate my little credit cards. And I will certainly tell our listeners how that looks.

Leo: And you wouldn't really need PPP if this were universal. You could use this for your PPP; right?

Steve: That's absolutely right. And in fact I purchased six of those credit card form factor units, and I fully intend to actually use the VeriSign system as my primary authentication for myself and Sue and Greg...

Leo: Oh, no.

Steve: ...because it is just very cool.

Leo: As soon as you publicize PPP, you replace it.

Steve: I know. I know. But I'm glad I did PPP. And that'll always be our fallback. If for any reason I lose this relationship with VeriSign, then it'll be like, okay, fine, that was good while it lasted, and we'll be able to fall back to the Perfect Paper Passwords.

Leo: I'm sure VeriSign must intend to charge people who want to implement this; right?

Steve: Oh, absolutely. And their model is a big company model. I mean, I don't know if it would ever be feasible for a little ones-y guy like me to be involved. It's a PayPal and eBay and a BofA scale solution. But I agree with you completely, Leo. Based on what I've seen, I mean, I've got the card in my wallet. I was never carrying a football around with me because it was just - I didn't want to end up with a necklace of them, and it just wasn't so convenient. But I already have some credit cards, and this thing is no different than - as you know, you've got one. It's no different than a credit card.

Leo: You know what I'd really love to see is them - what happens with PayPal, you don't have your card, you can ask the security questions. It's always the same two, totally easy to fake this one. What I'd love to see is maybe them use this backend Bank of America's using where it says, okay, as this guy's did, now we need to send your code to your cell phone or your email to verify. I would prefer that as a second fallback position. Because you can't assume that people will always have the dongle or the card, or they won't have lost it or that kind of thing. So you always need kind of a back door to it. But let's make it a better back door.

Steve: No, I absolutely agree. And remember, as one of our fundamental lemmas of security is, if it's more secure, it's less convenient. I mean, it's a tradeoff. And so, yes, I'm now glad that my PayPal account is locked with these two hardware tokens. I've got both a football and my little credit card, both from VeriSign, hooked onto this account. And I'm really happy that I've done that. But I wasn't really happy until I had the credit card format token because that's easy for me to have with me all the time. There were several cases where I was at Starbucks and wanted to buy something, but the football was at home. I was going, eh, oh, darn.

Leo: So you were able to convert - you decertified the football and used the VeriSign identity protection card instead?

Steve: No, no, no. That's the cool thing is that VeriSign recently brought up, in the most recent version, which is now current, you can have five hardware tokens simultaneously registered to a VeriSign account, and PayPal supports that. So you can leave your football authenticated on PayPal right now and add your credit card to your PayPal account.

Leo: Through PayPal, not through VeriSign.

Steve: Through PayPal. There is a UI at PayPal, which is where I did it, and it knew that I could have up to five different hardware VIP tokens of different form factors, either the original football or the new credit card form factor, simultaneously on my account. So I've got both of them on mine.

Leo: Yeah, that's not a bad idea. I mean, then you have kind of a backup. If you lose one, you still have the other.

Steve: Well, and for example, I've got the football sitting here right next to me. And my wallet happens to be in the back of the house because I'm here in my sweats right now and not in my outdoor clothes. So it's perfect. It's like absolutely no overhead because now I can have multiple tokens simultaneously authenticatable.

Leo: Cool.

Steve: It works.

Leo: I'm sorry. We've now just made it, like, 15 questions because I just snuck in three. Sorry.

Steve: No problem.

Leo: Bob Gusek in High Point, North Carolina wonders, if once is good, is twice better? I've been thinking about encryption and brute-force attacks, says Bob, actually probably any kind of attack, and how they classify success in breaking some encryption. I'm not sure if there's something I'm missing. And I figured with the amount of work you've done on encryption, you'd be able to identify an issue with this approach. In the process of encrypting blocks of text that I may be sending to someone, in other words, encrypting a file, I've experimented with doing double encryption. What I do is take my clear text, encrypt it first with one key, then take this encrypted text, encrypt it with a second key. If someone should ever try to brute force this, from what I understand, they try combinations of keys until they get clear text. But of course they'll never get clear text because I've encrypted it twice. Am I missing something?

Steve: I think this was a really great question. So to paraphrase what Bob said, he recognizes, and we've mentioned this many times, that the way brute-force attacks work is essentially like with a dictionary or just starting at zero and counting up. You attempt to decrypt the encrypted text, waiting until something recognizable comes out of the decryption. And if suddenly you see something, chunks of text that is readable, you go, whoa, we just got the key because that's the only way that you're going to be able to decrypt the encrypted text is with the proper key to run the decryption. So he says, okay, what if I do it twice? Then no matter what anyone did, all you could ever get, even if you got the proper key for the second encryption, you would be taking that wrapper of second encryption off, and you'd have the first encryption. But that's going to be random because we know that what good encryption does is basically create a random mapping between so-called clear text and the cipher text. So you wouldn't be able to tell that you had ever gotten the second key by examining it as you would with a single encryption because all you're going to get is more noise, and it's going to be indistinguishable from any of the other random noise that you get when you don't have the right second key. So he's absolutely right.

However, I'll point out that this really would pertain only if you had weak keys. That is to say, all contemporary symmetric encryption, which is essentially what we're talking about, is going to use 128-bit key. Well, that 128 bits is phenomenal strength, given that the bits of the key are chosen randomly. They're not chosen randomly if, for example, you took the simple password "dog" and hashed it into 128 bits because somebody else could put dog into the hash, get those 128 bits, and apply that to the decryption, and essentially reverse your code. So what you could do is, if for some reason you needed to use weak passwords, then certainly using two of them would be, that is, exactly as he suggests, encrypt, and then use a different weak password to encrypt it again. Or for that matter, if no one knew that you had double encrypted, you could use the same password. Because, again, they're not going to know you double encrypted, so they're going to decrypt it once and see noise and go, okay, well, I got the wrong one, even though if they did it again they'd get the clear text out. So it's an interesting idea. But if you have, for example, a really random 128-bit encryption key, that's already so strong against any kind of brute-force attack that doing it twice doesn't really buy you any more because you've already got something that isn't going to be reversible in the first place. But there's nothing he's missing. It's a clever idea.

Leo: Why do I think that PGP uses double encryption? It's using 128 bits to encrypt the passphrase; right?

Steve: Well, correct. And PGP is a public key-based system, so it's got a much longer key, probably 124 bits, for the public side. And then but because you can't encrypt the bulk payload with public key because it's too slow, the idea is that PGP will generate a random number that is a random 128-bit key, or maybe it's longer, and then it'll just encrypt the key using the public key. So there are multiple layers of encryption. One is public key, and then one is symmetric key because symmetric key is high speed, and that's how you do your bulk encryption. And then you encrypt the key using just the public key side.

Leo: Right, right, right. Matthew Beacher in Pottstown, PA has also been having fun with crypto. Got a lot of crypto questions today. I've been a listener of the show since Episode 1, and I think the advanced topics have helped a lot. I'm writing a PHP-based web application. I'm currently working on the login system. As a matter of convenience, I'm foregoing SSL and instead engaging in client-side JavaScript one-way encryption. So here's what he's doing. I'm taking the user's supplied password, concatenating a known string from the GRC.com Perfect Passwords page - hmm, interesting - doing an MD5 hash, concatenating a second known string, and doing a SHA1 hash of that. Then I take...

Steve: He's not done yet.

Leo: Then I take this blob and do an HMAC SHA1 hash with a key from the PHP version of Perfect Paper Passwords, and I turn that into a hex string. All this in JavaScript. This is then compared with a string on the server that was hashed the same way. On the server I'm storing passwords hashed with MD5 and SHA1, then adding HMAC SHA1 hash to authenticate. I have three questions. I hope you're following this, Steve. One, am I doing enough to protect the passwords without using SSL? Two, should I be using HMAC - I don't even know what that is, H-M-A-C.

Steve: We'll talk about that in a SEC.

Leo: To hash the passwords before putting them into the database. Three, do I need to do something more to protect passwords when creating accounts for the first time, knowing that whenever or whatever the client sends will have to go into the database? Should I be doing an HMAC hash before sending the password, then a regular SHA1 hash before storing in the database? Folks, that's a lot of hash.

Steve: You know, I'll bet you that Matthew wears both a belt and suspenders.

Leo: This is a belt and suspenders system.

Steve: This thing, Matthew, nobody is ever going to figure out...

Leo: What you did.

Steve: ...by doing a monitoring of your wire between the client and server, what the password was that went in the front end of this grinder machine that you've built that manages to have, I mean, it's amazing to me that the same data comes out the other end each time you put the...

Leo: I don't understand how he synchronizes the Perfect Paper Passwords server and client-side. Not the PPP, but the GRC password.

Steve: Yeah. From what he said, I'm not sure what he's doing with PPP, how that fits in there. He mentions HMAC, and it's something we've never talked about.

Leo: I know what MD5 is, and SHA1. Those are both hashes; right?

Steve: Right. And we've talked about hashing, where basically you can put any size of stuff you want in one end, and what you get is a fixed-size sort of a fingerprint, the so-called "hash," of the input. And so, for example, MD5 stands for Message Digest. These hashes, another word for "hash" is a "digest."

Leo: And they're cryptographically strong. They're unique.

Steve: Yes. Newer ones stronger than older ones. For example, Perfect Paper Passwords is based on, or uses, SHA256, where you put a whole bunch of stuff in, and you get 256 bits out, which is stronger than MD5, which is 128, and stronger than SHA1, which is 160 bits. But still those are strong. And, for example, SHA1, remember there have been some - a flurry of concern because people were mistakenly saying that it had been cracked, or it had been broken. Well, that wasn't the case. What was found, however, is that it wasn't as perfect a hash as we would like. There were some collisions, meaning that to a higher degree than would be statistically likely, you could put different things in the front end and get a collision. That is to say, the same 156 bits out the other end. And so that made people a little uncomfortable about it, that is, it wasn't as completely strong and unpredictable and random or pseudorandom as we would like the output to be.

So what an HMAC is, HMAC stands for Hash Message Authentication Code. And that's like a hash, but it's keyed, meaning that in the same way that we have, for example, a symmetric encryption, like Rijndael is keyed where it will - you give it 128 bits with Rijndael, that is, and out comes a different 128 bits where there's a one-to-one mapping between the input 128 and the output 128, but that mapping is entirely determined by the Rijndael key, that is, the symmetric key. In the case of an HMAC, that is a keyed hash, meaning that it's a hash inasmuch as you can put as much stuff in as you want, but then you always get out the same size result, being a hash. But the key determines the result that comes out in the same way that a key on crypto determines the result that comes out.

So what this is useful for is, if we look at the use of a non-keyed hash, like SHA1 or MD5 or any of the non-keyed hashes, you put something in, and you get essentially a fingerprint, a hash of that document. So that's useful for saying, if you were to say to someone - and in fact this is how a lot of hashes on the web are used. You'll see, like, someone in the open source community will say, here's the source for this build of something, and here's the MD5 hash of it. That allows you to independently hash the same stuff that they did and compare the MD5 hash output to make sure that it's the same.

Leo: Evaluate it that way.

Steve: Exactly. Basically it's like a fingerprint, the hash is a fingerprint of what you feed in. But the point is that anyone can do this and get the same result. So what's unique about a keyed hash is that, not only can it verify that exactly what went in resulted in the proper output; but by being keyed, if the key is secret, then you can also verify who did it, that is, who hashed it, because somebody with a key had to have that key and that content in order to get the result. So it also allows authentication in addition to verification that something wasn't changed. It allows you to authenticate who provided the hash output, rather than just the fact that somebody did.

So anyway, Matthew has, again, I mean, he's come up with something very strong by taking the password, adding to it a big blob of randomness from our Perfect Passwords page, then doing an MD5 of that, adding another known string, doing an SHA1 hash of that, then using an HMAC where the key of the HMAC is one of the Perfect Paper Passwords keys, and then he turned that into a hex string, and that's what he sends to his server. So it's like, okay. So, wow. It strikes me as massive overkill. But he's obviously been having fun writing JavaScript, and so I think that's fine.

Leo: Whatever rocks your boat.

Steve: The big problem is that he says he's foregoing SSL, which means that he has absolutely strongly encrypted the logon aspect, that is, the logon phase. But apparently he's not using SSL.

Leo: So nothing else is encrypted.

Steve: Exactly. And then the problem or the question is, if he's creating a logon relationship, then state is being saved somewhere. For example, he may be returning a cookie to the client, which is the client's session cookie, which if no SSL is being used, it completely exposes him to a so-called sidejacking attack because anybody monitoring the wire, and of course the only reason he's gone through all these, jumped through all these hoops, is that he's protecting himself against somebody sniffing his traffic. Well, anybody sniffing the traffic would see everything else that's being done, including whatever he's doing to maintain state. Which if it's not wrapped in an SSL tunnel to make it snoop-proof, then somebody could grab the cookie, assuming that he's using cookies to maintain state, or whatever else, and instantly impersonate that logged-on user. So what he's done is he's protected the event of logging on, but he's made it unnecessary.

Leo: He's closed the barn door before the horse got away, but the horse wasn't ever in there.

Steve: The horse went out the window.

Leo: There you go. Interesting, yeah.

Steve: So, I mean, that's really the problem is he protected the event of logging on, but unfortunately the rest of his dialogue, the rest of the conversation, without being protected by a secure tunnel, everything he does from then on not only can be recorded and sniffed, but I don't see how he's protecting anybody from obtaining whatever ongoing credential was established by that one-time galactically secure logon. I mean, no one will ever figure out what that hex string should be, but now they don't have to. All they have to do is grab the cookie that gets returned, and they are able to impersonate the person who painfully logged on the first time.

Leo: Oh, well.

Steve: So, neat idea, Matthew, but I don't know, unless you are continuing to - he said client-side JavaScript one-way encryption, which to me meant he's going through all this to encrypt the logon, but nothing else.

Leo: You could encrypt the rest of the conversation by hand, I guess.

Steve: You certainly could. And that's what would be necessary in order to - and that gets pretty tricky, though, because you would need to do it in a way that was snoop-proof. And I guess my point is that that's all been done. That's all worked out with SSL. So rather than foregoing that, I think it's probably a good idea.

Leo: Now, Steve, you can't tell people to not reinvent the wheel. You like to do that yourself.

Steve: That's true. And so I would say that Matt has taken a great first step by nailing down the logon phase. Now he's got to, you know, if he still has any energy left after all that JavaScript coding of all the crypto.

Leo: Oh, that's a nightmare, too. But, I mean, I think that's why you do something like that. He's obviously a student, probably a computer science student. And you do that to learn all this stuff.

Steve: Well, I hope he's listening because he's probably going, oh, shoot, I didn't even think about that. I'm glad he asked the question.

Leo: It is a great question. Brian Dewey in Crestwood, Kentucky needs 89 security patches, and by now who knows.

Steve: Exactly.

Leo: 90, 91, 92? Steve, I've just been tasked to reload Windows XP on my father-in-law's computer. I'm concerned that Microsoft hasn't yet released Service Pack 3 for XP. That comes out early next year, I think.

Steve: Oh, well, it's in beta now. Is it going to be that - well, of course early next year is not that far away.

Leo: It's a couple months, yeah. Yeah, it is in beta now. I think they said 2008. I'll have to check. I've been searching the Internet and have located a few websites that offer third-party slipstreaming of the hotfixes. I personally only trust - boy, third-party slipstreaming. I only trust downloading my patches directly from Microsoft. What I haven't located is a reputable and authoritative list of hotfixes that are essential before I connect the computer to the Internet. Are you aware of a master list of redistributable hotfixes and associated URLs? Even though I've put this PC behind a NAT router, I'd feel safer if I could download and write to CD all the patches since August 2004. Am I safe if I only visit update.microsoft.com? Brian Dewey, SpinRite customer since Version 3.1.

Steve: First of all, I don't think they would fit on a CD. You'd probably need a DVD.

Leo: Well, let's see, the Service Pack 2 was 273 megabytes.

Steve: Yeah, and there have been some big ones since then. Okay, so this is an interesting question. It sounds like, first of all, he's concerned about putting a raw Windows machine on the Internet. And in fact, that's sort of an interesting, fun, controversial issue because he's right. If you were to reload XP and hook it without protection onto the Internet in order to get it updated, before it had a chance to get patched it would be taken over.

Leo: Unless you have the firewall turned on.

Steve: And that's exactly right, Leo. Now, I think he said, he says, even though I have put his

PC, meaning his father-in-law's PC, behind a NAT router, I would feel safer if I could download and write to CD all - and we know it's going to take a DVD probably - all the patches released since August 2004.

Leo: Let me just say one thing right away. If he only goes to Microsoft.com to get these updates, and he's got his firewall on, he's safe; right?

Steve: Yes. Either his firewall or behind a NAT router.

Leo: So as long as he doesn't go to malicious websites, he's not going to get anything else.

Steve: Well, there are two natures of attack. There are all the services that the original build of, what was it, the build number? I've forgotten now, used to know, the original build of XP had some funky, like, it was a round number. Oh, 2600, was that it?

Leo: I guarantee you it wasn't 2600.

Steve: I might be confusing it with another OS. But...

Leo: I think it was 3000 for Vista. But I can't remember what it was. Yeah, they tried to get a round number out.

[Talking simultaneously]

Steve: Yeah. So anyway, so the point is that that original build of XP, as we know, famously had a firewall that was turned off by default. And thus the reason that there were so many problems. And the malware, the worms, the viruses, the things that were able to infect those versions of XP, those initial problems that XP had are out on the Internet still sending packets at random all over the place, and we'll probably never get the Internet cleared of all that.

Leo: There's always going to be some Windows 98 machine that's sitting in a corner, collecting dust, that nobody's touched in years, it's spreading Sasser. That's all it does is spread Sasser.

Steve: So again, there are two types of attacks. There's the unsolicited attack which is packets from the outside coming in through no protection, hitting any of the many services that were vulnerable in that original build of XP. I mean, it just makes me shudder, you know, how much we've gone through since that original build of XP. Then, as you mentioned, the other class of attack are where you visit a web page that is taking advantage of problems that have now been known for years, but is hoping to get people who don't have Windows XP patched up to date.

So yes, Leo, if he were to only go to Microsoft.com, or I'm sure that that first version of XP had the Windows Update button in the Start Menu because I push it all the time when I'm setting up a brand new Windows XP system. So, I mean, and I guess I'm a good example. I'm behind strong NAT, and I put these machines on the 'Net, and the first thing I do is go to Windows Update, and then I go make a fresh pot of coffee because it's 89 and counting security patches, and you've got to reboot I think about five or six times now because the patches have patches, as we've discussed before. And then if you add any of the optional stuff, like the .NET stuff is still optional, then there's been a lot of security fixes for those, too.

So but there apparently is a place on Microsoft's site where they are listing patches that can be manually downloaded. The problem is, I don't know how current and up to date it is, and there really is nothing wrong, as long as you're behind a NAT router, or if you did not have a NAT router, just turning the built-in original XP firewall on will give you enough protection to get yourself patched and updated.

Leo: All right. And by the way, you can go to Windows Update, and if you go to the network administrator mode, you can download these updates standalone. So the default mode on Windows Update is to download and install and not save them. But you can change your Windows Update mode to a mode that will allow you to download those files. So you could go to another computer, get a list of all the hotfixes, which is what he was asking for, and then put those hotfixes on a CD or, as you said, a DVD or two.

Steve: And then march through them one by one and install them.

Leo: It's not a roll-up, and that's what's great about service pack, it's all rolled up into one. But at least you can do that. Now, I'm looking at Windows Update, and I can't remember - and I'm looking at the Vista version. I change settings - you have to change into a network, a different mode, a network - a sysadmin mode, basically. I'm not sure exactly where to do that. They've changed how this looks. But you can, I know you can do it. I've answered that question a few times on the show. So you can get those files individually on another computer. That would be another way to do it. A painful way to do it, but you could do it.

William Marquiss in Mt. Vernon, Washington can't find his NAT. Where's my NAT? I know it's here somewhere. Steve, help. I'm starting a small business. While shopping at the various stores, I can't seem to find no stinking NAT routers. Not one store employee seems to know what I'm talking about. I want to actually look at and hold what I purchase. Where are those NAT routers hiding? There's no NATs.

Steve: I love the question because I can see poor William walking into a Staples or Circuit City or something and saying "NAT," and they, huh? You want a what router? I want a NAT router. I've been listening to Security Now!, and Steve and Leo keep saying, you know, I want to be behind a NAT router. And the guy at the store says, well, we've got routers, but I don't think we have any NAT routers. I don't think our routers have NATs.

Leo: What is NATs? What are NATs?

Steve: Anyway, I loved the question. William, all routers are NAT routers. I guess I just say "NAT routers" because I'm a geek or a nerd or something. I don't think you could, I mean, it's not the case that, for example, high-end Cisco routers are NAT routers, although they have NAT capability built into them. But all of those little consumer boxes for $49, I don't think there's ever been one that wasn't a NAT router because the reason they were created was for IP sharing, which is what NAT is all about. NAT, of course, standing for Network Address Translation. So you can confidently go into whatever store you've been going into and confusing the poor salespeople and just say I want a router. And any of the consumer routers have NAT built in, and you'll be getting NAT protection because they've all got it.

Leo: Yeah. Just when it says a broadband router, that's what it is. Not a switch, a switch wouldn't have NAT necessarily, or a hub, or a bridge. But a router, that's what routing is.

Steve: Exactly.

Leo: Andy Leidy of Escondido, California has an interesting phishing idea for you, Steve. He says, you've done a great job discussing the benefits of using the VeriSign/PayPal security fob, a.k.a. "the football." I liked the recent suggestion about waiting for a prompt to enter the security number as an additional anti-phishing measure. But the system still seems vulnerable to a one-time phishing event if a user is duped into providing his code to a phishing site. Even if the bad guys can only use the code once, that one-time use could be pretty bad.

What if we turned the security code concept around and asked PayPal to provide a security code to us for verification they really are PayPal? Oh, I like this. When I enter my username, PayPal could display a code that should match the code on my fob within the next, say, 30 to 60 seconds. If it does match, I could then reply with my password and a subsequent code, feeling quite confident I wasn't being phished. Obviously PayPal would have to change their system to enable this, and there would be some issues with time syncing. But it seems like this would be a pretty strong anti-phishing method. Some users wouldn't be happy with the extra delay and would opt out, but others may be willing to take the extra time in order to increase their security. What do you think?

Steve: Well, you're right. I have to say it was really clever.

Leo: Very clever.

Steve: Just think about this. A very good friend of mine who is probably smiling as he's listening to us because he listens to Security Now!, had suggested this several times in email. And I've been just swamped and haven't gotten back to him about it. But he was suggesting it relative to Perfect Paper Passwords because he observed that the server in the PPP system knows the sequence of passcodes that are printed. So you could further strengthen the Perfect Paper Password system by having it first tell you what is the next code, and then you enter the one afterwards. That process would consume two Perfect Paper Password passcodes, but you would get some authentication that you were really talking to a server that knew where you were in the sequence and get some good anti-phishing protection. And in fact it's a little more direct with Perfect Paper Passwords because there isn't this issue of time.

Now, we could strengthen Andy's idea a little bit based on the revelation that we shared last week. I think it was last week, maybe the week before. Remember where another observant person who hangs out in our newsgroups realized that the first digit of the football, because the football is time based, the first digit is not random, it's sequential. It goes 0 through 9 and wraps around again, the reason being that if the clock has drifted, then when you give the six-digit code to PayPal, who of course forwards it through their backend server relationship to VeriSign where the actual verification is performed, that allows the server to deal with sort of plus or minus - let's see. If the code changes every 30 seconds, then that means in 10 of those, that's 300 seconds, which is going to be five minutes. So you could essentially handle a plus or minus 2.5-minute drift in the clock in the football and not harass the user by saying you've got to enter more codes. Anyway, my point is that we could strengthen Andy's idea and solve the time sync problem by having the server that is going to provide us with a code, instead we push the button.

Leo: We start.

Steve: Yes, we start, but only give it the first digit. So we push the button, give it the first digit. It should then definitively be able to tell us the other five because, by giving it the first digit, we've compensated for any sync loss that might have occurred since we used it before, as long as it's not more than 2.5 minutes, in which case we couldn't be sure of whether we were 2.5 minutes behind or 2.5 minutes before because we would have wrapped around. But if we give it the first digit within that window, it should be able to definitively give us the other five. And we would then know that we were talking to a site that knew who we were because we'd given it our username, it had looked up our football ID and figured out what should be on the display at that time. Then we would, in order to prove to it that we were really us, we'd have to wait for a 30-second boundary to occur where that first digit would change to the next incremental one, but we get a new five. And so we would then give it the next, essentially all six digits and say, okay, and we're convinced you're you, and here we are, we're us.

Leo: Right, right.

Steve: So it's a cool idea.

Leo: I like it. Of course the people who are most likely to get phished, I mean, in other words, if you're smart enough to have the thing, the football, to do all this, you're not going to get phished.

Steve: Right. Right.

Leo: The people who are going to get phished are grandma, my wife - who's not sophisticated. She's not going to do all this.

Steve: Right. But it's a neat idea. Now, here's the big problem with all of this, is that all of this assumes that you have an SSL connection, and you have verified that your certificate is HTTPS:, that is, you have an SSL connection, and you checked the certificate and verified that it's www.paypal.com and that the chain of authority goes back to VeriSign, from whom PayPal gets their certificates. In other words, none of this works due to the possibility of a man in the middle. If a site were phishing us, and we did not have a secure connection, then everything we've talked about, whether Andy's idea or my friend John's idea about the idea of the server telling you something first, then if there's somebody in the middle, then that person, that entity in the middle could simulate everything we've just talked about. That is, they would get the site from PayPal, show it to you. If you gave, for example, an enhanced version of Andy's idea, you give it the 0, you're really giving it to the phishing site, which turns around and gives the 0 to PayPal. PayPal sends it back the five digits to prove that it's PayPal, but it's not, it's the phishing site. I mean, PayPal is, sends it to the phishing site, the phishing site sends those five digits to you. So the problem is you have more strength in your belief that this is not a phishing site when in fact it is. And this is the problem with any of these approaches is...

Leo: The man in the middle. The man in the middle.

Steve: Yes. And so this would absolutely defeat a less clever, non-man-in-the-middle phishing site, and there are many of those that are just static websites pretending to be PayPal. But it isn't absolute proof that you don't have somebody in the middle. The absolute proof is that you've got an SSL connection, and you have taken the time to verify that you've got a certificate that was issued to PayPal and signed by VeriSign, in which case you're golden, and there's no one able to sniff what's going on between you and that site. But a neat idea. I really liked that idea.

Leo: Jay in New Hampshire poses a question which suggests some interesting issues. He says: Hi, Steve and Leo. I had quick question about a discussion I had with a friend the other day. I was telling my friend she should use a secure connection when she's using a public computer, since all you've got to do is put an "S" in the HTTP line. The question she asked was if the server she's currently using can decrypt the sessions at will. The server she's currently using can decrypt the sessions at will. We both work for the government and are well aware of Big Brother watching us. Oh, really.

Steve: Yeah, that was encouraging.

Leo: Hmm, hmm. Wonder what...

Steve: We work for Big Brother, and we realize Big Brother is watching us.

Leo: ...what branch they work for.

Steve: I wonder.

Leo: Which is why I said stick to personal time. But the question is valid: Is Big Brother watching us when we use HTTPS? You tell us, Jay.

Steve: Well, you're right that his question was a little ambiguous. I wasn't quite sure what he was talking about. But it brought up a couple interesting points. First of all, he was suggesting that you could always put an "S" after the HTTP.

Leo: Not true.

Steve: Exactly, to create HTTPS, which is not necessarily the case. It is, well, because in order for that to work, the server must support SSL connections. Meaning that it must have the SSL port open, which is not 80, which is what normal web browsing uses, but is 443 instead. So it must have that open. And it must have an SSL certificate signed by somebody who your browser trusts. And we've talked about this whole notion of certificate authorities where our browsers have a long list of authorities whose signatures they trust. So the problem is that these certificates are not free, and many smaller sites won't be spending whatever it is, $700 a year, in order to allow users to make secure connections. Typically it's only sites which have a need for a secure connection that have gone out of their way to make that possible.

So it's certainly not the case that any site you're visiting can accept an "S." And in some cases, for example, the machine at a given domain which can accept an "S," that is, the HTTPS and SSL connection, may be different than the normal web machine. For example, people who have been paying attention may have noticed sometimes it'll say secure.domain.com, meaning that that's a machine that's able to do secure connections. It'll say that in addition to HTTPS. But if they normally go to just a non-HTTPS connection, there'll be, like, www.domain.com. So there's that.

But essentially her question was if the server she's currently using can decrypt the sessions at will. I think what Jay was asking was whether - it sounds like she was doing something of a personal nature at work, not on her personal time. And he was suggesting that somebody like local IT could be monitoring her web traffic if she was not using an SSL connection. So of course he's been listening to Security Now!. Obviously he's a listener because he sent this to us. So he knows that if she uses SSL, it'll create a secure connection to the remote server. And of course that server decrypts it. So the idea is that - I think what Jay was suggesting was that using a secure connection she could get out past the local network watchers, out onto the Internet, and to a remote server, where it would be decrypted. And the point is that SSL is a point-to-point security, that is, a point-to-point encryption. It encrypts it at your browser, and it decrypts it at its destination, whatever that is. But at that point it's back in the clear. So the server can certainly decrypt it. It has to, in order to be able to provide the services that she's asking for through that encrypted tunnel.

Leo: So in other words, yes, right. So in other words you can't always use "S"; and even if you did, you couldn't always be safe.

Steve: Right.

Leo: Unless you use VPN. Or something. That's why it's unclear. Did he mean the server side? Because if he means the server side, of course it's unencrypted then, otherwise they wouldn't be able to deal with you.

Steve: Yeah, I think - he says we both work for the government and are well aware of Big Brother watching us, which is why I said stick to personal time.

Leo: Yeah. So he means if you use it at work, can they see what you're doing, and the answer is yes, as we said before.

Steve: Exactly. And so by all means, if that's a concern, try to add the "S," and you will get a secure connection if the far side is able to accept one.

Leo: Right. Dusan Maletic in Babylon, New York solves PC tracking mystery. It comes as...

Steve: That was well done, Leo.

Leo: It came as a surprise to me that the last question addressed in Episode 118, while partially answered, did not lead to the discussion about Flash cookies, particularly as I first learned about them during an earlier Security Now! episode - hey, we know so much that we forget things - and that info provided then answered identical questions I've had in the past. It turns out that three out of three financial institutions I use online plant Flash cookies - wow - to track users' status, including BofA. I hope many other listeners alert you to this, leading to a good discussion of such semi-hidden techniques which are important for computer security in general. I'm particularly angered by this practice as the designers obviously have chosen an object poorly understood, if at all known to most of the public, and have not disclosed it to the users in clear manner. Well, fooled me. Typical spyware-like methods deserving critique and raised alertness to it. So that's interesting because remember I went through this whole rigmarole where I turned off cookies and stuff and so forth and so on, and it needed to be - I determined it was cookies. But maybe it is Flash cookies.

Steve: Well, and maybe - you and I talked about this, we've talked about it before here. We also did a whole session on it when you and I were in Toronto a couple years ago and showed the viewers of your Call For Help show where to go and how to turn these off.

Leo: Which I'd completely forgotten.

Steve: So I wanted to mention that to everyone who's listening because many people wrote in having done this experiment. They deleted their cookies, they emptied their browser cache, they shut down their browser, they rebooted their computer, they took their laptop to somewhere else, and they were - and literally at least 40 people wrote in and said, "It still knew me. How did it know me?" And so I appreciated this confirmation that this use of Flash cookies is becoming more widespread, clearly in this case, as he says three out of the three financial institutions he used plant Flash cookies.

So to all listeners, into Google you want to put "Flash player settings manager." Just put in "Flash player settings manager," and you get a link to Macromedia, maybe it says Adobe now, I'm not sure, I don't remember whether they've changed the URL. But the point is, most of us have Flash loaded in our machines now, which unfortunately is why the banks have all started using it. It's something that survives, as many listeners have discovered, it survives casual cookie deletion. And exactly as this guy has mentioned, it annoys him because it is unknown and is unclear.

The good news is, it's possible to control these settings and to prevent sites from using Flash cookies if for some reason you really didn't want that, or to restrict sites that you have specifically allowed. Anyway, there's good Flash cookie management available, and it's a web-based interface. You don't use your local Flash player, running it like standalone, because it is an embedded web page object. Instead, if you put in "Flash player settings manager," that'll take you to the Flash site, where you're then able to go to some web pages to bring up a little tabbed interface. Basically it runs your Flash player on the page and gives you access to a user interface you never knew you had. And you're able to browse through and see the domains that have registered cookies on your machine. You can delete them right there. You're able to change settings. You're able to do some worrisome things, like you can tell it don't ever turn on my microphone and camera without letting me know. It's like, okay, well, that's probably a good thing to tell it. So you're able to do that and a number of other things.

So again, "Flash player settings manager," and poke around in there. You'll find out who has stored cookies, so you know. You're able to delete them. You're able to then block them and prevent them from changing. Anyway, there's a whole bunch of tabs and settings that are definitely worth poking around in.

Leo: I don't see Bank of America in my cookies, however, so I don't know. Maybe I'm special. And wouldn't you need to see - wouldn't you see somewhere that Flash was running?

Steve: You don't see it. It's completely done behind the scenes using JavaScript.

Leo: So you can uncheck the box that says allow third-party Flash content to store data on your computer. It doesn't - JavaScript doesn't even have Flash going. Wow, that's interesting.

Steve: Now, is there a chance you would have changed these settings in the past?

Leo: Yes. Oh, of course, I had set it storage to zero. But there's more than that. You also probably want to deny all cookies and so forth. But then you have the same problem denying cookies on a browser, as well, which is that some sites don't like it. I see, I'm looking at the sites that have placed cookies on, you know, visited websites. And they're all, you know, they're mostly sites that do Flash media in one way or the other, like YouTube and Blip TV, Ustream. I don't see my bank. So I don't - anyway, I don't know. Twitter uses it for some reason. That's interesting.

Steve: My guess is that the banks probably issue standard cookies and Flash cookies. They probably just throw as much state...

Leo: As they can.

Steve: I'm sure they do. They throw as much state at you as they can, and anything they get back helps them to recognize you.

Leo: Right, which is fine. And in that case I want them to have some sort of way to recognize me.

Steve: Agreed.

Leo: Interesting. Thank you, Dusan, for that. Now, where did the questions go? Oh, here they are.

Steve: They got buried under your Flash browser.

Leo: 800 pages of browsers. Dan in Needmore, Pennsylvania needs more help with his PayPal token: Hi, Steve. Just heard you mention the first number on the PayPal security fob increments sequentially. I tried mine; it doesn't. First digit was a two, second a five, and third a nine.

Steve: And I'm reading it, and I'm thinking, oh, no.

Leo: What?

Steve: And then he said...

Leo: These attempts were several minutes apart. So you have to do it one after the other.

Steve: Exactly. I wanted Dan to know, to put his mind at rest, that the digits are changing, even if you're not seeing them.

Leo: Every 30 seconds.

Steve: Because - exactly. Because the football, as we fondly refer to it, unlike the credit card, the VeriSign credit card, the PayPal/eBay/VeriSign football is clock based. So those digits are changing, cycling 0 through 9, the first digit, 0 through 9 and back again as we were talking about before, whether you're displaying them or not. So...

Leo: So you just have to keep pushing it every few seconds to really...

Steve: And catch it across those 30 second boundaries. And then you will absolutely see this thing incrementing sequentially.

Leo: Yeah, makes sense. James Kilner in Israel has a correction, and then needs a suggestion. Steve, Leo, you're wrong about the use of the Firefox master password. I was listening to you talk on the Tech Guy radio show, #403, podcast version, about Firefox's master password. What caught my ear in particular was you claimed that this password can be subjected to a brute-force attack, so you should make it long and make sure it uses different types of characters. You then said you'll only ever need to type it in when you want to view all of the stored passwords. That's wrong. No, no, you didn't say that. I would have corrected you on that.

Steve: Yeah, I didn't think I said that. But I thought, oops, if I did, then we want to make that a correction for the record.

Leo: I enter it every time I launch Firefox, or actually every time I launch Firefox and then go to a page that needs a password.

Steve: So does this guy.

Leo: Yeah. I use Firefox to store my passwords, for example, for Gmail, so I don't have to remember them all the time. That's the point of being able to have Firefox remember your passwords. When I start up Firefox in the morning and load up Gmail or any other site that has a password stored, I have to type in the master password. Right. Me, too. If I have to close down Firefox for any reason, for example, it's leaking memory again, and then restart it, I have to put in the master password again. So it's not as simple as you suggest to just make the password something really long and difficult to brute force because you'll only have to use it to view all the passwords. That's not the case. Can you suggest a way to store a long password securely so I can copy and paste it in on a regular basis?

Steve: And that's why I really put the question up here, because I first wanted to correct the record, if I had said that you only needed to type it in in order to see all your other passwords.

But also because he asked an interesting question. How could he have, for example, one of those nasty passwords from GRC's Perfect Passwords page, and store it on his computer, yet still use it, because he wouldn't want someone to discover that. Well, one thing you could do is you could look at that wacky-looking thing, which is impossible to memorize or do anything with, and find something in it that is memorable as a split point. And when you - so essentially you store it that way, but you enter it in a different order. You take that split point and copy, like, the last X characters of it. Maybe it's got, like, an exclamation point pound sign, and that's memorable to you. So you do it from the pound sign to the end and paste that in, then do it from the front of that wacky password to the exclamation point, that is, do the front portion, and then paste that in second. So essentially you've taken - you've cut it at some point, and you reversed those pieces.

Well, that means that anyone who got that password from your computer first of all has no idea that you're a person who does this, nor whether you chopped it in one place or in two places. And if they tried to use it with Firefox as is, it wouldn't work. You have some way of mutating it so that it does. Or you could just add a couple of your own characters to the end of it or in the middle or wherever. So you take something like that, which is already absolutely strong against brute force, and change it some way.

Leo: Yeah. I just, you know, I just have one that's hard to brute force but easy to remember. It's possible to create those, as well.

Steve: Yeah, for example, many people, after we were originally talking about passwords, said what about the first letters of the lyrics of a song that we like.

Leo: Right.

Steve: It's like, yeah, there's a good source of a pseudorandom character stream which nobody else would be able to guess, but which you can, by running the lyrics through your mind and typing the characters as you say them, you can reproduce the password.

Leo: You create the string algorithmically, and you can remember the algorithm because it's simple. Another one I've heard before, but don't use, is you do the first initials of the last - the initials of the last name of the 10 presidents, capitalizing them if they're Republican. Something like that, so you know.

Steve: Well, that would lock me out. I'd have to figure out...

Leo: You'd have to remember it.

Steve: I'd have to get a paper and pencil.

Leo: You start with Nixon. So capital N, go to capital F, and then a lowercase c for Carter, and then a capital R for Reagan, and then a lowercase c for Clinton. Oh, no, I left out a Bush in there somewhere. There's a B, and then a lowercase c - capital B, lowercase c. So that would work. You'd have to...

Steve: And the problem is, if this election goes the way we're thinking, it's always going to end up with a BcBc.

Leo: BcBcBcBc. That's a good point. Didn't think of that one. Let us move on to question 11, and then we're going to take a break before our special question, our brilliant idea of the month. But first, Jeff in Manila in the Philippines and many others have been paying close attention: Steve and Leo, quick question regarding the Buffalo wireless router that Leo and Paul discussed on our last episode, actually two episodes ago, of Windows Weekly, the Zune episode. Leo suggested getting the Buffalo router since it supports both WEP and WPA because the Zune 30, the old Zune, doesn't support WPA. From my understanding of previous Security Now! episodes, WEP is just as good as not employing any sort of protection on your network. Wouldn't the WEP part of it be a vector for attack and eventually compromise your network, even if you had WPA on?

Steve: And I think he's probably right.

Leo: No, no, because these routers are designed to segregate the two.

Steve: They are for sure?

Leo: Pretty sure.

Steve: Okay. Well, I didn't know. And so many people also wrote in about this. Many people said, wait a minute, you know...

Leo: You'd want to make sure, obviously, that the WEP network was as if it were a separate network, didn't give you access to the full network, just to the Internet.

Steve: And that's my problem, is that I would think, I mean, we've talked about relatively sophisticated active attacks, for example, ARP spoofing, where if you were able to get access to WiFi, then you could fool the other machines on the network into believing that your machine was the gateway and route all the traffic through you. So I'm worried that the WEP leg might just be on a switch in the same way that the WPA leg would be, and that they may...

Leo: That would be sufficient, huh.

Steve: ...really be - yeah. So I think I'm going to have to pursue this because, I mean, so many people were interested that it's worth me tracking down a Buffalo router that has this and do a little research because...

Leo: There are a couple of routers designed for this situation, where you have some devices that are WEP only. And I just assumed they did it the right way, but that was probably a mistake. I should probably find out.

Steve: In any event, it's certainly the case that you wouldn't want to leave WEP on all the time. So unfortunately somebody who's a Zune user, like a first-generation Zune, who needs WEP all the time, would tend to have it on all the time. I was originally discussing this as a solution for what if your friends came over, and they had a laptop that didn't support WEP, how would you let them in without having to give them your precious WPA password? It's like, okay, you can have it on for a while, that reduces your window of exposure. But again from an absolutely how-strong-can-we-be security standpoint, all the people who wrote in about this question are correct because, if the router allowed the WEP network to touch the WPA network, all bets are off.

Leo: Right. They have this One-Touch Secure System.

Steve: Right. That's I think their acronym and their...

[Talking simultaneously]

Leo: Yeah. I'm reading the specs, trying to figure out if it talks about that, but it's not clear. When two or more AOSS clients attach to the AOSS network, the client router automatically negotiates the highest level of security the router can support. So if one supports WEP and the other WPA, it automatically adjusts the security to a level both clients support. See, that's not it, then.

Steve: And in fact, what that sort of says is that it will lower the security of your network to the lowest encryption that any given device is able to have.

Leo: Right, lowest common denominator. But that stinks.

Steve: And Leo, it must be in fact that these networks are together because who would, you know, if you Zuned, if you use Zune over WEP to get to your router, the point is you want to then get to your server where all your Zune music is stored.

Leo: See, I was thinking it was - ah, right. I was thinking it was like having two routers, where one was set for WEP, and that was then bridging to a WPA router.

Steve: It would be nice if it were so. But to me it sounds like it's not because, you know, you're going to want to be able to get to any other machine in your network.

Leo: Right. Well, if you want to do that, you're right, you're absolutely right. If the Zune needs to then get to a PC, you couldn't do it that way. I was just thinking, oh, it gives something like a Nintendo DS access to the Internet without giving it access to the subnet, but apparently that's not the case. It sounds like it lowers, no, I'm reading the specs, it says specifically it lowers the security of the network to match the new thing.

Steve: Well, I'm sure on a device-by-device level, so that - and that's really the cool part of this is that a given laptop...

Leo: Yeah, because it couldn't change the WPA. If it's a WPA laptop, it's not going to change it all of a sudden to WEP.

Steve: Exactly, exactly. So it's able to establish individual encryption links at the highest encryption level that each device can handle. The problem is, you bring one WEP device in...

Leo: That's a vector.

Steve: Exactly.

Leo: An attack vector. All right. Coming up in just a bit, Ernie Moreau from Kelowna, B.C., with a Clever Observation of the Week Award. Are you ready for the last question?

Steve: Absolutely.

Leo: The last question.

Steve: The last observation.

Leo: It's not even a question.

Steve: Not even a question.

Leo: Ernie Moreau in Kelowna, B.C. wins the Clever Observation of the Week Award. He says: In Episode 118 James Earl Ford from Apple Valley mentioned the following: BioPassword offers the only multifactor authentication software that combines a user's login credential, you know, the login and ID and password - login, ID, and password - with the behavioral biometric of keystroke dynamics, that is, your unique typing rhythm. I just wanted to mention, this sounds great until keystroke loggers become more sophisticated and also log the timing. Then you're hooped. Good point.

Steve: That's right. We are seeing serious evolution of technology in keystroke loggers. I've been noticing some reports out on the web that this notion, remember we talked about keyboards, clicking on keys and having the keyboards jump around the screen so that the coordinates were different? Well, it is absolutely the case that keystroke loggers are now being found on machines which take snapshots of the screen and are specifically designed to literally do as good a job as the designers can of capturing the login experience. And so if they don't already log timing, and if timing with this biometric keystroke dynamics, which I'm still sort of skeptical about anyway, if that ended up being a popular thing to do, well, they're certainly capable of logging the timing on the client side and reproducing the timing when they want to pretend to be somebody that they're not.

Leo: Well, there you go.

Steve: I just thought that was a clever observation. And Ernie wins the Clever Observation of the Week Award.

Leo: Now, how do people, if they want to ask questions for next time or make suggestions or make clever observations, how do they do that? I forgot. I have no idea.

Steve: No, it's GRC.com/feedback.

Leo: There you go.

Steve: That'll take you to a page where there's a form, you fill it out, and it comes to me directly.

Leo: And of course GRC.com is a great site to remember for a lot of things: Steve's vast array of free security software, including ShieldsUP, the most trusted firewall test, absolutely free. You get all kinds of programs, including demos of his PPP program, Perfect Paper Passwords. It's all at GRC.com. You also can get 16KB versions of this show. That way you can share it with your bandwidth-impaired friends. And transcripts, which make it a little easier to follow. You can read along or get in bed and read. Put it on your Kindle.

Steve: Yes, in fact we should mention that you and I have both ordered the new Amazon Kindle.

Leo: I resisted.

Steve: Yeah, you tried. I didn't. I just went right for it because I've got to mess around with this thing, see what it's like, how it works. So we'll probably have some impressions to share in two weeks when we record our next episode.

Leo: Not even that long. Who knows, maybe even next week. You know they're back-ordered.

Steve: No kidding.

Leo: Oh, yeah. I'm not going to be able to get mine until early December.

Steve: Yay. Oh, I'm sorry, don't...

Leo: Did you just say yay?

Steve: Well, no. What I was going to say was I got confirmation that mine has been shipped, so...

Leo: Yeah, I should have ordered it sooner. And I know Amazon, that they often under-promise and over-deliver. So they say December 3rd, but I may have it before the next, well, December 3rd would still be before the next episode. You know why I succumbed? I don't like the form factor. I have a Sony, the newest Sony Reader, as you do.

Steve: Well, it's the ugliest looking thing I've ever seen, Leo.

Leo: Yeah. Same screen, too, although it isn't exactly the same. Fewer shades of gray may make it crisper, I don't know. I'll be interested.

Steve: That's a little worrisome, actually, because the original PRS-500, the first Sony Reader, it had four shades of gray. I hope that this thing didn't get locked in with the previous Sony style screen, or we're not going to be happy.

Leo: No, we will not.

Steve: Because the 505 is a much better screen on the new Sony. And I read with it every morning.

Leo: The thing that's put me over the top, I get several newspapers. I get The Wall Street Journal, The New York Times, the San Francisco Chronicle. That's a lot of papers stacking up.

Steve: You stay so connected.

Leo: Well, that's part of my job. But it stacks up, and I'm killing a lot of trees. And you can get all those cheaper on the Kindle, automatically delivered every morning.

Steve: I just hate the ink rubbing off on my hands with newsprint.

Leo: Well, there's things I could do with newsprint, like clip things, and I don't know if I'll be able to do those with a Kindle. I'm very interested if I can somehow bookmark or somehow clip this content. You know you can email me your PDFs from now on on the Kindle?

Steve: Ooh. Although I get charged 10 cents, I think. And I don't think PDFs transfer very well.

Leo: Say Word documents. And so maybe they don't want to send PDFs. You have to Mobi-ize them.

Steve: Exactly, exactly.

Leo: I subscribe to a couple of newspapers - Salon, which I'm a premium subscriber to anyway. Just because it would be nice to be able to consume that content. I think the wireless is the key. I may use the Sony for books and use this for ephemera, for content.

Steve: Well, again, the idea of wireless where you subscribe to blogs or newspapers or something, and it just all is in there, I think that's very cool. And the fact that you have - they did it right from a hardware standpoint. They're over on Sprint's network with EVDO, which is much better bandwidth than unfortunately the iPhone, which is over with AT&T and the Edge network, which does not have the bandwidth performance that EVDO...

Leo: Although these files aren't that big.

Steve: True, in fact that's absolutely the case.

Leo: It doesn't really matter. But anyway, they did a deal with Sprint. And there's no extra charge for that, which I find interesting. I guess it's built into everything you buy because everything you buy is copy protected and costs something.

Steve: Yeah, well, I also bought the same book, both for the Sony 505 and for the Kindle, so that I'm able - I'll be able to do some side-by-side comparison of how the books look.

Leo: You are dedicated.

Steve: So, yeah, well, I just love - I love the eInk format. I've been an eBook user forever.

Leo: Yeah, we're early adopters on this. I mean, I know most people - somebody told me, I think Scott Bourne told me that the Sony eBook Reader has only sold in the tens of thousands of units. It's not a huge seller.

Steve: Yeah, and I'm not surprised.

Leo: No, it's a specialty. And I wonder if Amazon, I mean, it's gotten so much attention for the Kindle, I wonder if it'll break through. I have a feeling it will not. It's the kind of thing I think that's going to be gathering dust on a lot of people's shelves. But you and I both do use these eBook readers, so if it works for us - anyway, we'll have a review next week. At least you will. Mine might not be here yet.

Steve: I'll wait for you, Leo.

Leo: No. I don't mind. GRC.com is also the home of SpinRite, everybody's favorite hard drive recovery and maintenance utility. It is a must-have. If you've got hard drives, you need SpinRite. GRC.com. Steve, have a great week. We'll be back next week. Who knows what we're going to be talking about? It's a mystery. Do you know?

Steve: To me, too. Oh, I have a list here of topics we'll be getting to very soon, so I'll choose one.

Leo: We'll choose one. Maybe it'll be the Kindle, who knows. We could talk about the DRM they're using. That might be an interesting security issue. And also how soon before I start getting spam at my Kindle address.

Steve: Oh, yes. How soon before the Kindles get hacked.

Leo: That'll be interesting. Thank you, Steve. We'll talk again next week.