Listener Feedback #174

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

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-419.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-419-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve has questions and answers, 10 of them. He'll also talk about the latest security news. Stand by. Security Now! is next.

Leo Laporte: It's time for Security Now! with Steve Gibson, Episode 419, recorded August 28th, 2013: Your questions, Steve's answers, #174.

It's time for Security Now!, the show that protects you, your loved ones, your privacy online and off. And I say "off" because we do some of that, too. He's Mr. Steve Gibson, our Explainer in Chief here, from GRC.com, the man who created ShieldsUP!, SpinRite, the world's best hard drive maintenance utility, and of course first discovered spyware, coined the term, and wrote the first, believe it or not, antispyware program, lo these many years ago.

Steve Gibson: Ah, yes. I used to have hair back then, Leo.

Leo: And here we are in year nine of the Security Now! show. That's a long time for a show to be going. That's great.

Steve: Now, the reason it continues, though, is that we don't take ourselves too seriously.

Leo: No.

Steve: Which I'm going to prove with the first item here on our list to discuss.

Leo: This is a Q&A episode.

Steve: It is.

Leo: So we'll get a little news in.

Steve: We first need to discuss de-waxing ears.

Leo: All yours, sir [laughing]. I actually have some stories to tell along those lines, but go ahead.

Steve: Well, that's why we're discussing it, because I heard you on TWiT talking about how you are - this was on Sunday - you had been declined for having your custom ear pieces fitted.

Leo: My ear molds. The audiologist said you've got to clean your ears first. And she said, "I don't do earwax removal. You've got to go to the doctors to do that."

Steve: Well, not true. That's why I wanted to put this at the top of the show, Leo.

Leo: Oh, good. Thank you.

Steve: I have the solution for your earwax...

Leo: Of course you do.

Steve: ...removal needs.

Leo: No matter what it is, Steve will - this is what I like about geeks, particularly people like Steve. They'll do the research. They'll find the best. Doesn't matter. Could be a projection TV. It could be earwax removal. It doesn't...

Steve: And you can bring this up on the screen: www.EarClear.com.

Leo: You know, the Giz Wiz had some verkakte ear vacuum that he tested. I couldn't do...

Steve: Yeah, those are ridiculous. So this was maybe - this was after I had kicked in to my "let's take ourselves seriously" about health and supplements and all that stuff that I've been doing now for eight years. And always, my entire life, my right ear tended to collect wax.

Leo: Yeah. It's a genetic thing. It's actually a marker. 23andMe discusses this. There are two markers for earwax. There's wet and flaky, wet and dry. And you and I probably have the wet marker.

Steve: I think that's probably true. What I will tell you is the idea of using a syringe absolutely works. It is doctor approved. That's what the doctor does.

Leo: Now, on this page it looks like you're shooting Coca Cola into your ear. That's probably not the case.

Steve: No, he put the Coke can - he's an interesting guy. He actually called me on the phone when I ordered some of these. So I think it's a rather small operation. I don't think he's - and I don't remember whether there was a PayPal button at the time. His is very expensive. His is, like, $40. But there's one for $6.49, that first link in the show notes, HealthEnterprises.com, earwax removal syringe. Anyway, the point is...

Leo: Can't you just get a bulb or something to do that? I mean...

Steve: Well, you want enough volume - so let me just get this out, Leo.

Leo: Oh, get it out, get it out.

Steve: You do this in the shower. You have your syringe there...

Leo: It can be messy.

Steve: ...in the shower. And so you pull the plunger out, fill the little tube up, put it in at the top, and just squirt a syringe worth of warm shower water into each ear. I have turned several friends onto this, in addition to myself, and they cannot believe it. I mean, they were doing Q-Tips and coat hangers and...

Leo: Oh, don't do coat hangers. Holy moly.

Steve: It was bad.

Leo: That would be a bad idea.

Steve: It was bad. And so this is, I mean, this is the way you solve the problem. And so, if you do this just when you do your daily shower, you just squirt one syringe worth into each ear, in a couple days they are just - it's amazing how effective it is. And I wouldn't be sharing this if it didn't absolutely work.

Leo: Thank you.

Steve: And just solve the problem once and for all.

Leo: Because of you, I'm going to get my in-ear molds made.

Steve: Yeah.

Leo: That's pretty funny. They also, by the way, the same company makes vinyl eye patches and pill box - pill splitters. So it's a good company. They're in the biz.

Steve: The EarClear.com folks?

Leo: Yeah. Acu-Life, yeah. No, no, no, the Acu-Life - Health Enterprises. The $6.49 one.

Steve: Okay, yeah. I went for the $35 or $40...

Leo: You went for the expensive one, huh?

Steve: Yeah, well, it's actually the first one that I saw. I don't how because Amazon has, like, a million of them. You put in, like, "ear syringe," you just get a bazillion of them. Somehow I found this guy.

Leo: The $40 one.

Steve: Yeah, he was - he took himself very seriously, too, on the phone. He was like - and it comes with - well, in fact, here's mine. I had mine...

Leo: Holy cow. That thing's got a lot of volume.

Steve: Yeah. It's just, yeah, nice little pop. And so you just fill it up and then squirt it into your ear. And he talks about how it's got this little bent angle so that you can't go too deep.

Leo: That's good. You don't want to do that. Yeah, you could injure yourself.

Steve: And he's got his little set of instructions it comes with. And, yeah, anyway, all I'm saying is, that's the solution. And it works perfectly. It just ends the issue for anyone who has an issue with earwax.

Leo: Okay.

Steve: Okay. Actually, we're going to do some security, Leo.

Leo: Good.

Steve: Yeah. On the subject, the category of "that didn't take long," we have a huge lineup of, like, everybody being shaken out of the rafters who have secure communication solutions to the NSA dilemma.

Leo: Oh, god. You know, I got some emails from people talking about nyms and all sorts of things.

Steve: Yeah. And so the phase we're in at the moment is the everybody rushing to capitalize and cash in on the current frenzy over security. That'll pass, and we'll end up with the right solution, or a couple of them. At the moment, everybody's jumping in. So of course Kim Dotcom...

Leo: Of Mega fame, yeah.

Steve: ...of Mega fame, who's - he's in New Zealand; right?

Leo: Yes.

Steve: And he's not happy with the New Zealand government because they're beginning to make some noises about changing regulations to thwart his ambitions. Anyway, so his deal is he's going to fill in the - fill the shoes of Lavabit.

Leo: Yes. Of course he is.

Steve: Uh-huh.

Leo: Being the humanitarian that he is.

Steve: Yes, exactly. So Vikram Kumar, who is with Mega, told ZDNet that the company was being asked - was being asked - to deliver secure email and voice services. We don't know who's asking, but somebody, apparently. He says...

Leo: Please, please, Mega.

Steve: You have to solve this problem.

Leo: You, it's up to you.

Steve: In the wake of the closures, he expanded on his plans. Kumar said work is in progress, building off the end-to-end encryption and contacts functionality already working for documents in Mega. Quoting him: "The biggest tech hurdle is providing email functionality that people expect, such as searching emails, that are trivial to provide if emails are stored in plaintext (or available in plaintext) on the server side," says Kumar. Continuing to quote him: "If all the server can see is encrypted text, as is the case with true end-to-end encryption, then all the functionality has to be built client-side. [That's] not quite impossible, but very, very hard. That's why even Silent Circle didn't go there. A big issue is handling emails to and from non-encrypted contacts when Mega's core proposition is end-to-end encryption," says Kumar. Of course, yes, that's what we've been talking about the last couple weeks. So he says, "On this and other fronts, Mega is doing some hugely cutting-edge stuff," he says. "There is probably no one in the world," Leo, "who takes the Mega approach..."

Leo: No one. No one.

Steve: Nope. Nope.

Leo: And that's because they're all so stupid.

Steve: "...of making true crypto work for the masses."

Leo: True, true.

Steve: "Which is," says Kumar, "our core proposition."

Leo: It's what we do.

Steve: So we don't know yet what he's up to. He talked about Bloom filters, which is an interesting filtering technology. Maybe it's going to provide searching of encrypted email. I'm not sure why that's a big need, but we'll see. So they're weighing in. Also, Moxie Marlinspike has apparently come in off of his boat and...

Leo: You know, it brings them out from all the corners, doesn't it.

Steve: Exactly. [Indiscernible]. Wait, wait, we can solve this. So Dan Goodin, our friend over at Ars Technica, reported. And so we know about RedPhone. RedPhone was the secure solution that Moxie was doing with ThoughtCrime.org. Remember, ThoughtCrime is Moxie's site where he's working on this. So now they've got something called TextSecure for Android and iOS. And again, details are still unclear. And I'm a little uncomfortable by the way they're trying to solve the problem. The problem is, the problem with texting security is that the protocols like OTP that we've talked about, are that they're online protocols like SSL. If you think about it, the whole idea of a handshake in SSL is real-time exchange of cryptographic content in order to do key agreement and obtain a secret which you share, which you then use to encrypt your interchange.

The problem with text messaging is it's not necessarily real-time, that is, I mean, by design. It's like email. It can be a store-and-forward sort of operation. So the question is, how do you secure this? And so I think what we're going to be seeing for the future, and what we'll be covering, are various attempts and approaches and ideas. I'm interested to see where this goes.

I think long-term, once we get out of this all, sort of this reactionary, oh my god, we're the ones who have the solution, we'll probably come up with something that works because I really do believe that a consequence of all of this will be the development by the big boys, by the heavy guns, by the RFC committee kind of guys, of some next-generation solutions. There will be pressure to create them that we really haven't seen before.

So the reason I'm a little uncomfortable with what Moxie's doing, aside from the fact we don't really have an understanding of it, is there's something called "prekeys" which they store on the server. And that immediately makes me feel nervous. Apparently, when you create an account for this TextSecure, at least for iOS - and there's some confusion between iOS and Android because, if your Android machine is on, Android is better than iOS is about allowing things to run in the background. And so there can actually be a handshake in the background with Android that iOS just will not support. It just - iOS is going to fight you on this.

So, again, details are very fuzzy. But what I've seen is comments like a hundred keys are pregenerated and stored on the server. So your phone - I'm making this up now. I'm just, it's like, okay, well, how would this work? Apparently a hundred keys get generated. And the idea is that that's a way for you to receive a hundred text messages in a what I can only regard as semi-secure fashion because they have the keys. But, I mean, maybe they're encrypted so that - we just don't know. They could be encrypted so that something - it's just impossible to guess how this thing can be secure, where your phone is pregenerating some - your phone or the server is pregenerating keys, and then somehow that decouples you from needing to be online in real-time. But Moxie's in the game, and with ThoughtCrime, and we'll maybe get more details as this thing matures.

Leo: Now, I'm interested in remail, the idea of remailers. And at some point I'd like you to look at that. The Cypherpunks, who I really trust in all of this, have to do some remailing. Which looks like the only real way to be completely anonymous and private. Because when you send mail, it has to be, I mean, you've got to use PGP;

right? Well, anyway, we've talked about this before. I don't want to...

Steve: Yeah.

Leo: I don't understand how somebody could provide a service, unless it's software on your desktop.

Steve: I completely agree. End-to-end encryption. We have a great question...

Leo: Somebody may do it.

Steve: We have a great question that we're going to get to later in the show about the notion of maybe, well, why not let the email server do that, rather than the client? And it's an interesting sort of thought experiment in moving that one step back toward the server, which we'll talk about. I want to cover a couple more of these things. There's something called Wickr, again, another one of these, oh, we've got a solution for that, W-i-c-k-r. Not clear where the name came from. MyWickr.com is the company. And these guys, okay, they sort of sound like - they're using a lot of the right words. It's free. And it's like, okay, well, it would be nice to understand why you're doing this and how we trust you.

So it's a free app. They show it, I think for an iPhone. I'm not sure how much platform cross-compatibility there is. But there on their site: "The Internet is forever. Your private communications don't need to be." And so deleting stuff is one of their benefits. So they say: "Wickr is a free app that provides military-grade encryption" - it's like, okay, well, I guess they have to say that for Mom and Dad - "of text, picture, audio, and video messages; sender-based control over who can read messages, where, and for how long; best available privacy, anonymity, and secure file-shredding features; security that is simple to use." So, okay, that all sounds good.

Then they said: "We have made this app with the best available security technology, but we strongly encourage you to only send private messages to people you trust." What? Oh, okay.

Leo: Whatever.

Steve: So, you know...

Leo: I don't know why they even say that.

Steve: I guess Anthony Weiner could use this, but he needs to trust the recipient, which actually does make sense.

Leo: Sounds like more snake oil to me.

Steve: Well, that's - yes, exactly. Then they said: "Wickr uses AES-256 to protect data and ECDH-521," so we know that that's Elliptic Curve Diffie-Hellman key exchange...

Leo: Why, that's military grade.

Steve: Oh, I know, Leo.

Leo: Yeah. Whoa.

Steve: Maybe they have Navy SEALs. They haven't said that, but maybe - "for the key exchange. RSA-4096 is also used as a backup and for legacy app versions. So they used to use that, but they don't anymore. They went to elliptic curve, which is shorter keys, and it's going to be faster and forth. Wickr also uses SHA-256 for hashing" - which is the only thing you can use it for - "and Transport Layer Security. Encryption keys are used only once, then destroyed by the sender's phone." Okay, that stands out, I mean, that sounds good. "Each message is encrypted with its own unique key, and no two users can have the same AES-256 or ECDH-521 keys ever. Our servers do not have the decryption keys. Only the intended recipients on the intended devices can decrypt the messages." So that all sounds good. And blah blah blah. I wrote more here when I was making the show notes.

Leo: Blah blah blah is good. That's a good summation, a summary of all of it.

Steve: Yeah. So, and...

Leo: Yada yada yada.

Steve: "We can't see information you give us. Your information is always disguised with multiple rounds of salted cryptographic hashing before (if) it is transmitted to our servers." Okay, I don't know what that means, "if." "Because of this..."

Leo: Well, if you never mail it...

Steve: Yeah, you keep it to yourself.

Leo: Keep it to yourself. That's really trustworthy, then. That's the most.

Steve: If you don't trust anyone, Leo, just don't send it.

Leo: I don't send it.

Steve: Anyway, I don't know. Another one of these things. Then we have Cackle - secure, safe, private, and confidential. I don't know where - it's Cackle-It.com. And then this is a little disturbing. Under the "How we do it" category is, I'm quoting them, "Explaining exactly which ciphers we use at which times and for what reason..."

Leo: Oh, no, we shouldn't tell you that.

Steve: "...would be tantamount," Leo...

Leo: Tantamount.

Steve: Good word, tantamount, "to giving away our company secrets."

Leo: Oh. Oh.

Steve: We wouldn't want that to happen. However...

Leo: Look, look, they put a key on an iPhone. Wow.

Steve: "What we can divulge is an incomplete list of some of the cryptographic methods we make use of: 16,384 bits of ID-based encryption." Leo, that's a lot of bits.

Leo: That's a lot of bits.

Steve: "384-bit elliptic curve encryption; 256-bit and higher AES encryption; the Diffie-Hellman protocol for the handshake." But that's all we'll say.

Leo: Unh-unh, no more, unh-unh.

Steve: No, we don't want to give away any of our company secrets.

Leo: They're in Cyprus, by the way, Cackle is.

Steve: I did see that.

Leo: Yeah. So I'm a big fan of open source. So, you know, obscurity...

Steve: I know, I know.

Leo: Security through obscurity is not...

Steve: And then, you know, and then I gave money to Hemlis. We talked about Hemlis...

Leo: Yeah, yeah.

Steve: ...some time ago.

Leo: How'd that work out for you?

Steve: Eh, not very well. Heml.is. Unfortunately, apparently everyone has been asking them how their crypto works. They're spending time on the pretty colors on their UI. There's been a lot of focus. In fact, on their second update blog posting, they have a spectrum of colors, and they show which letters of the alphabet each color corresponds to because that will help you see which conversation, to, like, follow the threading of conversations. And this is, Leo, it's very pretty. But unfortunately, in the third update, under the topic "The questions about encryption," they wrote - this is they: "Most questions about Heml.is is about the encryption we're going to use."

Leo: It is. It is.

Steve: Yeah, no kidding.

Leo: Most questions is about these.

Steve: Yes. "How it's going to work and details about it. For different reasons" - now, to be fair, English is not their first language, so their English is better than my Swedish. But they said: "For different reasons, we've stayed away from talking too much about the details. It's not because we're arrogant, it's just that dealing with the crypto" - now, this is - I'm quoting them from their third update. "But dealing with the crypto community is really time-consuming." They don't have time to...

Leo: Ohhhhhhhhhh, we've seen that happen before. No time.

Steve: Then they said: "Whatever solution we've decided on would be criticized..."

Leo: Wouldn't want that.

Steve: "...and we aren't interested in the flame war that's inevitable."

Leo: Right.

Steve: Right. "We'd rather create and get things going. Maybe a small lesson for the crypto geeks out there would be to be supportive instead of negative." And it's like, uh, well...

Leo: Steve, you're not being very supportive.

Steve: I thought these were security guys, Leo.

Leo: No.

Steve: That's why I...

Leo: You know, it's very obvious they're web designers.

Steve: Well, I gave them money. Yeah.

Leo: They're not security guys.

Steve: But they did say: "After taking all things into careful consideration, we've decided exactly how the encryption will work." Which is nice. I guess they're not going to tell anybody because they're afraid they'll upset us, which is not really the right strategy. However, they said: "We've listened to all the comments and wishes from you guys, and we are now quite happy with the implementation we're going for."

Leo: Thank goodness. I'm glad they're happy.

Steve: "It's based on free and open source solutions, and we'll release the full source we create for the usage of it."

Leo: Oh, good. All right.

Steve: So that's neat.

Leo: Yeah.

Steve: "More details will follow later, closer to release." So that's hopeful. We'll see what they come up with.

Leo: Mm-hmm.

Steve: So that's sort of where we are right now. We're, I mean, as I predicted last week, this essentially is what's happened, is everyone's gone, oh my god, look, everybody suddenly wants security. Didn't you say that your mother was a security expert? Well, let's [indiscernible]. Okay, so...

Leo: She stores all her passports on the desktop in a file called Passports.doc. That's secure.

Steve: Yeah. So that's where we are. We'll keep track of this. I want to keep receiving people's findings. Send me a little - a tweet or a note...

Leo: You don't use PGP, do you. You don't really use email, so you don't use...

Steve: No, I've just never had a need for secure mail.

Leo: I am going to - I got a couple of very interesting emails, encrypted emails, from a guy who styles himself Demosthenes. You may remember...

Steve: Yes.

Leo: ...the character of Demosthenes from the "Ender's Cycle." He gave me very specific instructions on how to use Cypherpunks' "nym," as in anoNYMous...

Steve: As in a pseudo...

Leo: PseudoNYM servers. Which is an interesting - and it still uses PGP, but it's about the metadata, hiding metadata, as well. So an interesting idea. But I'll pass those along to you, and you can enjoy.

Steve: Okay, cool. So, okay. The most tweeted, to me, topic of the week was the, unfortunately, the source of great hyperbole for members of the press, whose headlines were "No Passwords Are Safe Any Longer."

Leo: What?

Steve: The end of secure passwords as we have known it. Now, if you were using "monkey," Leo...

Leo: Yes.

Steve: Maybe that's true. All that happened, however, is that the well-known, high-performance, HashCat GPU-based brute-force cracking system...

Leo: Yes, which has become very good...

Steve: It has become very good. They had a password length limit of 15 characters. Forever. It's always been 15 characters. And so all anybody had to do is to use a 16-character password, and HashCat couldn't handle it. At the cost of rewriting half of their source code...

Leo: And slowing it down a little bit, too.

Steve: Yes, it was, yes. Essentially they lost 15% because the limiting the password length to 15 characters for all kinds of technical, just like the data path width requirements, there were optimizations they were able to apply at 15 characters or fewer. Which is why, in the beginning, when this was first written, that's what they did. That was their target. But they realized that that couldn't stand. So Jens Steube, who also go by the handle "Atom," wrote in the release notes for this upgrade: "This was by far one of the most requested features. We resisted adding this 'feature' as it would force us to remove several optimizations, resulting in a decrease in performance for most algorithms. The actual performance loss depends on several factors - GPU, attack mode, et cetera - but typically averages around 15%."

Dan Goodin, who's a pretty good technical writer for Ars Technica, wrote: "As leaked lists of real-world passwords proliferate, many people have turned to passwords and passphrases dozens of characters long in hopes of staying ahead of the latest cracking techniques. Crackers have responded by expanding the dictionaries they maintain to include phrases and word combinations found in the Bible, in common literature, and in online discussions. For instance, independent password researcher Kevin Young recently decoded one particularly stubborn hash as the cryptographic representation of 'thereisnofatebutwhatwemake.'" Which if course we all know came from "Terminator 2."

Leo: Oh, really. I didn't know that. But good. Good on you for recognizing that. No?

Steve: And so there is - "thereisnofatebutwhatwemake" is obviously a concatenation of a bunch of words. So those of us who have thought about this a lot recognize that, yes, that's good...

Leo: Not a good password.

Steve: ...but it doesn't have actually that much entropy because...

Leo: Right. Well, and did you see how they found it? It was in - there's a Wikipedia entry with the quote in it. So apparently they're hashing all the Wikipedia entries or something.

Steve: There may be that. But I think maybe you're thinking of this second one.

Leo: Oh, oh, okay.

Steve: Yiannis Chrysanthou...

Leo: Yes, yes, that was that, yeah.

Steve: ...a security researcher who recently completed his master's of science thesis on modern password cracking, was able to crack the password.

Leo: Skip this. Skip this.

Steve: Now, this one is just ridiculous.

Leo: Skip this.

Steve: I can't pronounce this.

Leo: No.

Steve: Ph'nglui mglw'nafh Cthulhu R'lyeh wgah'nagl fhtagn1.

Leo: Which looks like random, but it's not.

Steve: No. That's the fictional occult phrase from H.P. Lovecraft's short story "The Call of Cthulhu."

Leo: Cthulhu, yeah, Cthulhu.

Steve: Cthulhu, ah.

Leo: By the way, James Spawn [ph] said, oh, yeah, that's from "The Call of Cthulhu." In our chatroom. Right away recognized it.

Steve: That's why we have good people in the chatroom. It would have been impossible to use a brute-force attack or even a combined dictionary to crack a phrase of that length. But because the phrase was contained in this Wikipedia article, it wound up in a word list that allowed the security researcher to crack the phrase in a matter of minutes. So this podcast, our listeners' takeaway is abandon anything but true random characters. It's the only thing we have left is...

Leo: If you used a random passphrase, but actual words, wouldn't that be okay? Or no?

Steve: Well, there is...

Leo: Not as good, of course.

Steve: ...[indiscernible] what we make. That may have actually appeared because it appeared in "Terminator 2." But...

Leo: But somebody in the chatroom said, "My doorbell has cow mustard on top of its sticky side walls." It's not - that's such a random phrase. You could remember it. Because that's the issue; right? The best obviously is truly random. The more entropy the better.

Steve: Yeah, I really think we're at the - we're in an era now where tools like LastPass, where they are long, truly random passwords, and you have given up, you are no longer remembering any of those, you've turned responsibility over to this technology, we're at that point now where you have one master password which also really needs to be good, but some - I'm a fan of using a keyboard-based algorithm and something to come up with something really screwy, and that's the way I remember my master password - something sort of semi-mechanical. And then just give up and have a good random number generator or a random password generator make things up for you.

That's just the way I've been operating now ever since I found and vetted LastPass. It's just like, okay, you know. And the good news is that it's - LastPass is ubiquitous. It's on all your platforms. I've got it running on my iPad. And so it'll, when I want to do something on my iPad, it's like, oh, I use LastPass Tab, which is the iOS-based browser, and it says, oh, yeah, here. I'll it in for you. It's like, oh, thank goodness. So you need ubiquity if you're going to have passwords in your life that you absolutely don't know any longer.

Leo: Yeah, the challenge is this master pass, which you have to remember.

Steve: You've got to have one.

Leo: Yeah. I'm going to try, you know, Chris somebody, somebody from YubiKey sent me a note saying that they have a new YubiKey. One of their YubiKeys supports PGP passwords, passphrases. So I'm going to try doing that. Of course, if you lose the YubiKey, you're screwed.

Steve: Yup. So don't do that.

Leo: I still think it's best to have something that's in your mind that you can remember.

Steve: Yeah, well, and the YubiKey, great as it is for its purpose, is USB. And so..

Leo: That's not going to work on Android; right.

Steve: On your phone; right.

Leo: You're going to have to do what Walt did and just memorize the GPS coordinates of his stash.

Steve: And then, did you see what he did?

Leo: Yes, I thought that was brilliant.

Steve: Brilliant.

Leo: Now, this is not a spoiler because it's not a plot point.

Steve: Very, very obscure.

Leo: But he needed to remember a GPS coordinate, so he memorized it. But of course he...

Steve: Short-term.

Leo: Yeah, short-term, he didn't trust his long-term memory, so he bought a lotto ticket with a number of numbers. It wasn't the number that he used, but the second number in with the number of numbers that he'd purchased, you know, he - worked.

Steve: Yes, basically he turned it into a lotto ticket so that there was a record on the refrigerator that no one would ever imagine was GPS coordinates. Yeah, very clever.

Okay. Now, this is bizarre. This is - it's sort of fun and interesting. People should not panic. But it turns out that the Netscape Security Suite, NSS, which is the foundation for both Firefox and now Chrome, has an SSL logging feature. If you create an environment variable, all capitals, SSLkeylogfile, and you set that environment variable to a filename, then you launch Firefox or Chrome and do anything with SSL and then look at that file, it has dumped all of the security keys that were negotiated.

Leo: Wow.

Steve: Yeah.

Leo: Now, is that - that's, like, the key. That's all you need.

Steve: It is, in fact, it is so much the key that if you also, even somewhere else, captured the traffic, Wireshark will decrypt the dialogue for you using the hex which has been logged in that keylog file. So you can try it yourself: SSLkeylogfile as an environment variable, set that to a filename - I set it to C:\herearemykeys.log - and fired up Chrome, went to GRC, looked in the file, and here was this beautiful log of all of the negotiation that had been done.

Leo: Now, it doesn't do that unless you set that environment variable? It doesn't do it by default.

Steve: Correct.

Leo: Okay. So it's a debugging feature.

Steve: It's a debugging feature the developers use. And it is handy if you yourself are a developer, and you don't have something like Fiddler or one of the ways of intercepting secure transactions because Wireshark will - you're able to decode this protocol, drop the hex in, and it's like, bink, there's all of your dialogue in the clear. So you certainly - if you turn this on to play with it, remove it after you're done.

I mean, so this is not a huge issue because, remember, we have to understand what our security perimeter is. Our own system is our own system. We want to keep malware out of it. We want to keep people out of it. This is like Chrome that doesn't encrypt your website passwords, that sort of thing. It's like, in RAM are all these keys all the time. They have to be there to be used, to have dialogues. Normally our browser doesn't write them to disk. And these are keys relative to the server you visited, but that's only going to be the server's public key and the keys that you negotiated for a while. But you don't want to have that happening. So just I ran across this, actually just this morning, this SSLkeylogfile as an environment variable that will cause the local security suite, the NSS, Netscape Security Suite, to log what it does to a hard drive.

Leo: Unbelievable.

Steve: Yeah. And so I just wanted to mention, since all of your other podcasts had commented on Steve Ballmer's news that he was leaving, that he was one of my favorite people there, Leo. I got to know Steve...

Leo: Really.

Steve: Oh, yeah, back in the days when he and Gates were hanging out at Comdex. Bill was always very focused and all about business. And Steve was someone that would have a beer with you and remember your name.

Leo: He seemed like a nice guy, actually, yeah.

Steve: Yeah, he was. And, you know, at the same time, he could never have built Microsoft.

Leo: Right.

Steve: I mean, he wasn't Bill Gates by any means. And so he was a great person, I mean, he was a nice person to work with Bill, to sort of go to meetings when Bill was busy doing something else or couldn't be bothered. And I thought he was - he kind of kept things going for a long time. And I think this whole issue of Microsoft's sort of faltering is not surprising. We've seen Microsoft faltering ever since PDAs first happened because they're a one-trick pony. They've got an operating system that is massive, and it's never been able to run on batteries. And so they've been having a problem with that ever since. Initially it was PDAs. Then it became telephones, as PDAs and phones sort of merged.

And I just - here they are now, there was a news blurb that was saying that, a year from now when XP stops getting security updates, it's expected that fully one third of PCs will still be running XP. And it's like, yeah, because companies don't have money to burn right now, and Microsoft's, unfortunately, their model is one way or the other forcing you to move forward, even if XP works just fine. And so new machines which are purchased will typically have, well, hopefully Windows 7, maybe Windows 8 at some point, God help you. But so they won't have XP. And so it'll just be XP will end up dying off because the machines that had it will end up dying themselves, and new machines will have a new version of Windows. So anyway, I mean, I didn't think Ballmer was ever a genius, but I don't think he was ever expected to be. He was just - he kept things going as well as he could.

Leo: Yeah, yeah. I mean, I think this was inevitable. But, yeah, he's a nice guy. That doesn't mean he should be running Microsoft, just he's a nice guy. It's an important point to make. It's kind of apparent when you read his quotes and so forth. He's a fun guy.

Steve: I did want to share - I got a kick out of an xkcd carton, Leo: xkcd.com. This one is 1256. So you can see it at xkcd.com/1256, or 1256/large, if you want the big version. And what this is, this, again, is classic xkcd. These are questions found in Google autocomplete. And it really is wonderful. It's where you begin to...

Leo: It's hysterical.

Steve: It is fabulous. You begin to type something, and then Google guesses, based on what other people have asked in the past, what you may be in the process of asking. And so it's just a massive screen of sort of fun things. But someone, I don't remember if it was through Twitter or GRC.com/feedback, noted that in the very far bottom left corner, you go to the bottom left corner, then you go to the right one column, so it's in the second column, and the third line up is "Why is there always a Java update?"

Leo: [Laughing] It's right in between "Why don't boys like me?" and "Why are there red dots on my thighs?"

Steve: [Laughing] Why is there always a Java update? Yes, a question for the times.

Leo: Good question. Good question, yeah.

Steve: Our friend Mr. Wizard, Bob Bosen at AskMrWizard.com, has continued producing his video versions of our prior podcasts that he thinks are core and important. He sent me a note saying, "I have just completed new Episode 29 on 'Ethernet Insecurity,' where you covered 'ARP Cache Poisoning.'" And he continues to say, "Your narration provided a fine addition to your 'How the Internet Works' and 'How LANs Work' episodes from back in February and March of 2006." So he's continuing to work on those. For listeners of ours who don't already know, Bob produces some videos that use the audio from the podcast along with his animations to sort of further embellish what we're doing at AskMrWizard.com. So just wanted to give a note about that since he's continued to produce those.

And work on SpinRite continues. I don't think I mentioned before that I have confirmed now that one thing that people have been asking for, kind of, the next release will be able to offer, and that is cooler operation. One of the things that laptops have a problem with is getting rid of the heat that they produce. They often have - there just isn't much room for air to move in a laptop, just because of the physical size of it. And so many laptops, I know that all of my Lenovos have a little vent area, and you can feel, like, hot air being actively blown out of this thing as it's trying to cool off the CPU. I've perfected the technology of completely halting the processor while SpinRite runs. The processor will almost never be running.

Leo: What?

Steve: Yes.

Leo: Well, it's got to run a little bit.

Steve: It turns out that an interrupt, a hardware interrupt can take the processor out of halt, and does. And so we've got the technology now proven for where I am already at one phase of the work that we're doing, halting a processor for three and a half seconds in order to make it absolutely quiet, to determine some aspects of the system's timing.

Leo: That's a good idea. That's a great idea.

Steve: So that nothing else is going on. Well, it also dramatically lowers the power consumption, and thus the heat production, because the whole core, the processor clocks are just stopped. And so what'll happen is 6.1, we've also confirmed, will be able to transfer in 32MB blocks, up from much like a 64K buffer to a 32MB buffer. So we'll be transferring 65,535 sectors at a time. That transfer is initiated, and then the processor is stopped. And it will be sitting there doing nothing, just frozen, while all of that data flows into RAM. And then it'll wake up, check to make sure everything worked, queue up and start the next transfer, and shut down again.

So its operational duty cycle will be, I'll end up measuring it because it'll be really fun just to know, but fractions of a percentage. And so it'll do that, and we're also going to spin down any drives which are not in use. So the drive SpinRite's running on, many people have machines with five or six hard drives in them. It's amazing, the machines that we're testing were sometimes two drives, sometimes more. SpinRite will shut any ones down that it's not working on, again, to further reduce power consumption and to dramatically run the system cooler while it's operating.

Leo: Clever.

Steve: Yeah. So, getting there.

Leo: You may have noted The New York Times went down for a few hours this past week. They now know that it was a spearphishing email, not attacking The New York Times, but a company called Melbourne IT, which is an Australian firm that buys addresses, domain names, and that was a domain name reseller, and hackers changed the DNS records once they got the logins from the domain resellers. So it was a really good example of a spearphishing attack, sent specifically to staff at Melbourne IT. And, boy, that's a...

Steve: And it was effective. yeah.

Leo: Yeah. It wasn't even Melbourne IT, it was a sales partner in the U.S., a partner of Melbourne IT. So very indirect way to get at The New York Times. They weren't actually, it looks like, going after The New York Times, but New York Times plus. And they got quite a few credentials, I gather.

Steve: Wow.

Leo: Yeah. Hey, we're going to take a break. We have questions from our audience. Somebody in the chatroom said, "Can I just come in the chatroom and ask?" No, Steve likes to research his answers. He doesn't like to answer off-the-cuff. He takes it pretty seriously. So what we do is we have a website, Steve's website, and a form there that you can ask questions. If you want to ask questions now, go to GRC.com/feedback. And Steve picks 10 or so questions every other week, twice a month, answers them. And so, yeah, we don't take ad hoc questions because that's for the Tech Guy show.

Steve: I was going to say, I also do keep an eye on my Twitter feed.

Leo: Ah, that's a way to do it, yeah.

Steve: And so if you mention, yeah, if you have something short, and you mention @SGgrc, if you tweet that out, by all means. I normally keep current with everything happening in Twitter.

Leo: Yeah. Ten questions from our vast listening audience, starting with No. 1, Dan in the U.K. He's @dansgalaxy on Twitter. Is it possible, he asks on Twitter, for the NSA to identify a stream of VPN - well, this is very appropriate - VPNed data, and then match it based on variable bitrates to the VPN server outbound? I don't know what he's asking.

Steve: Well, so, now, that's an interesting question because he's talking about traffic analysis. And so if you had a VPN server, is there a way for someone, and in his example he's assuming the NSA would have an interest, in mapping the unencrypted public traffic back to the encrypted, tunneled traffic? And so this is a concern because it's known as traffic analysis. And, for example, one of the things that the Tor nodes deliberately do is introduce variable amounts of delay in their forwarding of the traffic when it comes in the node and leaves the node because they would like to break the association between packet coming in, packet leaving. And so the feasibility of doing this is entirely a function of how busy that server is, whether it's a Tor node, for example, or a VPN server. And also just, like, how closely someone is looking.

So, for example, say that you had a server that only one person was using. Well, it's going to be very difficult to try to convince someone that your traffic that was encrypted going in is not related to the traffic coming out because there would be a burst of incoming traffic and immediately a burst of outgoing traffic. There'll be a one-to-one relationship. And so that makes it very obvious. At the same time, if there were a thousand people all using the server, then to a much greater degree you're able to hide amid them. But at the same time, if you remember also that normally people are going to a specific location out on the Internet, so there will be an IP address where their public stream out of the VPN is going, and then there's an IP address where the encrypted tunnel coming out the other side of the server is essentially going.

And so it's a hard problem to really hide from that kind of analysis because, even if there were a thousand people, one of the things that is characteristic of our use of the web is it

tends to be extremely bursty. That is, you click a link, and there's a flurry of activity where your browser requests the web page from the remote site. Then, when the page comes in, your browser asks for all the resources, another burst of outgoing, and then typically it's quiet for a long time while you the human cogitate over what you just received and read it, scroll, and then maybe click something. And then another furious burst of activity.

So the fact that the traffic is as bursty as it is really, I mean, it helps anyone who, even if you had a thousand very bursty individual users on a single server, here's a burst, there a burst, and then bursts come in, bursts come out, traffic analysis is a way of deanonymizing and associating the VPN user with the public user. And this is a problem for which there's not a good solution. That's one of the things, one of the benefits of Tor is that it's the reason Tor doesn't use just one node, for example. If you just had a single Tor node, this would be - it would be easy to deanonymize. It's by having it hop several times and the nodes deliberately introducing a delay, it's specifically to confound somebody trying to do traffic analysis. So, great question, Dan.

Leo: Yeah, wow. All that in 140 characters. Advait in India, I think, Kerala, India, he's a SpinRite user, and he wonders about Linux versus Windows: Steve, when I browse the 'Net in Windows, I always use NoScript. When I browse using my fully patched, up-to-date Ubuntu, I'm assuming I'm much safer, and I don't worry about using NoScript. Am I putting myself at risk by not using NoScript in Ubuntu? Is it true that almost all current web-based malware and threats will simply not execute in Ubuntu? My understanding is Ubuntu is just a GUI shell around Linux. Thanks, Advait, happy SpinRite owner.

Steve: So...

Leo: His understanding of what Ubuntu is is mistaken, but that's okay.

Steve: Right. The thing to, I think, appreciate here is that there are, unfortunately in this day and age, many different ways of getting yourself in trouble. So, for example, if you are using Java, need I say any more? So a Java exploit could be tied to the underlying operating system, but doesn't have to be. It could be leaking your identity, leaking your session keys for things you're doing. It could be causing a web-scale leakage of information, for example, logging what you're doing on your banking site, even when you're on Ubuntu Linux, not because of the Linux as a problem, but because the plugins and add-ons - like another, of course, frequent culprit is Adobe's PDF. By the way, Leo, have you noticed how that sort of just - has that died down?

Leo: We haven't heard a lot of that, yeah.

Steve: Yeah, that sandbox, it went from, like, a topic every week to, wow, we haven't heard of any more PDF problems for a long time.

Leo: Interesting.

Steve: Because they really did, they finally got sandboxing. They took it seriously and got sandboxing working, and that has really slowed these things down. But so there's the potential problem of a web browser exploit. Then sort of the next level is plugins, which the browser is bringing along to enhance your experience - JavaScript, Java, PDF reader and so forth. And then finally at the lowest level is the OS itself. And it is absolutely true that at the OS level, we're still seeing vastly more Windows exploitation than we are Mac. I would say Mac is probably No. 2, and then Linux is a very distant third, just because the hackers are going where the people are. And the majority of people are still using Windows. And Windows seems to be giving no end of security opportunities for compromise.

Leo: Jim Breen in Chicago notes the security implications of Yahoo!'s recycling its one-year dormant email accounts: Steve, as you might be aware, Yahoo! began recycling email addresses this month. I became aware of it as my employer, a large online eCommerce site, scrambles to figure out how to handle the fact that the email addresses associated with some of our user accounts could soon belong to someone else. I wish - this is stupid.

Steve: Yes.

Leo: Stupid. As Yahoo! starts recycling the accounts. So if you had leo@yahoo.com, didn't use it for, what is it, a few years? What is it?

Steve: No, it's six months, I think.

Leo: Six months?

Steve: Yeah.

Leo: That they might give leo@yahoo.com to somebody else. Who doesn't want it, trust me. There are two big problems caused by Yahoo!'s decision to recycle email addresses which have been dormant for a year. A year, I guess.

Steve: One year, yeah.

Leo: The first, which seems to be getting the most attention based on Google searches, is that email senders with these Yahoo! addresses in their mailing lists risk sending email to people who didn't sign up for the list - who knows what kind of stuff, password resets - and having those people mark the email as spam, which then hurts the sender's email reputation with ISPs. Well, that's a good point.

Steve: Yeah.

Leo: This risk can be mitigated by removing the email address from the mailing list, if Yahoo! has been returning a "hard bounce" error for previous sends to those email addresses. But will they hard bounce if they reassign it? No.

Steve: Nope.

Leo: The second problem is relevant to a security - I wonder if they're going to do, after a year we hard bounce for six weeks and then assign it. Maybe they'll do that.

Steve: Yeah, actually what I read was that they were going to hard bounce for a month and then reassign it.

Leo: Okay. That would be better than nothing. The second problem is relevant to a security audience. It's harder to solve. If the original owner of the Yahoo! email address associated it with an account on another site, then the new owner of the new email address will be able to take control of that account, of course, using the standard password reset functionality. Since most websites base proof of ownership of an online identity on ownership of an email address...

Steve: Uh-huh.

Leo: True - Yahoo!'s decision to recycle email addresses jeopardizes the accounts of those email addresses' former owners on sites across the Internet. This seems like something the Security Now! audience should know about. Thanks for the great show. Wow. I didn't know about it. That's terrible.

Steve: Yeah.

Leo: Terrible.

Steve: It's caused - even our old friend Mat Honan has weighed in, saying, no no no no no, do not do this.

Leo: Horrible. Marissa Mayer ought to know better.

Steve: And actually she's been quoted as being the motivation behind this.

Leo: Of course.

Steve: Yeah, saying, oh, well, we're going to spiff up Yahoo! and give it a facelift. And, I mean, it's been around forever. And what they're looking at is they're looking at all these

email addresses that no one can use anymore that are, quote, "good email addresses," unquote, not BarneySmith3272653274897, but just Barney Smith.

Leo: Hey, you didn't get an email address earlier, you missed the lottery. Sorry, buddy. You know what I mean?

Steve: I mean, it would make so much more sense to, like, do Yahoo2.com or, I mean, like change the - make a small change to the domain name or something, rather than take this huge, huge bulk of retired email addresses and make them available again. They're still getting email into them, which are bouncing. They're going to forever. But more importantly, and there was some commentary that I appreciated, is that they look at - they decide that the account is dormant after some length of time. If you're not logging in, even if you're getting email, if you're not logging in to receive it, they say, oh, well, it's dormant. And so that's their reassignment basis.

Well, the problem is people often use a dormant account as their email security. We've talked about have a separate account for your password recovery. Don't use your normal high-traffic account for that. Have password recovery go somewhere else so that it's extra secure. And so now they're saying, oh, well, we're going to - if it's been dormant for a long time, we're going to free it up so people can get it. And the point being made here is it is email, as we all know. That's the way you authenticate. That's the only thing we have for, like, proving who you are. And so they're going to say, eh, no, we're going to let that happen.

Leo: Baffling.

Steve: Bad idea.

Leo: Baffling. I guess one thing they could do that would be more sensible is to make a new TLD, Yahoo.me or Yahoomail.com or something like that.

Steve: Or I said Yahoo2.com or something.

Leo: Yeah. And then you've got the whole set again, fresh. And the other thing, this is for anybody listening, and I think our audience would know this, but this is the very strong argument for buying a domain name of your very own and using it for your email address, so you have a permanent address. And then if something like this happens, you don't have to worry about it, you just move it to some other service. Just own your own damn email address. It seems like nowadays that should be really the real answer to this. Don't use somebody else's.

Pat Cho in Sacramento wonders whether files can be securely deleted - oh, this is a good one we get once in a while, you're the guy to answer - from flash drives? Steve, is it possible to securely delete files from flash drives with the AxCrypt utility you recommended from a previous podcast or any other utility? From what I have read, flash drives do writes differently for wear leveling, and so it may not be possible to overwrite a file. Is this a cause for concern? Pat Cho.

Steve: So, yes. It's sort of related to hard drives, but even worse. Hard drives will remove a sector from use when it becomes unsafe to store data there because of a defect in the magnetic storage surface. So hard drives have a pool of spares, and they will "spare out," as it's called, spare out a sector and replace it with sort of a fresh good one only when there's a problem. Flash drives work differently. Flash, as we've discussed, the actual technology of writing to a flash drive involves fatiguing the material of the drive. You use a high voltage to break through the insulation and sort of squirt electrons onto a little isolated pad, where they're stranded. And that creates an electrostatic charge which can then be passively sensed. So you read that there's that little charge there. But the act of writing a one or writing a zero squirts or drains electrons through this insulation, which over time fatigues the insulation. The insulative-ness breaks down.

So in order to solve that problem, because our operating systems tend to heavily use certain areas of the directory, the actual directory structure, the metadata, which contains filenames and the directory tree, many files tend to be written much more often than others. So that would create hotspots where the regions of the flash drive containing those files would fatigue much more quickly than areas that weren't ever being used. So flash drive controllers deliberately do this "wear leveling," as it's called, and it's exactly what it sounds like. They level the wear so that there isn't undo wear occurring in one location. And it's ongoing all the time.

So whereas for a hard drive it's only if there's a problem, it is fundamental to the way flash drives write is they have logic that is constantly remapping the surface so that the entire region of the flash drive is overall being written about the same amount. What that means is prior versions of a file may exist on flash drives, and the work that the group down in San Diego has done on recovering from wear level drives, like essentially circumventing the controller to say, no, I don't want that sector, I want to look at the raw storage region, they've done that, and they have verified that this wear leveling means that all kinds of prior instances of data on the drive is there and is definitely available. So what all of this means is, if you're concerned about security, you should never, ever, not once, write unencrypted data to a flash drive.

Leo: Wow.

Steve: The first thing you need to do is, for example, install TrueCrypt. TrueCrypt will do a beautiful job of encrypting the drive so that everything you write goes through TrueCrypt on the way to being stored, and then you absolutely don't have a problem. Or use AxCrypt, which is a nice little freestanding utility. Or even WinZip. WinZip is now cross-platform. I was looking at it the other day. It uses very strong encryption. And so you just use a good cryptographic key and then use AxCrypt or WinZip or some enciphering tool so that what you store on the flash drive is always encrypted. Otherwise, to a much greater degree than for hard drives, if somebody really wanted to get at your data, they could.

Leo: You'd have to physically destroy the SSD. You'd have to smash it, and smash the chips.

Steve: You've got to crack it open and get to it. But it turns out it's entirely possible to do that. The guys in San Diego...

Leo: Interesting.

Steve: There's a group of researchers who have been experimenting with this.

Leo: Now, if I, after the fact - I have an SSD, external USB SSD. If I, after the fact, apply full disk encryption, even though I've been writing to it unencrypted for a while, they'd have to have the key to the full-disk encryption to get down to the stuff; right?

Steve: If you - no. If you...

Leo: I turned on Apple's FileVault encryption, which is whole-disk encryption.

Steve: Right, or TrueCrypt. If you add TrueCrypt later, then the problem is you'd have, I mean, over time of using it that way, it would tend to replace the less recently written regions with more recently written regions.

Leo: Which would be encrypted at that point.

Steve: Which would then be encrypted, exactly. So over time the previously unencrypted data would kind of get pushed out of use by the wear leveling, which is what created the problem in the first place.

Leo: So it sounds like, if you buy a new computer, and almost all laptops and many desktops come with SSDs, the first thing you should do is turn on full-disk encryption. Otherwise you're just really...

Steve: Before you do anything else.

Leo: Because it's going to be too late if you do anything else.

Steve: Exactly.

Leo: And Windows comes with BitLocker, and Apple comes with FileVault. These are, as we've spoken about before, I don't really trust any non-open source utility. But that would be the easiest way to do it because you don't have to install it.

Steve: I just like TrueCrypt because it's portable. I mean, it's been very well engineered.

Leo: Yeah. All right. Okay.

Steve: But you're right. There are native solutions, BitLocker and...

Leo: Yeah, they're just operating system based.

Steve: Yup.

Leo: Now, is there any way, after the fact now, to wipe the drive? No.

Steve: No, there isn't. No. This is...

Leo: It's too late.

Steve: This is done below the level of the API. There are, in the latest specs, the so-called ATA and ATAPI, the AT attachment spec, this is where I've been living for the last couple of months, this is where SpinRite lives, there are commands, for example, a secure wipe. But it's not exactly clear to me yet how that functions. And the other thing I want to look at, and I will be, is the idea of the password on the drive, whether using the drive's own password, what level of security and where that could be bypassed. But bottom line is I would - if it's a black box, it's a black box. It's very much like the cloud, the Internet. We say "pre-Internet encryption" because we don't know what that cloud is going to do. Similarly, we don't know what that drive is going to do. Much better not to be concerned about it. TNO applies to your drive, too. Encrypt it before it gets written on the drive.

Leo: Wow. And if you haven't, which most of us haven't, I mean, I have tons - all my computers have SSDs now.

Steve: Hmm.

Leo: Too late.

Steve: Yup. The good news is SpinRite will recover them when they have a problem. But it is the case that you would want to add encryption immediately...

Leo: Do it, turn it on right now and just keep using it.

Steve: Right now, and just keep using it. And as it gets used, the wear leveling that caused the problem in the first place will also cause the solution because it'll still be wear leveling, but you'll be leveling - it'll be pushing the unencrypted stuff out into history,

overwritten by encrypted data.

Leo: Wow. Mike in Philadelphia has some PGP worries: Thanks for the PGP episode, Steve. That was last week's, by the way. I had already downloaded Mailvelope and generated keys a few weeks prior, so I was happy for some background. Here's my worry, though. Where do those PGP keys come from? Are they generated by my browser, or are they auto-generated by some key server? My worry is that these are generated by a key server, and I can choose to accept them or generate a new pair. My concern, of course, is snooping while the - we can just skip this, because they're not.

Steve: Right.

Leo: He's got all sorts of concerns.

Steve: But, yes. So I wanted to make sure that nobody else was concerned. The whole concept of PGP or even SSL keys that servers use is the private key, the keys to the kingdom, absolutely never leave your control.

Leo: And they're created by the PGP program you're using, in his case Mailvelope.

Steve: Right.

Leo: Locally. They're not created by a key server anywhere or anything like that.

Steve: Correct. So Mike's concern was that there seemed to be a server involved. And because this was a web-based solution, he was uncomfortable with, well, I'm seeing things appear in the browser. Did the browser make that, or did it get it from a server? Which is a reasonable question because everything we see on a web page comes from a server somewhere. But in this specific case, it is cryptographic code running in the browser that on the fly generates the key. And it never - your private key never leaves your control. And that's fundamental to the whole public/private key technology, which is one of the things that makes it really so fundamentally cool.

Leo: And it's generated locally, by your local software. And that's why we say use open source software, so that you make sure that there's no code in the software that generates the key, a good, secure key, and then emails it off to the NSA or something like that.

Steve: Exactly.

Leo: Open source software, you or somebody can validate that it's fine. And so that's a nice thing about PGP. It is open source. I'm sure Mailvelope is open source

so you can see what's going on, generate secure keys. And again, I've said this before, but even if you happen to lose control of your private key, there still needs to be a passphrase to use it, to unencrypt it.

Steve: Right.

Leo: So this is why you should not use - what was that "Terminator 2" passphrase? You should use a good - and passphrase might really be the wrong thing to say because it implies an English-language sentence.

Steve: It does, yes.

Leo: And we really shouldn't be using that. We should use long, random strings of letters.

Steve: Passcode, yup.

Leo: Passcode would be much better, or password. Even "word" isn't a good - yeah, passcode. Random string of upper and lowercase letters, punctuation, and numbers.

Steve: GRC.com/passwords.

Leo: He'll generate one for you of any arbitrary length.

Steve: Yup.

Leo: And then do like Walter White does: memorize, memorize. It's good for your brain. Buy a lotto ticket. Steve Gibson, security guru; Leo Laporte. A few more questions for you, Steve, starting with this one from a radio station, Dan Uff in Allentown, PA. He writes: I'm the owner of a small Internet radio station, WDMU Internet Radio. I'm in the process of trying to fill my radio time with quality content. I'd like to consider having your show as part of my station's lineup. I think the show would be a great fit and help educate my listeners about Internet security at the same time. I see the show is made once a week. That's what I am looking for. I'm also a big fan of Leo's and blah blah blah and would be honored to have him played on my small station. Thank you for considering this request. If you have questions, please feel free to contact me directly. Thank you.

Steve: So what do you think?

Leo: It's okay by me.

Steve: That's what I thought.

Leo: So here's the deal, yeah, here's the deal. And you should contact lisa@twit.tv to get formal approval; okay? Lisa is our CEO at TWiT.tv. But here's how it works. If you look at our license, it's at the bottom of every web page at TWiT.tv. We are Creative Commons licensed. And you can read the details of that license. We do ask that you adhere to our license, which has three requirements. It's okay to play it freely for noncommercial purposes as long as you give full attribution, that's simple, TWiT.tv. In fact, you don't even have to do anything because, unless you cut it off, it's at the beginning of every show.

Steve: It's all in there, yes.

Leo: And the third thing is that it's a share-alike license. Which means, if you make a mashup, which you're even allowed to do, that you have the same license on the mashup. Now, I would ask personally that you leave our ads in because that's how we monetize. But our license does not require - I probably shouldn't even mention this, but does not require you to do that. If you're going to put it on a commercial radio station, if you have your own ads on there, then you need to get our permission. And lisa@twit.tv is the email address. But people do this all the time. We're rebroadcast a lot of different ways.

Steve: And I don't think we'd ever talked about it, so I just wanted to...

Leo: Yeah, I love it. And you may add your own license. I don't know, Steve. But that's the...

Steve: No.

Leo: ...license on all of our...

Steve: Creative Commons.

Leo: Yup, CC, noncommercial attribution share-alike. And you can click the link at the bottom of every web page at TWiT.tv if you want to see the specific verbiage. As long as you adhere to license, you don't even have to ask our permission. If you want to use it in a commercial environment, which this might be, then we're very - we're absolutely lenient. But do email us, lisa@twit.tv. What we don't want to do is have people, like, use it to make money, cut our ads out, put their ads in, things like that. That's just rude. Be polite.

Steve: That's creepy.

Leo: Yeah. John in Sacramento has an interesting thought experiment about PGP. He's been listening since Episode 1, which is about as long as you can, nine years. He was listening to our last episode [SN-418]. He had the thought, why not put PGP in the email server instead of the client?

As I understand it, we have SSL encryption between the user's email client and the server, whether it's using IMAP, POP, webmail, SSL. And it works with mobile, as well. With PGP on the server, the user connects securely to their email server in whatever way is most convenient to them, creates an email. And then, as part of pressing the Send button, once the server gets it, the server will encrypt it with PGP before sending it out to the recipient's server. Once there - by the way, there's the rub - the recipient's email server would handle the decryption when the recipient connects, again securely in whatever way is most convenient for them, and download the decrypted message. All that's needed is a client capable of connecting to the server securely, which is much easier than trying to deal with plugins, et cetera, et cetera. What do you think, Steverino?

Steve: Well, the problem is, well, there are several problems. One is that the moment you step back from - and the right term is "end-to-end encryption." That's one thing we're going to be hearing. It's a term we haven't really spent that much time focusing on, but it's been implied in many of the things we're talking about. SSL is end-to-end encryption. You are encrypted where you are; it's decrypted where it's going. And everywhere in between it's a pseudorandom stream of noise that has no meaning to anyone. So that's the only way to be secure. As soon as you step that encryption back even one stage from the end-user, you start having problems.

And so, for example, as you've alluded to even when you were reading this, Leo, John is right in that, if we have an encrypted link to our server, then our email is safe. And if the server then encrypts it, and it's encrypted in flight to the other server, it's safe. But notice that he said, "Once there, the recipient's email server would handle the decryption when the recipient connects." So it's the weak link at the server where now we have vulnerability. Whereas, if you do leave it encrypted all the way to the endpoint, you're not having that vulnerability.

But there are more problems. For example, one of the beauties of the way PGP works - and Leo, you've alluded to this - is, if you've got PGP locally, and you've got a bunch of email addresses associated with the recipient's PGP keys, your client knows when to encrypt and when not to. It knows whose keys it's able to encrypt your message for. Or, if your client sees that you don't have a PGP key, it just sends it in the clear. So you would be - you'd lose that level of control and feedback the moment you move encryption away from the client, away from the endpoint, which is really where it needs to be.

What you want is it's encrypted before it leaves the device. It's not decrypted until it arrives at the recipient. Anything else, I mean, and John's right that you would have secure links every step of the way. But it does create a problem where, for example, the NSA could say to whoever it is running either your server or the destination server, oh, we want to take a look at that.

Leo: Yeah. I mean, there are solutions that do this. HushMail does this. But you're trusting HushMail.

Steve: Yup.

Leo: And so it doesn't - so the problem with PGP is that you have to create a key and share the key. And people find that complicated. You can't send an encrypted email to somebody who's not shared his key with you or for whom you cannot find a key online. And so that's the pain in the butt.

Steve: Well, okay. In fact you are - this is the perfect segue into the next question, Leo. I think you - go ahead.

Leo: Here we go. Question 8, Tim in Kansas, who really likes the idea of secure email, says: I just thought I'd ask if you could go over the challenges of using PGP with Outlook - there are free options, but Gpg4win only supports older versions of Outlook - and those on Exchange servers. As far as I know, only commercial options exist for Exchange, and I'm not certain if the flexibility is on par, say, with Thunderbird or GPG or Enigmail. Thank you so much for covering this topic. I hope to see greater adoption of it as, realistically, I've only been able to email myself and Leo. Thanks for replying to my test, Leo. No one else I know uses it, nor can they yet be convinced. Best, Tim. Yeah, I have now almost 200 keys in my keychain.

Steve: Yeah. And this of course is the problem. I mean, famously, this delayed Snowden's release...

Leo: Yeah, he couldn't get Glenn Greenwald to do it.

Steve: Exactly. Glenn refused. He said, oh, I'm busy...

Leo: I don't have time.

Steve: Yeah, I don't have time. And...

Leo: Now, I should say, once you exchange keys, or you figure - once you have that going on, it's really easy. I just push a button and enter my passphrase in some cases. In some cases, like on the Mac, the passphrase is stored in the secure Mac keychain, I don't even have to do that. Once I'm logged in, I'm sending - any time I email these people, I can send them completely secure email, except for the metadata.

Steve: Yeah. I just don't know. It'll be fun as an observer to sort of watch and see where this goes. My sense is people don't care that much. They're like me. It's like, well, you know, I'm just sending, you know...

Leo: Just understand your email's a postcard. If you don't care, it doesn't matter.

Steve: Yup.

Leo: And I agree, this is more an exercise. My first PGP key I created in 1997. I've been...

Steve: Because you could.

Leo: Because I could. I've encouraged people to do it, and I have always published my public key on my website, Leoville.com. I had an hysterical exchange with somebody who said, "I've looked everywhere. It's not on your website." Well, it is, it's on Leoville.com. "Well, no, but when I Google you, the first result is Tech Guy Labs. Why isn't it there?" Well, it's not there. It's on my personal website, Leoville.com. "But it should be on all your websites." No, it's on Leoville.com or on the key server.

Steve: It's where it is.

Leo: It's where it is.

Steve: Not where it isn't.

Leo: Yeah. Sorry, dude. Yeah, I should also put it in your pocket, but it's not [laughter]. Nevertheless, more than 200 people have - and I apologize. I have actually been unfortunately swamped by people who want to exchange keys and test their setup.

Steve: Because you're the only one they know who...

Leo: I know, me and Tim.

Steve: Yeah, let's get - okay, Tim, maybe we can give out your email, and then you'll be part of a crowd of people.

Leo: Well, what I do is I sign every email that goes out. And people see this PGP block, maybe they start to get the message. I don't know. I've done it for years, and it's not - you're right, nobody wants to do it.

Steve: No.

Leo: John O., Denton, Texas wonders about "Cookieless cookie tracking." Steve, have you heard about - of course you have, Steve - about this cookieless track

technology that cannot be blocked? It doesn't rely on cookies, JavaScript, HTML5, localStorage, sessionStorage, globalStorage, Flash, Java or other plugins; nor on your IP address, your user agent string, or any methods employed by Panopticlick. Can you tell us about it?

Steve: Yeah, this was also in the news. I'm not sure why it sort of sputtered up to the top. We've talked about it...

Leo: It's nothing new.

Steve: Right. It involves ETags. One of the things - which has essentially, it's been referred to as a cache cookie, the idea being - and it's in use by websites right now. When a website provides you with an asset, like the icon, the little favicon for the URL, or any of the embellishments that are standard on the site, as we know, browsers cache that, on the theory that it's much better to just save a bunch of this stuff locally - I mean, look how big our hard drives are, we can afford to save it locally - than going through the roundtrip of asking the remote server to send it to us again. So it saves on the server resources. It saves on time. It makes the whole experience much better.

What happens is, though, browsers may want to verify that there's been no change. So the server, when it provides an asset like an icon, for example, for the page, it will add a header, so-called metadata, to the resource, called an "ETag," which is - it's meant to be an opaque token, which is to say, the value is what's important, not what the value means. That is, the value means something to the server. It might be a hash. It might be a CRC. It might be a checksum. It might just be an incrementing value that changes when the object changes. But that's the key because the browser sends back when it's being - when the page is loaded, says I need to display this icon, the browser sends back the same ETag that it received, saying I have this icon that you provided me with this ETag. If the ETag is different now, send it to me.

Well, that's tracking. That's a way for the browser, I mean, it's exactly like a cookie. Well, it's not exactly like a cookie. It's cookie-esque, highly, because basically it's tracking your cache. It's the way for the remote site to be issuing these tokens to all the things in your browser that the browser's going to cache. And the browser sends them back to make sure they haven't changed. Which identifies you, uniquely identifies you, if the ETags are unique. So that's the key. If the ETag was the same for every icon that it sent to everyone, then when they came back, the server wouldn't know who you were.

But ETag, again, is an opaque token. So if it's something about the object plus something about you, then when it comes back it essentially contains uniquely identifying information so that your copy in the cache is different from somebody else's in their cache, even though they went to the same site. So cookieless tracking via ETags, via the cache in your browser, is something, it's one more way that we're being deanonymized, or at least tracked, as we move around the Internet.

Leo: Robert Osorio in Lady Lake, Florida wonders about the Trusted - what is TPM? Trusted Program Module?

Steve: Platform.

Leo: Platform Module.

Steve: Platform Module.

Leo: And the German government's concerns. We've been talking about this for a while. The article about the German government says that the German government thinks Windows 8 is a trojan horse for the NSA and recommends people not use it: A year ago I would have dismissed this as tinfoil nonsense, he goes on to write. But now? One has to at least admit it's possible. After going to the Techno Security conference for a couple of years, I know that enterprise customers have to vet hardware for trojans embedded in firmware.

I'm not sure about TPM 2.0 being forced upon you in Windows 8, though. Also, since it's a device, and any device can be disabled by some kind of hack, I would think you could bypass it. If Windows doesn't have a driver for it, or if the driver is disabled, I don't see how it could be forced on you. As I understand it, what TPM does right now is just provide a hardware crypto engine and a very good random number generator. Vista, Windows 7, and Windows 8 will use it if it's available, but it is not required. Love to hear your thoughts on this.

Steve: Well, so we've done an episode in the past on the Trusted Platform Module. And it is as Robert suggests. It is basically a crypto engine. It's a physical chip by requirement, essentially, soldered onto the motherboard, that creates an identity for the machine and also a vault for secrets, for cryptographic secrets. And the way the thing is designed is it is possible for you to ask it to validate things; but it is not possible, there is no API that will cause it to give out your secrets. So as I understand it, Leo - and I have not researched this because I've been busy with SpinRite. But as I understand it, all of this nonsense seems to me that the German government is complaining about is this notion of sort of a loss of sovereignty to, like, their own independent control of a Windows 8-based system which has secure boot technology and is locked to the Trusted Platform Module on the desktop or the laptop's board.

Leo: Yeah, I don't - if you think about it, the logic of it is strained. I mean, if you're using anybody's closed source code, it's difficult to know what it's doing.

Steve: Yeah, virtually impossible.

Leo: Yeah. And so the TPM module or not, I guess what they're saying is don't assume that TPM gives you some sort of security from Microsoft.

Steve: Well, one of the things that I've got on my short list of topics is the Windows 8 secure boot technology. I want to look at it closely.

Leo: Which, by the way, can be disabled.

Steve: Yes, yes. And in fact has to be in order to run SpinRite. A lot of the testers who've been playing with the testing code are booting their Windows 8 machines, and turning secure boot off is one of the things that people will need to do in order to - essentially the idea is that how do you ever know that a trojan or a rootkit didn't get into the system before Windows began? That's the Achilles heel of ever being able to trust Windows once it's running, is was there a shim? Was there any opportunity for running something untrusted? And so it's really been carefully thought out, this notion of we're going to start from something we absolutely know, and every single step forward we're going to validate that we're only running - we're still in a trusted enclosure, essentially. And so that's, I mean, it turns out it's a hard problem to solve. And fascinating from an academic standpoint.

Leo: And whoever wrote this article didn't really understand what's going on. I mean, the German, whatever German government official said this. If you're using it, you know, it's the same reason we don't use Chinese operating systems here in the U.S. If somebody writes the operating system, and it's closed, has got any closed source code at all...

Steve: They have no idea what it's going to be doing.

Leo: No idea what it's doing.

Steve: None.

Leo: TPM or not.

Steve: Yup.

Leo: And TPM does not secure you from the operating system by any means. That's not a point of it. Steve, we've come to the end of 10 fine questions from our lovely listening audience. That means we're at the end of the show. I thank you. Encourage everybody to follow Steve on the Twitter. He's @SGgrc. He's also at GRC.com on the web. If you go there, you can get SpinRite, world's best hard drive maintenance utility, a must-have. But you can also get lots of other stuff, lots of freebies. Steve really labors away to help you be safe, secure, and sound. The passwords are great, all sorts of stuff. GRC.com. If you do have a question, that's the place to go to post your questions for future episodes: GRC.com/feedback.

Steve: That's where these came from.

Leo: Every one of them.

Steve: Yup. Well, except for the one...

Leo: Well, one on Twitter.

Steve: ...from Twitter, yup, exactly.

Leo: And you can also, while you're there, you might want to get the 16Kb version if you're bandwidth impaired; or, even smaller, the text version. We have full transcriptions at GRC.com. The high-quality audio and video versions are at TWiT.tv/sn, for Security Now!, and wherever finer podcasts are aggregated. Subscribe to your favorite version, or several. That way you're sure never to miss it. Steve, I thank you so much.

Steve: Always a pleasure, my friend. And we'll talk to you next week.