Listener Feedback #137

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-340.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-340-lq.mp3

SHOW TEASE: It's time for Security Now!. This is a fun one. We do this every other episode. You get to ask questions. Steve answers. He'll explain all about the SSL security myth and more. It's coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 340, recorded February 15th, 2012: Your questions, Steve's answers, #137.

It's time for Security Now!. Get ready to fasten your seatbelts and protect yourself on the Information Superhighway with this man, our Explainer in Chief, Mr. Steven T. Gibson.

Steve Gibson: Explainer in Chief.

Leo: Explainer in Chief. Steve's the man in charge at GRC.com, the creator of SpinRite, the world's finest hard drive maintenance utility, and many other free security utilities, and joins us every week to talk about all these issues. It's a Q&A episode. Is that right, Steve?

Steve: It's a Q&A episode and we have just a bunch, a flurry - a flurry of news has landed on us this last week, a bunch of it recently, some really interesting things. A bunch of people have picked up on a report from a paper that was released in advance of the conference, which is not till August. The paper was released because the researchers felt it was too important for them to sit on, which is an analysis of the public and private key system which we've spoken of many times, I mean, I don't know if there's an episode of Security Now! where we don't talk about SSL and this. It turns out that the

random number generators which have been used to generate these, and we depend upon their randomness, eh, they're not so random.

Leo: It's pseudorandom.

Steve: It's not good. So anyway, there's been actually a lot of misinformation about this, even from some well-known tech writers who have said that there are flaws with the SSL system. Well, there have been some, but this is not one. This is a particular implementation problem. And we often discuss these fine points, like what does that actually mean relative to there being a problem with the system, to have a problem with its implementation. Of course those are two very different things from a security standpoint.

So, lots of neat news. We did have a second Tuesday just passed us. Microsoft did their typical batch of updates. These are important. They did nine different updates which closed 21 security holes. I noticed - I hesitated there because I had a typo in my own notes. I said "Nice updates closing 21 security holes" instead of "Nine." Four of them are remote code execution, critical rating. So no effort required on the part of the user to get their systems compromised. Oh, and Microsoft is thinking that they have that weird new term they use for - it's the "exploitability index." And so they're saying that 12, about half of these, are going to be readily exploitable. So this is not something you want to sit on. Here we're recording this the day after these all came out, yesterday. So this is an update your machines and reboot them, as will probably be required. And you can feel at least a little bit safer.

Adobe did a critical Shockwave update. We don't run across Shockwave that much, and that's good news because it's pretty much an obsolete technology. I think every time I say that - oh, no, I guess I'm talking - I was going to say, every time I say something, Elaine reminds me that she's still stuck using it, but it's not Shockwave. (Elaine our illustrious transcriber, who actually we'll be referring to at the very end of the show because we have a bonus unintended consequence of something that happened which I got a kick out of last week that I'll share with our audience.) Anyway, but she's using that horrible, obsolete audio format - oh, RealMedia. She uses RealMedia for some reason to do her podcast transcribing. I think her player, maybe it, like, runs with footswitches or something that she needs to have.

But anyway, in this case Shockwave, if you don't need it, you want to get rid of it. You can go to Adobe.com/shockwave/welcome, and it will tell you if it's installed. And if it's not installed, that's good news. Don't install it. If it is, then you definitely want to update it because they fixed some problems. And unfortunately this is the kind of thing where, if it's installed and not being blocked, for example, by NoScript on Firefox or ScriptNo over on Chrome - and blocking is good, of course. If it's not blocked, then this is the kind of thing where drive-by websites are able to infect you, or targeted email, where you click the link, and that is able to load some funky Shockwave file that leverages a known exploitable problem in order to execute code on your computer.

And later in this podcast we talk about - we answer a question from someone who just can't seem to keep from getting his family infected all the time, or they're infecting themselves. And he's like, what can I do about this? So...

Leo: I can't help myself.

Steve: Yeah, exactly.

Leo: Tech news.

Steve: So, okay. Top of the list here is, quote, many, many websites have mis-covered the story. It's a complicated story, so I don't fault them. But for our listeners, who are equally technical and able to understand these subtle differences, it's important that we not throw the baby out with the bathwater. It has been claimed that flaws were found in SSL's public key encryption, which did not happen. What happened, however, was really interesting, which is that, through an analysis of the actual keys, the public keys in use on the Internet, a bunch of researchers discovered that they were not as random as we all assumed they would be.

Leo: Well, that's kind of interesting.

Steve: Oh, it's really, oh, no, this is really interesting because there are some consequences to this. Okay. So, and I love how we're always talking about things in the show that end up tying into things that happen afterwards because just last week, Leo, you and I were talking about the scarcity or lack thereof of prime numbers.

Leo: Right.

Steve: And how, as we know, the whole public key technology, or at least the RSA style - because there are different kinds of public key. There's Diffie-Hellman and DSA and Elliptic Curve and other ways of having asymmetric keys, where the keys are different, one you encrypt with and one you decrypt with, as opposed to symmetric keys, where you use one for both.

Okay. So the RSA style, it relies on the fact that it is very difficult to factor numbers, that is, really, really, really big numbers. They're just, no matter how much time has been spent and how much smartage has been aimed at it...

Leo: Smartage?

Steve: Smartage, yeah.

Leo: That's a good word. I'm using that from now on.

Steve: I've been drinking coffee since 5:00 a.m., Leo. No matter how much smartage is aimed at the problem, we've had no breakthroughs, no serious breakthroughs in factorization. Now, we're getting better at it. We're creeping forward. For example, 512-bit numbers have been factored. 768-bit numbers have been factored. And it's expected that we're probably not too far away from 1024-bit numbers, which is why, in general, there's some pressure moving the industry towards 2048-bit numbers, doubling again the size of these numbers, which we're going to be asked to factor.

But the point is that we take two large primes and multiply them together in order to get the product of those primes. And that's the operation which is not easily reversible, by which I mean we don't know how to reverse that. So we rely upon the unknowability of both of the primes in order for that to be the secret. So the secret part is what were those two primes. They're multiplied together, and that's what's used to produce the public key of our whole public key crypto system. That's what any web server sends to us when it wants to establish an SSL connection. It sends us its public key. And we're able then to use that to verify that it was signed by someone we trust, so we trust the certificate, and we use the public key in order to encrypt something that we send back to the server. And only the server that has the matching private keys, which are those two prime numbers, essentially, is able to decrypt what we sent.

And that's how we solve the problem of the man in the middle getting involved. The man in the middle saw the public key go by, but has no way of intercepting what we're sending back, which is encrypted using that public key. Only the private key can decrypt it. So here's the problem. If the original random primes are not very random, then the whole system sort of falls apart. I mean, the technology, the concept, everything we're fundamentally relying on in terms of the mechanics are fine. But the assumption that nobody could guess the primes, if that isn't valid, if they're not really good randomly chosen primes, then something, a requirement for security falters.

And what the researchers found by looking at many, many millions of public keys - remember, that's what the servers are sending. So any time you're hooking up to a web server, part of that SSL negotiation is here's my public key. Use this to send me stuff that nobody else can decrypt. That's how it works. So all the SSL connections, all the HTTPS connections everywhere are proudly broadcasting their public key. And when you look in your web browser, if you have an HTTPS connection, when you check the security certificate, one of the things listed there proudly is the public key. That's out there in the open.

Leo: Well, I use, yeah, I use PGP to sign my email. And the beauty of using PGP to sign your email is I literally can offer for download my public key. Anybody can have it. You want it, I'll give it to you. It doesn't help you in any way in decrypting my email. It only lets you encrypt to me.

Steve: Now, what they found was an unexpectedly large number of collisions. Essentially, they found instances, and with no apparent common affiliations, that is, it wasn't like all the bad keys came from one particular certificate authority, one CA was issuing these. Nor was there a correlation over time. They looked at the issuing dates and the expiration dates. So it's not like they were bad and they were getting better, or they were good and they got worse. There just - there wasn't a correlation, a variation over time or source. But there were collisions in the prime numbers that were used to generate the public keys.

And what this means is, the reason this is a problem is - take GRC. Now it happens I've got a 2048 public key because I recently, as we all know, switched to an extended validation EV certificate when I moved from VeriSign over to DigiCert. And so I'm in better shape. Largely the collisions are occurring in the larger population of 1024-bit keys. But so you take a website, a webmaster who knows what their public key is. Well, it turns out that there is a much greater chance than we expected of being able to find another website out there somewhere with the same public key. No one ever looked for them before. It turns out there's collisions. And so the problem is, if I were to find

another website with the same public key as mine, I'd know their private key.

Leo: Oh, that's not good.

Steve: No.

Leo: That's not good.

Steve: No. So that's what this means.

Leo: I wonder if the same holds true for PGP, I mean, because...

Steve: Yes.

Leo: And I'll you why this is an issue, because the PGP key servers hold millions of people's public keys. What you do is you upload your public key to this directory.

Steve: Yes.

Leo: So it would be trivial to download and search it.

Steve: Yes. And their paper, I don't think I tweeted this recently...

Leo: It's at eprint.iacr.org.

Steve: /2012/064.pdf.

Leo: Oh, that's easy.

Steve: Yeah.

Leo: Well, it's easier than a 16-bit prime.

Steve: Yeah, it actually is pretty easy. Anyway, their paper, they address PGP explicitly because, as you say, Leo, there is such a huge database of existing keys. And there were PGP collisions, as well.

Leo: Oy. Oy.

Steve: Yeah. So, I mean, this isn't the end of the world. What they found...

Leo: [Sputtering]

Steve: Yeah. What they found was...

Leo: I'm scared. But I think I use a passphrase to unlock my PGP. Wouldn't they need the passphrase, even if they had the private key?

Steve: I haven't looked at it closely enough to say.

Leo: Oh, boy.

Steve: What I can tell you is that two out of every thousand...

Leo: Yikes.

Steve: I know.

Leo: That's too high.

Steve: That's too high. Two out of every thousand. So that means it's 0.2 percent are insecure.

Leo: That's way too high. So all you'd have to find is a match in the PGP database.

Steve: Yup, and then you would know that person's private key and be able to spoof...

Leo: Ah, but, you know - so you generate - okay. So I'll generate a key. I know the private, and I know the public. Now I search for a match for that public key. And let's say there's a million keys in the database, which there probably are, that means, what, I have several thousand private keys that I can - messages I...

Steve: No, no, no.

Leo: No?

Steve: No. This also - remember that we also had the birthday attack phenomenon going on.

Leo: Yes.

Steve: Remember that the birthday attack is not my birthday collides with one other person's, but in a population, somebody's...

Leo: Yes, got it.

Steve: ...two birthdays collide.

Leo: So it isn't the case that, if I just generate a key, that 0.02 percent or whatever it is will match my key.

Steve: Right.

Leo: Oh, then I'm not so worried.

Steve: Right. It's that there is - but between...

Leo: There'll be a collision, yeah.

Steve: ...all of these 7 million websites that they analyzed, or rather 7 million public keys, they found a statistical anomaly. There should never have been that many collisions. And what that told them was, and here we come back to it, what that told them was the random number generators which are being used to find the primes, are not being seeded correctly. They're not being seeded with sufficient entropy. So, I mean, looking at the technology, they don't think that the random number generators are bad. But remember, we've talked about this extensively, it is very difficult to get really random data out of computers because they're all about not being random. They're just math boxes. Every time you put in the same stuff, you're going to get exactly the same stuff out. And so lots of work has been done in trying to make these things random.

And I think what we're seeing is that certificate authorities ought to be one place where huge amounts of money is spent on generating much higher quality sources of randomness. They shouldn't be relying on moving someone's mouse around in a circle for a while, which is what TrueCrypt tells us to do in order to get sufficient randomness for our hard drive key. I mean, these guys ought to be listening to Geiger tube sparks and watching lava lamps move and put video cameras of mice in mazes, I mean, everything they could think of at the same time to suck in...

Leo: Entropy.

Steve: ...as much entropy as they can find. And then aim a camera at static on an old-style TV with its antenna disconnected and digitize that. I mean, just everything you can think of. This is where we want money spent on generating the highest quality random numbers possible. So what happened is the researchers notified the sources of the archives, and they have been taken offline because they would be too useful for bad guys. And where possible, they have notified the owners. Where they were able to backtrack, they've notified the owners of the worst, the least secure keys, and because certificates can have things like email addresses in them. They don't have to, but they can. So they've tried to be responsible. They've also not issued the more - I'm tempted to make up another word now.

Leo: Smartage? More smartage?

Steve: The more disclosable - they have two versions of their paper. They've got a tamed-down one, which is the only one that's out today, where they explain this; but they also say, uh, there's more that we're not saying yet because we need to gently introduce this problem to the industry so that there's some opportunity for people to fix this problem. And presumably they've got a more in-depth paper, we know they do, where they explain in more detail what they've done. The problem is, for the really smart bad guys who just get off on reverse engineering these tidbits of information, they've really got all they need.

So anyway, this isn't the end of the world. The sky's not falling. I don't even really think that there's anything individual website operators need to do. They say in their paper that they considered producing a service where website owners could submit their certificates and check them to see if they're vulnerable or not. But the problem is there's no way to prevent that from being abused because then the bad guys would collect public keys, just harvest them from the 'Net and run them through and see whether they're vulnerable or not.

So I'm glad that I moved up to 2048-bit certificates recently. 2048-bit certs are not invulnerable, but there's vastly lower collision incidence because there aren't that many of them yet relative to the 1024-bit keys, which are more vulnerable just because of, like I said, the birthday attack problem. There are just so many more opportunities for two of them to collide.

Leo: I do have to reiterate what I said before, and I was really glad you clarified this. And maybe it's difficult to understand if you're not a math wiz. But there's a significant difference between me generating a number and two out of a thousand other keys matching, and two out of a random thousand matching.

Steve: Correct.

Leo: Significant difference. So much so that it's not probably a cause for concern. It doesn't mean that a bad guy could then generate a PGP key or an SSL certificate and

then look in the database for matches and find that many hits. He won't.

Steve: Right.

Leo: I don't know what the actual number is, but it'd be significantly lower.

Steve: Yeah, in fact it's much more likely that, if you took a key and checked it, there would be no collision.

Leo: Right.

Steve: That is, no one given key will collide. And that's what the birthday attack - that's the reason it's such a surprise to people.

Leo: It confuses people, yes.

Steve: Yes.

Leo: So it's an old - it's a great probability thing where you take, I think it's 36 people in a room. The chances are that two of them, some two of them will have the same birthday. You would think it would be 365. But that's different than if you said how many people would have to be in a room to collide with my birthday. Very different number.

Steve: Correct.

Leo: Very different number. So I don't know if that clarifies things or muddies them, but that's a significant difference.

Steve: Yes. So we need not panic.

Leo: 23 people in a room. Okay.

Steve: Yeah, 23. And, yeah. And so any two will have the same birthday.

Leo: Right.

Steve: As opposed to the probability of how many people in the room to have the probability be at least even that your birthday would be shared with somebody else.

Leo: My birthday, right.

Steve: Much larger number, yeah. So anyway, so not the end of the world, not a cause for any great alarm. We'll see how this evolves. The EFF and their SSL Observatory have been working with these guys. They generated a fresh restart of their observatory at the beginning of the year, so it's only four or five weeks old at this point, or I guess maybe six weeks old at this point. And they've already seen that this problem exists in all the certificates they've collected since.

So, I mean, this is a problem. The good news is it will end up being - it'll drain out of our currency in the same way that I'm glad that certificates expire, which is certainly the certificate authorities who are issuing these certs, this is a big deal for them. They absolutely need to focus their attention somewhere they haven't before, which is on the entropy of the primes they're using to generate certificates. That's where we need to have this fixed. Then, within two or three years, the certificates, all the certificates in use now will have expired and have automatically been replaced with ones with ultra high entropy, fully high entropy prime numbers. And then we'll be okay.

Leo: It's not much of an attack.

Steve: No.

Leo: All right. Just checking.

Steve: So I meant to mention this last week, and it wasn't in my notes. And of course if it's not written down, I'm not going to see it.

Leo: Doesn't exist.

Steve: But we did get information from Symantec about the mistake that was made in pcAnywhere. And it's so classic that our listeners, their eyes are going to roll back in their heads. Okay. Description.

Leo: Yes.

Steve: "Symantec has released a patch to address a vulnerability in its pc" - and this is quoting Symantec - "address a vulnerability in its pcAnywhere product. pcAnywhere consists of a server and a client that allow a user to connect to a computer and control it remotely. The server component accepts requests for authentication on port 5631 and copies the user-controlled" - that is to say, the user-provided - "username in these requests to a fixed-length, 264-byte buffer. By sending an overlong username, an attacker can exploit this buffer overflow in order to execute arbitrary code on the target's machine with system-level privileges."

Leo: Wow.

Steve: It doesn't get any more cut-and-dried than that. I mean, this is so brain dead. This is so security programming 101. So they allocate a short, 264-byte buffer. Then they accept a username of any size and blindly copy it into the buffer. And the buffer is probably allocated on the stack, which means that they - and in the past we did a podcast and, in excruciating detail, explained how, when the buffer overruns, it copies over the history of things like where you were called from, so that when this routine which is doing the checking attempts to return to its caller, it takes that information from the stack, which was stacked before the buffer, which means you're able to, if you cleverly design this data, you're able to cause the execution to return to your own data on the stack. If the stack is protected from being executed, there are still ways you can cause the stack to return you to other code in the system which you reuse for your own purposes.

So it can be made more difficult. Over time it has been made more difficult to exploit stack-based code. But these are very powerful, very flexible computers; and pretty much where there's a will, there's a way. And so this all starts with a system that accepts a response from the user without checking its length and just copies it into a buffer without making sure it doesn't overwrite. And that's what's going on with pcAnywhere. So anyone who's using it definitely wants to update to the latest version.

Leo: Yes, yes.

Steve: And I should mention also, we talked about this long ago, but port 5631, this is a classic instance of where you absolutely don't want to use the default port because 5631 is something, for example, that I've checked at GRC for users to make sure they didn't have that open. You definitely want to change the port number of your own instance of pcAnywhere to something else, anything else, because bad guys are out there scanning the 'Net now, looking for anything that answers on port 5631. Chances are it's going to be pcAnywhere, and then they'll exploit this in order to run code on that system, that pcAnywhere server. So you definitely don't want to leave that default port.

Also in the news, I'm sure you saw this, Leo, was all the kerfuffle that arose over Google's Wallet PIN.

Leo: Yes.

Steve: What was disclosed was that there was a vulnerability in jailbroken versions of some smartphones that had the Google Wallet installed on them. And I think there was another requirement such that it was actually only two of Google's phones where this was actually a vulnerability. And the problem was that there's a four-digit PIN which protects the privacy of the data stored in the wallet, things like your credit card number, which you'd like to have privacy protected. And what Google does is they do an SHA-256. Now, that's a hash, as we know. That's a very, very good hash.

The problem is they store it in the phone. That is, they don't take the PIN and go check in with some central authority somewhere. They wanted to avoid the need to do that. They wanted to allow you to unlock your wallet locally, without checking in. Well, all of

our crypto-savvy listeners understand the problem of that. It's like, okay, that can't be secure. It can't be infinitely secure. You enter a four-digit PIN, which you hash into a 256-bit token. But then you've got to check it against the token which is stored in the phone. So if it's jailbroken, and the bad guys have access to the database, then they've got your SHA-256 hash of your PIN. And here's the problem: It's only four digits. Which means it's easy to brute force. I'm sure there are tables of all, what are there, like, 10,000 only.

I mean, we talked about the problem of four digits not long ago with the whole WPS problem. So what this means is that anybody who, if there's malware in a jailbroken phone, it has access to your Google Wallet's SHA 256-bit hash. It can get it and instantly, or quickly, figure out what the matching four digit PIN is. Then they've got the four-digit PIN for your wallet. And that's not good.

Leo: Not good. Very not good.

Steve: Yeah. So the problem is that it's being stored local. And Google says, well, we told you jailbreaking is not safe. And they're right. So I don't really hold Google at fault for this. There's really nothing they can do if they want to allow you to unlock your wallet without sending off your authentication somewhere else. Now, they could use something much more sophisticated, much more complex than four digits. One of the weaknesses here is that it's just a four-digit PIN, so it's easy to brute force all 10,000, 0000 to 9999. There's just not that many. You do the SHA-256 of that, and so you have a lookup table that converts that back to the PIN. And so that's the problem. So if users used much more complex, were allowed to use much more complex passwords for their wallets, then that would secure them against this particular attack. But it's still going to be a problem. So I would say don't jailbreak.

Leo: Okay. Huh. But Google turned off the NFC capability in the wallet until they fixed this. So there will be a fix; right? Or is this not possible? Is this in the design of it?

Steve: I don't see how it's possible. If malware has access to that database, which jailbreaking it affords, then you've got a problem.

Leo: Too bad.

Steve: I would hope maybe what they'll do is give you the option of better protection than four digits. Four digits is just not enough.

Leo: Longer PIN. And five digits would be a lot better. Six digits, I mean, couple more digits...

Steve: Well, I would get away from digits.

Leo: Digits is a bad choice, yeah.

Steve: Yeah, because it restricts your alphabet to a 10-character alphabet. You'd like to just be able to use, for your wallet, what we all know as a strong password. Use the Haystack technology, the idea being that you want to, I mean, first of all, if you've got malware in a jailbroken phone, I don't think I would...

Leo: Well, you're screwed for so many other reasons, anyway.

Steve: Yes. So I don't think anyone should be using any sort of local - and no one should be relying on local encryption on a jailbroken device because it does allow malware access to much more of your device than is safe. So if you want to play with a jailbroken phone, fine, just don't store anything important there on that same phone. Get another toy to play with.

Leo: I need a clarification. "Jailbreaking" is a term used on the iPhone. We're talking about Android phones. We call it "rooting." And, now, there's two things. On an iPhone, jailbreaking means you can get apps from third-party sources. You do not need to do that on Android. It's just a checkbox. Is that sufficient? In other words, just downloading malware, is that sufficient? Or does the malware have to have additional permissions?

Steve: Now I'm confused. So I hope I haven't given everybody a bum steer.

Leo: Well, it's just it's unclear from the terminology they use, not you.

Steve: Yeah, okay, because I was sure that what I was seeing was that people were doing something to their Android devices, so I guess rooting them is...

Leo: It must be that rooting because, once you root your phone, an application then can ask for superuser permissions. And I would guess that it would be the combination of downloading from a third-party source some malware and then allowing that malware to have root permissions. I should tell you that you are notified when an application requests and is given root permissions. I don't believe they can do that silently.

Steve: So it could just be - it could be an application which you don't suspect is malicious, that you voluntarily give permission to because you want whatever happy service it's going to give you which needs root. And then in fact behind the scenes it's able to access things. And so that's just a dangerous thing to do.

Leo: And there is a log, there's a program that you get when you root a phone called Superuser that gives a log of all apps that have requested permissions and

what permissions they've requested and so forth.

Steve: As long as one of the permissions is not rewriting the log.

Leo: Well, that's true, too. There may be an issue there, yeah. So it sounds like the advice, then, is do not root your Android phone. Or, if you do so, be very careful about what apps you put on there.

Steve: Exactly. If you're going to root your phone, then you really don't want to rely on encryption of sensitive data on that phone.

Leo: I routinely use - root almost all my Android phones because there are apps, backup apps chiefly, that you need root access for backing up, Titanium being one.

Steve: Right.

Leo: Okay. That's good. This is all good information. Sounds like it's not fixable.

Steve: I just don't think it is. It's exactly like the problem of the movie industry trying to encrypt DVDs. We've got to decrypt them.

Leo: Right, the keys have to live, yeah.

Steve: Yes, the keys have to be there. And so...

Leo: In the clear.

Steve: ...I just don't see how it's solvable except what you really should have is a much stronger password than just the four-digit PIN. I'm surprised that's all they have, actually.

Leo: Yeah. On the phone itself, Android gives you many choices of both alphanumeric - and they have a gesture-based password which I use. That would be, I presume, stronger than numeric.

Steve: Anything would be better than four digits.

Leo: Okay. So this might be a fix that Google could implement or somebody could implement with the Superuser app. You know, the Superuser app can be set to block

access. You have to explicitly give permission each and every time an app requests it. You might then ask for a gesture, for instance, to unlock it. Which I think would then certainly increase security significantly. I would think, but I don't know. All right. Sorry, I didn't mean to slow you down. I just wanted to get some clarification on that because I routinely turn on root.

Steve: Are you a Google Wallet user?

Leo: I am not because I don't have NFC. There's only a couple of phones that have NFC currently. I guess you could use Wallet without NFC. So anyway, but I don't, no, I don't Google Wallet. I use a similar thing, though, that might be similarly crackable, from Square. It's called Card Case. It's wonderful. I walk into my local coffee shop, I make an order, and then I say put it on my tab, and they have an iPad with my name and picture on it, and they say, good, we'll add it to your tab because I've been sensed by geofencing to have come into the store.

Steve: Wow. That's cool.

Leo: It's really cool.

Steve: Yeah.

Leo: However, it probably is even more susceptible to these kinds of attacks.

Steve: Okay. So I tweeted something that I thought was interesting, just because it caught my eye, which was someone did an analysis of the speed of browsing with and without web tracking. And they used standard tools like NoScript on Firefox. And in fact, after this, somebody told me they were playing with ScriptNo on Chrome and verified for themselves the same thing, which is not protecting yourself from web tracking, not blocking trackers, slows your browser load time often by as much as a factor of two. So popular websites, which are just loading down your browser with all kinds of junk, are really, I mean, they're not only causing a privacy threat, but a serious performance bottleneck, I mean measurable, and as much as a factor of two, sometimes even more.

So Patrick Dubois tweeted the note to me, and so I wanted to thank him. And I tweeted it out, and it generated some attention in my Twitter stream. So I wanted to make sure our listeners knew that, not only is all this blocking we talk about doing good for you from a privacy standpoint, but it really does accelerate your use of the 'Net.

Moxie Marlinspike, our wonderfully pseudonymed hacker, introduced just a day or two ago a new password-cracking service called CloudCracker.com. Maybe you can bring it up before everybody who's listening to this does that, Leo.

Leo: Good point. I'm hurrying. Not right now. CloudCracker.com.

Steve: CloudCracker.com.

Leo: Can I type faster than a TWiT listener? Apparently I can.

Steve: Maybe he's got a lot of horsepower there. It's a very nice-looking site. And it's just about as disturbing as anything could be.

Leo: Oh, lord.

Steve: It's not free. But I think it's $17 an hour? For some reason that number sticks in my mind.

Leo: Yeah.

Steve: I don't have it in front of me.

Leo: It's "Run your network handshake against 300 million words in 20 minutes for $17."

Steve: Yeah. So you capture somebody else's traffic out of the air, which is to say the network handshake, into a file, a capture file. You upload it to Moxie's new cloud-based...

Leo: Holy cow.

Steve: Uh-huh, supercomputer.

Leo: It's quite nicely done. It's a very Web 2.0 site.

Steve: Very attractive, yup.

Leo: Wow.

Steve: So this is cloud-based password cracking. And for not much money he will crank, he'll turn loose, I don't know if it's using EC3, Elastic Cloud Computing from Amazon, or what service, or how many...

Leo: Probably, yeah.

Steve: But he fires up a bunch of technology to work on cracking the password by doing

brute force dictionary attack against it and lets you know what the password is, if you...

Leo: Awesome.

Steve: Yeah.

Leo: In case you forget.

Steve: Just FYI, yeah. CloudCracker.com.

Leo: Yay. Thank you, Moxie.

Steve: And then Twitter stepped their protection up another notch. We covered some months ago when they had offered SSL optionally. They just announced in a blog posting a couple days ago that HTTPS connections are now the default for Twitter when you're on their website.

Leo: Right on, daddio.

Steve: So yay and bravo.

Leo: Yeah.

Steve: Another real, I mean, we need to see that universally. And as soon as I get some time, I'm going to do that myself.

Okay. Now, I tweeted the URL for this video. And it's not easy to say it live, so I won't try to. It's a YouTube video - so just go to Twitter.com/SGgrc, and it's only a tweet or two back, so you should find it easily - about an amazing-looking spray-on antenna system.

Leo: This is a Solve for X series that Google does. I think it's Google that does this, WeSolveforX.com.

Steve: Very professional.

Leo: Lot of great stuff on there.

Steve: Yes. And...

Leo: So this is a fascinating - and by the way, don't be put off by his presentation because he's a little shy and stumbling, but...

Steve: Yeah, well, he's apparently the engineer...

Leo: Yeah, he's a geek.

Steve: ...who was involved in this. He's a geek. And so in my tweet I said something about the guy's not the greatest presenter, but at about five minutes into this video your mouth will fall open, and it will stay open for the rest of the video.

Leo: Now, I know Solve for X is legitimate, and they have great people like Nick Negroponte and Neal Stephenson and just wonderful - it's kind of like TEDx, in a way; right?

Steve: Yeah. It's very TEDx-like. And they had the Google guys on and, I mean, so it's clearly - this is not a hoax.

Leo: Right. But I've got to ask this - it doesn't seem possible.

Steve: So what he's got is a spray-on material with nano capacitors. And there's a close-up microphotograph of it. And it is true that capacitors conduct AC and block DC, and radio frequency is AC. So this would be a conductive surface under the influence of radiofrequency. And...

Leo: So what they're doing is they're painting this stuff on trees, buildings, all sorts of stuff. And it turns it into an antenna.

Steve: Well, and I've got to say, Leo, the picture of the wires connected to the tree trunk...

Leo: That's a funny one. Yeah.

Steve: Oh. Now, but the guy is very deadpan. If we're to believe the video - and we all have to have some healthy skepticism here. But, I mean, if it's a hoax, it's the most beautifully perpetrated hoax I've seen in a long time. I don't think...

Leo: He claims that this tree thing was an army, a U.S. Army test. They're doing this for the military.

Steve: And he talks about 20dB of gain. That's a factor of a hundred. I mean, 20dB is

nothing to shake a stick at. And they talk about having duplicated their measurements, and the army was involved, and used the army's antenna, and then they sprayed a tree.

Leo: Here's the - somebody said, well, gee, Apple should hire this guy for the iPhone. In fact, they did test it with an iPhone.

Steve: Yes, and they sprayed an RFID tag, and it went from something like five inches to 700 feet or something. I mean, watch the video. I know this seems wacky. But as I said to you in some email, I hope this isn't another cold fusion sort of thing. But it seems just barely tantalizing enough.

Leo: It really feels like a hoax. But so what we did is we passed this along to George and Bob and Gordo at Ham Nation because, if there are any antenna experts in the world, it's them. And we're going to find out what they say. And they're going to try to get the guy on the show. Now, is he selling this stuff?

Steve: Well, okay, there is a picture of a spray can. And that was definitely just meant to be sort of...

Leo: A joke.

Steve: Like what it could look like. I'm sure they're not at the spray can stage yet because it looks sort of retail. But if you could spray on nano capacitors, I mean, it is so convincing looking. If nothing else, enjoy it as a spoof because, if so, it's beautiful done. If not, and if the claims are true...

Leo: Well, it's revolutionary.

Steve: We have a revolution in antenna technology.

Leo: I mean, his point, in the speech he says what if, instead of cell towers being these ugly boxes on buildings, you just sprayed the side of the building, and that's the cell tower, and it worked better?

Steve: Well, and he even mentions that the efficiencies are so high that you can get power out of the air, from just all the RF floating around. And so you might be able to create self-powered...

Leo: Oh, wait a minute.

Steve: ...towers.

Leo: Now, wait a - hold on there one cotton-pickin' minute. That sounds like the Robert Heinlein story where they had these antennas waving in the air, collecting energy from the outside world.

Steve: Well, remember that Tesla did transmit power across the globe. He created lightning on the other side of the Earth from his big tower in Colorado by resonating the Earth's electromagnetic something or other. So, I mean, that was something Tesla was known to have done back then. So who knows.

Leo: I can't wait to find out more. I'm really - I'm fascinated by the whole thing, and we're going to try to kind of narrow it down and track it down. The problem with military, he has a good excuse for why he can't talk anymore about it. And remember supercapacitors, which are real, we talked about them a lot, but there was...

Steve: Well, it was nano capacitors.

Leo: No, but yes, I know, but remember we also talked about supercapacitors and how they exist. They're not real. They're not a hoax. But there sometimes is a bit of a gulf between concept and implementation.

Steve: Yes. There's a new supercapacitor-powered mouse also, by the way. And we know that there's supercapacitor out...

Leo: I have a screwdriver.

Steve: Yup, screwdriver, flashlight. And so we're beginning to see these things happen. It does take materials breakthroughs, typically, in order to make this go. This would be neat if this were true. And lord knows we could use a revolution in antennas because, I mean, so could Apple with the iPhone after that fiasco with the antenna on the outer edge that we talked about extensively back then.

So, okay. And last week follow-up, I had here. Someone sent me a note that I ran across in the mailbag saying that was an awesome episode. "What had mostly prevented me from switching permanently" - oh, I know why I wrote this, or why I picked this up. "What had mostly prevented me from switching permanently from the memory/GDI object-leaking Firefox to Chrome for a long time was the lack of a NoScript extension for Chrome. Now if only there were a decent tree-style tab-like extension for Chrome, I would be totally set."

Leo: Are you sure you didn't write this?

Steve: I have, for the most part now, switched to Chrome. And so the reason I - if there's anybody listening. Now, the problem is I don't think that's possible. I'm the same way. As we know, I'm a tabaholic. I mean, I'm a confessed tabaholic. I've got, looks like

about 30 open right now because they're over on the side, and they're small, and they're nice. Now, you can, on Windows, move Chrome's tabs to the side, but they're still big. They take up much more space than they should, so you can't have as many of them. So anyway, I'm hoping that Chrome will evolve over time. I know Google loves the way it looks, and they're very proud of how clean it is. But there are people who just live on tabs, and I'm one of them. So we need - I agree with this guy completely.

Leo: Wait a minute. Wait a minute.

Steve: If they could do small tabs on the side, that would be wonderful.

Leo: Is this - okay. Get ready for this one. This comes from the chatroom. The guy who was talking about spray-on antennas was from a company called ChamTech or ShamTech. Right? They're selling this. Spray-on antenna kit. Call for pricing. This is - all right. Somebody order that and tell us how it works. That's got to be a joke. ShamTech; right?

Steve: Careful where you spray it, though.

Leo: You know what, this is a joke because I don't see a phone number next to the call for pricing. Is this him? Is this the same guy? Yeah, they have self-spray-on kits. There is a phone number. I don't know. Somebody order that.

Steve: I wonder if...

Leo: ShamTech?

Steve: I wonder if, yeah, that does sound a - okay. But if you were going to be a spoof, why would you call yourself ShamTech?

Leo: Right. Well, and Solve for X was fooled if - I don't know. Somebody order this.

Steve: Or maybe the Solve for X video is a spoof.

Leo: No, no, no.

Steve: That this is not...

Leo: No. I don't think so.

Steve: I wonder if it's actually on the Solve for X website.

Leo: It is. It's on the Solve for X - it is, I think.

Steve: Well, I looked at Solve for X and saw a lot of other cool things. And it's under YouTube.

Leo: Oh, maybe it's somebody else. Let's see. Let's see if we can find this guy. Now, there it is, it is on the Solve for X website. This is fascinating. It doesn't feel right, and yet I've got to call them and order - you can order it.

Steve: So I guess the spray can is real.

Leo: It's real.

Steve: Anyway, everybody look at the video. Is there an easy way to find it in Google, Leo, like "spray-on antenna," if you Google or go to...

Leo: You can go to their website, ChamTechOps.com, or SolveforX.com. Either way, you will be able to find this video of Anthony Sutera, who is their CEO and an entrepreneur in communications.

Steve: Well, he sure is.

Leo: Wow.

Steve: He definitely, whether it's RF communications or spoofing communications...

Leo: I don't know what to say.

Steve: It's so tantalizing.

Leo: I want to get it. Omri, who is also a speaker at Solve for X, is going to come on one of our shows. He's a fan, and he's been to the studio. So it's real. It's a legitimate Google event.

Steve: Solve for X is real, yes, yes.

Leo: So I don't know.

Steve: Wow. Anyway, we will follow up and keep our listeners informed.

Leo: We've got - HiWEB in our chatroom is ordering it right now. Let me know the price, HiWEB, will you? All right. Moving right along.

Steve: So we're actually on a very related topic. I just wanted to give our listeners an update on the LightSquared/Global Positioning System band spread collision. The fight continues. We'll remember that LightSquared, that's a commercial company wanting to build a next-generation 4G wireless network in the so-called "L-band" spectrum, which unfortunately, inconveniently, lies adjacent to the frequencies used by the Global Positioning System. Due to L-band's close proximity to the frequencies used by GPS, about half of the frequencies that they plan to use, that LightSquared plan to use in their network, have been shown to cause interference on some GPS receivers.

Now, as we mentioned last time, this is partly the fault of the fact that the GPS receivers aren't very narrowly tuned, that is, their own reception isn't tightly tuned and constrained to the GPS frequencies because there wasn't anybody nearby. And so in terms of the frequency response, they were able to use less expensive, less narrowly tuned receivers to save money. And what's called the "skirts" of their response curve falls out into this L-band, which had previously been unallocated and unused. Anyway, today, which may have been yesterday, the NTIA, the National Telecommunications Information Administration, sent a letter to the FCC which declared the interference to be unavoidable.

Quote: "Based on NTIA's independent evaluation of the testing and analysis performed over the last several months, we conclude that LightSquared's proposed mobile broadband network will impact GPS services, and that there is no practical way to mitigate the potential interference at this time."

Well, you can imagine that LightSquared Corporation promptly replied. They said, quote: "LightSquared profoundly disagrees." This is sort of how...

Leo: Of course they do.

Steve: This is like what's his name, Mitt Romney is a "severe conservative." "LightSquared profoundly disagrees with both the NTIA's and the PNT's" - I don't know what the PNT is, anyway, somebody else's - "the PNT's recommendations, which disregard more than a decade of regulatory orders and, in doing so, jeopardize private enterprise, jobs, and investment in America's future." Do we have any music you can play in the background, Leo?

Leo: [Humming "America the Beautiful"]

Steve: "NTIA relies on interference standards that have never been used in this context and were forced by the GPS community in order to reach the conclusions presented today. This, together with a severely flawed testing process that relied on obsolete and niche devices, shows that the FCC should take the NTIA's recommendation with a generous helping of salt." Actually, I don't think them using this kind of humor in their communications is useful to them, but, you know.

"Despite LightSquared's success in finding technical solutions and the acknowledgment

by a senior government official that GPS receivers are specifically designed to rely on spectrum licensed to LightSquared" - which that refers to the sloppiness of the GPS receiver design - "it is extremely disappointing that this recommendation was made today."

So I don't know what's going to happen. I mean, we have a situation where band that should be usable, can't be, due to the fact that an existing install base of GPS receivers are caused pain when this band is used. So there you go. I think LightSquared's going to end up having a problem.

Leo: Too bad, because of course the idea of ubiquitous wireless Internet is fantastic.

Steve: It is.

Leo: And we're all for that. And I had hoped they had come up with a good solution.

Steve: Yeah. Speaking of a good solution, Nick Bowen dropped me a note which I saw as he wrote it on Sunday, February 5th, so just a couple days ago. From Walnut Creek, California, he said, "A short time ago a friend brought me his computer for me to run DBAN" - which we've spoken of, Darik's Boot And Nuke - "on prior to him getting rid of it. It was an old Win98 machine, and DBAN would not run because the hard drive would just grind. He had purchased SpinRite" - oh, I, says Nick, "I had purchased SpinRite a couple of years ago and had not used it yet. But I thought this would be a great opportunity. I ran it on Level 2, and it fixed the drive. This, then, allowed me to securely wipe the data before it was given away. Thanks for the podcast and great product. Nick." So yet another use for SpinRite. You run it before you wipe your drive. And in fact, at some point, SpinRite will do that for you, but not quite yet.

Leo: Oh. We're looking for SpinRite 7?

Steve: I think it's going to be 7. Maybe 6.1. But 7 for sure. I did trademark "Beyond Recall," which I just love. And for a while I was going to do a separate thing. But I thought, eh, that's dumb. It's just so, I mean, it's everything SpinRite's doing except just use good source of pseudorandom data to wipe the drive, and in the process fix any other problems, and there are actually some tricky things that SpinRite can do also.

Leo: So at the same time, kind of.

Steve: Can do at the same time. So to do a better job than Boot And Nuke, do a much faster job, leverage all the same technology, and make it very clear to people that, if they proceed, their data will not be there.

Leo: It's gone.

Steve: It's beyond...

Leo: Beyond recall.

Steve: Gone for sure, beyond recall, exactly.

Leo: We have 10 questions plus, as you say, the Bonus Unintended Consequence of the Day. Let's start with Gareth in Germany. He actually asked this on Twitter, hence its brevity. @gazz2010 says: @SGgrc Steve, love Security Now!. Wondering if a NAT router without SPI is any less secure, would you say? Many thanks. What is that? I don't know what SPI is.

Steve: Stateful Packet Inspection.

Leo: Oh, of course.

Steve: And my sense is, Gareth, no. Now, I ought to mention that I wanted to respond to Gareth, and I tried to. But he's not following me on Twitter. So he got the message to me by mentioning SGgrc, and that I saw. But the only way for me to respond to him would be if I mentioned him. And the problem is, with 20, I don't remember now how many, 27,000 or something followers, I would be cluttering up my stream if I...

Leo: Ah. I can give you a solution on that.

Steve: Oh, really.

Leo: Yeah. And this is complicated. But Twitter, if you just reply to an @...

Steve: Reply to a single posting, huh?

Leo: Yeah. If the very first thing in your reply is @hisname, it will only be seen by him and those who follow you both, the intersection of your followers and his followers.

Steve: Oh.

Leo: Which, yeah, which presumes that those - and I think it makes sense, if you think about it, because those people saw his question to you because they're his followers. So in effect you're having a conversation with people who are already in on the conversation.

Steve: It does make sense. And in fact I have wondered, I've seen dialogues occurring in my Twitter stream through @ mentions. And I've thought, how are these people all

seeing each other? I figured they must be looking at my stream, but in fact they're just following the people who are sending...

Leo: It's the intersection, right.

Steve: Ah, nice.

Leo: Now, to get around that, people sometimes will put ".@" and then...

Steve: And I've seen that, too.

Leo: Yeah, and the reason they do that is now this is visible to everybody in your stream, whether they follow the original poster or not. Or, and sometimes I'll do that, too, I'll put a sentence fragment ahead of the @. I'll say "Good question, @. The answer is..." if I want everybody to see it. And usually I do that because most of the time what I'm talking about is of general interest. But if you were just having a conversation, just begin it with the @ and that person's name, and it will only be visible to the intersection of your followers and his or hers.

Steve: Ah, perfect. I love that.

Leo: It was a clever thing Twitter figured out. And it's not well documented. And I think you just have to learn it.

Steve: It's brilliant.

Leo: Yeah. And you're right. You can't DM somebody unless they are already following you, another way that they protect you a little bit from spam DMs.

Steve: From being spammed.

Leo: Right.

Steve: Absolutely. So to answer Gareth's question, now that we've all learned something new about Twitter - thank you, Leo - my sense is paying for SPI buys you nothing worth the money. A NAT router is stateful packet inspection. I mean, that's what it does. The reason you don't get Internet background radiation flooding into your network is that only packets that are expected are allowed back in. So, and that is state. That is the router holding state. So if you send a SYN packet out to a remote server to open a TCP connection, the SYN/ACK packet responding to that is allowed in because your router remembers that pending outgoing SYN. It interprets the packet and statefully remembers what's coming back.

Now, it's true that there are abuses of protocols which additional stateful packet inspection could theoretically help prevent. But we've all now also got firewalls on our personal computers. Not only are we protected out at the boundary of the Internet and our network, but every individual computer within our network has protection. So my sense is, if it's there, and they use it as a marketing term mostly, then it doesn't hurt you. My sense is, though, any NAT router is providing you with stateful packet inspection, effectively, and it's never been clear to me what more you actually get.

Leo: Well, doesn't SPI look into the content of the packet?

Steve: Yes. So does NAT.

Leo: NAT actually looks at more than just the port and the routing?

Steve: Well, it can, for example, in order to NAT FTP. FTP protocol requires that your server respond, that you understand that it's an FTP connection being established because FTP uses two ports, and the server will be coming in over a different port for your data than over your control channel, your command channel. So there is more going on than just IP and port number.

Leo: All right.

Steve: Anyway, I mean, if they would ever say what it is they do...

Leo: They don't, of course.

Steve: ...then it's like, okay, I would believe them. But all they do is say SPI and expect you to pay more for it. It's like, well, unless you tell me what it is, I'm not buying it, literally.

Leo: Right. Question 2, a deliberately anonymous user somewhere near Yakima, Washington says: A few years ago I was a treasurer for a martial arts club. All treasurers were given a website login that changed every year. However, I noticed, if I memorized the URL path after logging in, something like martialartswebsitename.com/treasureradmin, I could simply bookmark that and always access the administrative materials without having to first use a username and password. This is somewhat akin to the problem you described with the TRENDnet cameras and the CGI script. My question is, how do websites protect against people bypassing login pages in this fashion? Could attackers simply bypass the whole username and password attempt by simply brute-force guessing the URL post login? Thank you.

Steve: Well, in theory. It worked for the TRENDnet cameras. It clearly worked for this martial arts website.

Leo: But that's because it was poorly implemented. If you use .htaccess, your web server protects you against that kind of stuff.

Steve: Well, and the way that is done is, I mean, the right way to do it we've talked about often, and that is with session cookies, the idea being that, if you ever tried - every query from the browser is accompanied by whatever cookies your browser has for that site. And the site should, if it's a protected area, verify that you have an unexpired valid cookie, which is essentially what gives you roaming permissions within that protected area. So that's the way logins expire. That's the way, when you log out, that cookie is deleted or it's disavowed at the server so the server will no longer accept that cookie from you. You need to log in again in order for it to give you essentially a permission cookie, and then all subsequent queries from the browser carry that and permit you to go on. It's certainly possible, as you said, Leo, that really cheesy websites would do nothing but present you with a login request on the front page, then just send you to a permission page. But it's horribly insecure for exactly this reason.

Leo: Just poorly implemented.

Steve: Yeah.

Leo: So, I mean, I always used .htaccess, which is a web server control file on the server side that says this particular page is not accessible unless a person provides - so the server is doing it server side. But of course the transaction, once opened, has to continue without continuously asking for a password. So that's when it's going to start using cookies, at that point.

Steve: Exactly. You give it a valid username and password in response to the challenge of this area's protected by a password, please provide username and password to log on. When you do so and hit Send, that query goes back to the server. It looks up your credentials, verifies they're correct and, in its response to you, where it says thank you for logging on, Joe - or I guess we did have his name. No, no, he was anonymous. Thank you for logging on, unknown martial arts administrator.

Leo: John Doe.

Steve: You now have permission on the site. With that response is a set cookie header in the response which could have an expiration. It could be a session cookie where it will automatic - it will never be stored on disk, and it will just disappear when you close your browser session. Or it could be valid - we've seen checkboxes, like keep me logged in for 24 hours. And so it could have a 24-hour expiration, where it will self-expire. Or even, to make sure that's honored, the server could keep its matching credential for that cookie and make sure that it's not honored after 24 hours. So there are good, solid ways to do this.

And of course, without saying, all this is over SSL because the problem with not doing it over SSL is exactly what Firesheep was finding. It was catching those cookies over non-SSL, and that was the credential that then allowed people to impersonate others. They

were then able to just go to them, open their Facebook page as them, impersonating them using that cookie as their credential.

Leo: All right. Just...

Steve: Neat stuff.

Leo: Yeah. And .htaccess, which is a server-side file, is supported by Apache. But I bet other servers like IIS or nginx have similar server-side techniques.

Steve: Actually, there's just a billion ways to do that. If you, any time you went to a page that was protected, you could have the server using server-side stuff, see if the query for that page is carrying a valid cookie. If not, rather than delivering that page, you deliver the password challenge and make them authenticate themselves, then you give them the cookie, then they're good to go.

Leo: Question 3, Craig in Nashville, Tennessee. He wonders about the need for 2048-bit certificates: Recently went to renew a website cert for my company through Thawte. I noticed - which is owned by VeriSign now, I think.

Steve: Yup.

Leo: I noticed they're going to start requiring 2048-bit certs in about a year. Soon you won't even be able to get a 1024-bit cert from them. I would think 1024 bits is more than enough. Are they being overly paranoid? Or is there some legitimate reason behind changing from 1024 to 2048? Thanks for a great podcast. Keep up the good work. I drive three hours each week to work. Holy cow. It gives me something to look forward to while I'm sitting in the car.

Steve: Well, I referred to this question at the top of the show. And I was glad to see that there is this kind of planning happening because, as was mentioned in that paper about the collisions of SSL certificates, it is the case that academicians have factored 512-bit composites of primes. They have also factored 768, which is that next step. That's 512 plus 256. They've factored that size key. And we can assume that at some point in the future they're going to go to the next step at 1024. This problem gets exponentially harder, that is, even though 1024 is only another equal-sized step from 768 as 512 was from 768, it's still way harder. So there is no expectation that this is going to be happening soon. But we want to always keep our security several steps ahead of where the technology is for cracking that security. And so going to 2048-bit certs really makes sense.

Right now, if you have an extended validation cert, you have no choice. EV certs must be 2048. That's just part of the package. So that's where I went with mine. And I wouldn't be surprised if we start seeing 4096 before long. It takes them longer to generate and longer to handle. But machines are getting faster, too. So, and boy, at that point, we're talking serious security.

Leo: Yeah, we kind of keep parity because even though it takes more to generate these, the reason we want longer ones is because computers are getting faster, so there's kind of this kind of parity there.

Steve: Exactly, exactly.

Leo: And the good news is desktop generation of keys is faster, much faster, and growing at a much faster rate than is the decombobulation, the factoring.

Steve: Yeah. And, although, remember that it is desktop generation of keys where we're depending upon the random number generators, the pseudorandom number generators in the desktop.

Leo: We've got to improve that. We really do.

Steve: Yes, it really is a reason why it's much better to get your keys from somebody with a huge obligation and commitment to entropy, like somebody who's issuing public keys for their livelihood.

Leo: When I created my key, my new PGP key, because mine had expired, so it was time, I made it 2048 bits. And it generates it on the desktop, but it asks you to mouse around, open windows, do a lot of stuff. So it collects entropy, I'm gathering, from the activity going on in the computer, which is somewhat pseudo. It's pseudorandom, but somewhat randomized.

Steve: Yes. Although I did see in the mailbag, actually, just this morning, somebody complaining that whatever they were - they were trying to generate a certificate, and they kept getting the message that the system had insufficient entropy because apparently some other processes that were also running in the same system were draining the entropy more quickly than the system was able to generate it.

And so that's the other thing happens, is that there's this notion of an entropy pool. And the system is, from whatever sources it can, from the timing of arriving packets, from keystrokes, from mouse movements, from any sources of true randomness, it's trying to collect this. But then there are things that, like setting up - every time you initiate an SSL connection. You're not generating a public key, but you are using up entropy to choose a private key, which is the session key under SSL. So systems which are negotiating SSL connections at a high rate have a constant demand for entropy. And the idea is that the pseudorandom number generators should only run for so long before they're reseeded with fresh entropy. And that needs to come from somewhere. So there is a lot more going on behind the scenes than a lot of us realize, yeah.

Leo: Yeah, it's interesting. Jim Dwyer, Tipperary, Ireland is seeing SSL certificate errors at work, but not due to the usual causes, at least so he thinks: Great show with Leo, listen every week here in damp old Ireland and envy the Southern

California weather you two guys occasionally mention. My question is how does one exactly track down the cause of a certificate error? I ask because lately my employers' - which is a bank with very strong security policies and practices - Internet browser IE8 has been throwing up warnings about certificate errors to sites that I know have good certs. I don't get this warning when I access these sites in my home laptop or computer. I use Firefox at home, while IE8 is the bank's browser of choice, unfortunately.

For example, he gives a TD Ameritrade address. He says it generates a certificate error on IE8 at work, with the warning "The security cert presented was not issued by a trusted certificate authority." IE8 says close this page down. But when I check the certificate, it all seems normal. I don't see my employer as the certificate authority. He thought that they were doing a man-in-the-middle, I guess, as we'd talked about before.

Steve: Yep.

Leo: It shows *.tdameritrade.com, issued by VeriSign, valid from 09/09/2011 to 08/09/2012, so that all looks good; right? So what the heck's going on? Do you think my employer is intercepting SSL websites, or is there another factor at play here? There are two tabs about the certificate details and a certification path that's too long to write about here. Thanks for the great shows. I recently got an Apple TV, which is very handy for watching the show, and other TWiT shows, too, on our 46-inch screen, although my kids get annoyed when Daddy watches TV for an hour or more after school, and they can't watch "Garfield." So I think you're helping me be a better parent, too. Kind regards from Ireland, Jim. P.S.: I bought SpinRite years ago when the show came out, just as a means of payment and thank you for the great show. And I'm sure one day soon it will save my bacon. It's ready and waiting, just in case. Thank you, Jim. What a nice email.

Steve: Well, a great email and a really interesting question. So he's got IE8 complaining, saying the cert presented was not issued by a trusted certificate authority. We've never talked about this before. But my hunch, from everything he said, is that he has on that particular system a corrupted certificate authority store, S-T-O-R-E, as in storage. And that can happen. So, for example, I would go to any other machine and see if it can open a link that his machine can't, that is, within the corporate facility, in his bank. My guess is that nobody else is having this trouble, just him on this one machine; and that some corruption did occur, specifically in this case, on VeriSign's Class 3 Secure Server CA cert. And so in fact there isn't a valid certificate. Even though its wants to be valid, there's something wrong with his store. And I have seen this kind of thing happen before. So my guess is it's just localized corruption, and that it's probably possible to fix that.

Leo: Yeah. VeriSign should certainly be recognized as a certificate authority. They're probably the No. 1 certificate authority in the world, I would guess.

Steve: Yeah. Yeah, exactly.

Leo: Of course I went to the same site on my browser, on my computer in front of me, and it says the certificate's valid, and I get all the extra little doohickeys and doodahs. So, yeah, it sounds like it's something wrong with his particular system.

Steve: I think just one machine is having a problem.

Leo: There wouldn't be a way for his company to be causing this problem and the certificate still to look good, would there?

Steve: No. And that's the problem. If his company were doing a man in the middle, then it would be impersonating TD Ameritrade in his example. And the certificate would show as being something other than VeriSign. And then he would be getting exactly the same error, that it's signed by somebody non-trusted, if his computer also did not have his bank's own public certificate that was used to sign the man-in-the-middle certificate. So you could get that error, but then you would see that it was not signed by VeriSign. The fact that it was signed by a valid CA says that your computer is not recognizing the validity of VeriSign, and there's no reason for it not to except corruption in the certificate authority database.

Leo: Right.

Steve: That explains it.

Leo: Question 5, Kevin Gunn in Denver, Colorado wonders about Perfect Posthumous - Posthumous? - Passwords: Steve, how do handle or suggest handling giving passwords to loved ones after you die? Sorry for the direct, morbid question, but this is one I've been wrestling with lately. I think we all are, Kevin. I think we've talked about this before, and it's a great question. I'm a LastPass user. My password is a complex, 10-plus character password - numbers, letters, lower and upper, as well as special characters. All of my accounts have complex passwords I've created using LastPass or some other complex password generation tool. Occasionally those passwords change, including my LastPass.

The issue, then, is when I do, I realize my wife won't be able to log into any of my accounts. Not even I know my own passwords. I just know the LastPass one. That's how many of us do it. You trust LastPass, but you don't have to remember all those complex passwords. I've heard about some sites that require you to check in periodically and, if not, they presume you're dead and send out a message to whomever you define. But I'm concerned about security. I obviously don't want to hand out my LastPass password to some site that would give unwarranted access to all of my accounts. Others have suggested writing passwords on paper, storing it in a safe deposit box at the bank or something, but that last one would have to remember to update the list, or at least the password, on password changes. I'm just curious, as I begin a quest for a solution, if you have any suggestions. What do you do yourself? Thanks for your work.

Steve: So I think it's a great question. One of the things that any of us who are being

responsible need to think about is what happens if...

Leo: Right. If the worst happens.

Steve: ...something happens to us, yeah, if the worst happens, and people we would wish could get access to our accounts of all kinds are unable to. The beauty of LastPass is that it aggregates - or any similar password manager. It aggregates this mass of disparate logon credentials under a single authority. So there's one point of permission, of authentication, that then expands to all. So that's really good from that standpoint because one could imagine listeners to this podcast may have gone to extremes for security that would be hard, you'd be hard pressed to expect wives, siblings, whomever, to deal with.

Okay. So I had a couple thoughts. One would be you could generate a list of passwords. For example, say that you decided every six months you wanted to change your LastPass password. You generate, you pre-generate a list. And say that you've got, you imagine, 50 years left of your life. So every six months, that means you need a hundred of them. You could go to somewhere like GRC's Perfect Passwords, which will give you gibberish to your heart's content, and just copy them down, make them whatever length you want, and create a little spreadsheet of January 2012, July 2012, January 2013, July 2013, and so forth, and show the matching password. It's then up to you to keep your own list private yourself, and to follow that to change your master LastPass password every six months on that schedule, knowing that locked up safely, maybe it's in your attorney's file for you, or in a safety deposit box which will be unlocked upon your demise, or whatever. So that would be one way.

Another way that doesn't require any kind of, like, list pre-generation, I had the thought of, well, there are websites that offer web-based hashing, like MD5 or SHA-1. I just put into Google when I was posting this question, "web-based SHA-1," and up came some. So what you could do there is there would be instructions which you leave which say, if anything happens to me, put in this phrase, dash, and the year, and into this site. You hash that into - through an SHA-1 hash..

Leo: Clever.

Steve: And that's your key. So...

Leo: So all you have to put in the safe deposit box was this one piece of paper with a date next to each one saying this is - in this time range, this is the password; this time range, this is the password; this time.

Steve: You could do that.

Leo: Or - you could do that, or do what I like is use this - I use SuperGenPass, which does this. Your master password would be the year and day. And it would change according to those days. I think that's a good idea. Very clever. And then you wouldn't have to worry about it. You just remember to change your master

password on a regular basis.

Steve: Or one final solution is - this is a nice reason. So using all of that requires technology. We have a non-technology solution called Off The Grid.

Leo: Yes.

Steve: And that's exactly what this is for. So you could leave instructions to use an Off The Grid grid where you simply put in the year, 2013, you do the Off The Grid approach using the Latin square. And then from wherever you land, you just, for example, do 12 characters to the left, for example, or whatever your instructions are.

Leo: Ah, yes.

Steve: And that way you have a changing password, no technology. It goes forever. You don't have to do it in advance any number of years or months or whatever. And you can make it as complex or as simple as you choose. And as you say, Leo, then your obligation is to make sure you stay current with the instructions you've left behind. Oh, and by the way, if you didn't, that's not a problem, either, because you could say, and if it doesn't work...

Leo: Go back.

Steve: ...try the - exactly.

Leo: I forgot.

Steve: I forgot.

Leo: Right.

Steve: And so go back in time until it does. And you'll find the master one that works.

Leo: That's clever. Now, somebody in the chatroom does point out that there is a LastPass sharing feature. You can share your LastPass with somebody else. Of course the reason that he doesn't want to do that ahead of time is he doesn't want to hasten his demise by inadvertently giving his passwords to his wife before he's passed on.

Steve: Right.

Leo: Right. Just a little joke there. Very little. Samy, Pasadena, California wonders how secure are anonymizers. Steve and Leo, I'm fairly new to Security Now!, but you've got me hooked. Your explanation of the WPS protocol was amazing. Thank you for all the work you do. I'm curious what your thoughts are on anonymizers like TOR or Ipredator. Are they truly anonymous? Or is it possible to decrypt and trace the hops back to the true source? Secondly, in the case of TOR, do you think the risk of a peer node injecting malicious code into returned results makes TOR too dangerous for use? Thanks again. We have covered this. But I agree, it's probably worth talking again.

Steve: Well, yeah, I thought, quickly, just saying that I guess my take is that you're trading off one thing for another. So if you don't use a system like that, then your traffic sprays out from you all over the Internet, point to point. If you run through a VPN or TOR, and Ipredator is just a VPN offered by the Pirate Bay people, then your traffic is going through that system. And at some known point it is then emerging, unprotected and unencrypted - unless you've got other encryption in the connection, but not encryption provided by this service - onto the Internet.

Because those are high-value emission points, that is, the powers that be with initials in their names could figure, well, these TOR endpoints are worth watching, or the Pirate Bay's VPN server is worth keeping an eye on. So now your traffic is emitted from the Internet at a high-value location, much easier for organizations to watch, if they wanted to, than if it was just you at home, sort of lost among the millions. So there's a tradeoff, I think.

Leo: Yeah.

Steve: It does provide you with anonymity, but it also brings your traffic, I think, more to the notice, potentially, of organizations that would care.

Leo: Furthermore, you can't count on Pirate Bay or other organizations to not respond to government subpoenas.

Steve: Yes.

Leo: We've learned that with HushMail, for instance, which is an email encryption service that just handed over keys when asked. And that's, in most cases, what any of these services are going to do unless they want to be outside the law.

Keenin, Moses Lake, wherever that is [Washington], offers some fun math: I had a discussion with a classmate about encryption and how he thought the government could break anything. I knew better than to argue with him, so instead I did some math for the fun of it. I know you already know this, but I had to vent. I took 256-bit encryption and assumed that the only way to crack it was, as we currently believe, a brute-force attack against the 256-bit key. After all, we're talking 128-bit this and 256-bit that. It's the bit length we're focusing on.

So let's say the tricksy government has a secret algorithm that somehow allows them to weaken the strength to one trillionth of the original. That's a good number, one trillionth. And let's say they had a computer that can try 100 trillion guesses per second. And let's say this computer was one cubic millimeter in size, and let's say they build a cracking complex the size of the entire Earth made out of these one cubic millimeter crypto cracking computers. If I did my math right, it would still take 34 trillion years to crack. I like that.

Steve: I like that, too.

Leo: Did you check his math?

Steve: I did not. But it sounds right. And I liked - I wanted to include this just because we're so glib about saying, oh, we went from a 128-bit key to 256-bit. It is so easy to sort of get this sense of, well, that's twice as strong. Oh, honey. No, 129 bits is twice as strong. 130 bits is twice as strong as that. And so on for another 126 times. So, I mean, it is so ridiculously stronger. So anyway, I loved Keenin's math.

Leo: People have trouble with orders of magnitude, exponential, things like that. And so that just brings it home.

Steve: Yup. It's not intuitive.

Leo: It's not intuitive, exactly. Nile Davis in Mesa, Arizona has some questions and some thoughts about assembly language, SpinRite, and the Navy SEALs: I've been an avid listener of Security Now! since the beginning, listening on the ski slopes, the airplane, road trips, while mowing my lawn and so on. I can't get enough. Although I've been intending to reply for a while now, after listening to Episode 332 I wanted to ask you a favor. I have dabbled with assembly language programming since I taught myself how to use the 6502 chip in an Apple II a long time ago. You mentioned you're writing assembly language, but it's really pretty. Pretty is in the eye of the beholder, I just want to point that out.

Could you write a simple "Hello World" type program in assembler and let us see what it looks like? I'm sure it would help those of us who would like to program in assembly, but would like to see how pretty yours is. I've been a user of SpinRite since v3.1, I don't remember the exact version number, and it's helped me with multiple hard drives, floppy disks, iPods, TiVos, even an old Xbox. I'm looking forward to v6.1 since I have a couple of motherboards with BIOS issues. Got any idea on its release date? And finally, my wife's first cousin is a Navy SEAL. And after sharing with him the story about SpinRite saving the SEALs, he said, "Yeah, that could have happened." So, well, I believe the story. Thanks, and keep up the great work. Nile Davis, Mesa, Arizona, an orthodontist who's crazy about technology, but I usually work with "bites" of a different order.

Steve: Okay. So it has come up from time to time that people wonder what my assembly code looks like because I talk about how it's possible to write...

Leo: Pretty.

Steve: ...high-visibility assembly language. Assembly code which is ultra efficient, executing individual instructions of the raw chip, insanely fast, incredibly small, yet doesn't just look like gobbledy-gook. So I just posted in my Twitter feed links to two GIF images of my editor. This is what I see when I look at my assembly code. I did it as a GIF image because I use a colorizing system and a text box. So just giving you a text file, you'd lose the line drawing characters and a sense for what I actually see, which is what makes it pretty. So, Leo, if you go to Twitter.com/SGgrc, you'll see right there at the top of the feed a couple links. I did little bit.ly shortcuts of SG Assembly and SG Assembly 2, or you could just put those into bit.ly/SGassembly or SGassembly2. And you'll see...

Leo: Look at how nice the comments are with boxes and everything. Do you have a macro that does those boxes? Or you don't type those, yeah.

Steve: I do. I do. No, I've got to just press one button that creates the whole framework for me. But that's - and that's what it looks like. Even though it's structured, you've got if, then, else constructs, you've got looping constructs. Those all are, before I even considered using them, I looked at what Microsoft's Macro Assembler did with those, and they all resolve to a single instruction. So it's got zero overhead.

Leo: So when I see - so I'm looking at a "repeat until" loop that has an "if elsif end if" inside it. Those commands begin with a dot. They're not actually assembler commands, they're macro commands.

Steve: Correct.

Leo: So they expand out as predefined by Microsoft's MASM into assembler, but invoke...

Steve: Yeah, and I can't just - they invoke one instruction. So, for example, I'm not able to just put any random phrases in the "if" statement.

Leo: Right, right.

Steve: I'm, like, putting register descriptions or variable names. I mean, it's exactly - I know exactly what code is being created, and so I use that because it's so much easier to read. I mean, it doesn't look like - I see weenies who write assembler, and it's just like a byte of opcodes down the left-hand margin of the page. And I think, well, good luck with that.

Leo: So this whole block that we're looking at has, as far as I can tell, no actual

assembler in it. If we go to the next block, you'll see some move commands, which is assembly language. But invoke is not - is that an assembler command, or is that a macro?

Steve: Well, it's a macro which is used to invoke Win32, the API. So those are API calls.

Leo: I see. I see. So these are API calls - read, application, DWORD is an API.

Steve: Actually, that's my own. So I'm also able to define, so those are actually subroutine calls or API calls. So I have subroutines that I've defined that look just like that, that you give parameters to. You'll notice it's all spelled out. I'm not using cryptic language for anything. And the reason is, when I come back to it a couple years later...

Leo: You want to be able to read it, yeah.

Steve: I can read it.

Leo: Well, and so if anybody's programming in a high-level code, this really looks somewhat similar to high-level code. There's a few things. There's things like "check eax," you have to know that that's a register and things like that. But it does look like high-level code for the most part. It does not look like gobbledy-gook. You can - but you're still working in assembler. You're still doing a "mov eax, reply" length, which is moving a value into a register. That's pretty low-level.

Steve: Oh, it is, it's low-level.

Leo: That's down there.

Steve: The other thing is that the subroutine names, the variable names are long and meaningful. So it's not XR2Z or something, which has no meaning at all. It's much more useful variable names.

Leo: This really is beautiful code. Actually I've never seen your code before, so it is kind of fun for we...

Steve: That's what it all looks like.

Leo: ...we who are geeks to take a look.

Steve: And SpinRite 6.1 timing, I have no idea. It is, however, the next major thing I'm going to do. It will be a free upgrade for everyone who has 6.0, oh these many years.

And the target is to catch it up with things that have happened since, like the big 4K sectors that WD drives have. Even though SpinRite works on them now, there are things I can do to make it better. Deeper awareness of SATA drives. Better awareness of hybrid drives that use both flash and magnetic storage, and other stuff. So it's going to be not tons of new features, but performance improvements and sort of a catch-up. And then we'll see where we are. I would love to move on to 7.0 and add a bunch of new features, as well. And I'm glad to know that the Navy SEAL adventure could have happened. We haven't ever repeated that, Leo.

Leo: It's a great story.

Steve: We're going to have to repeat that one of these days.

Leo: Yeah, it's a really great story. Question 9 from Doug in Illinois, having trouble staying ahead of his family and friends' PC infections. Oh, Doug, you have my both sympathy and understanding.

I do more than what I would like at times in cleaning family and friends' PCs, or at least hearing about them. I've noticed what seems to be a rash of infections from fake antiviruses. My parents' PC just became the latest casualty. I've done what I can to educate my family on protecting their PC. I've made sure their "real" antivirus and Windows are both kept up to date. My mom uses Firefox, I don't what version, most of the time, and jumps to Chrome and IE when all else fails. I have a feeling the source of infection on my mom's PC, believe it or not, are Christian websites whose administrators don't know what they're doing or haven't updated the site or server in years. The gist of my question is, is there any way a user can know if a site is trying to install software on the PC?

Steve: I just wanted to say, if there were a solution to it, we'd all be using it.

Leo: I love the fact that, and we've been bitten by this, but that Google now will give you an alert if it sees - and that's one of the reasons I like Chrome. If it sees that - because as it's - see, Google has an advantage. They're spidering these sites all the time. So if they see known code on there, known bad code, malicious code, they'll add you to a database. And when you go there on Chrome, it'll say [alarm sound]. And for the most part that's very accurate. And Explorer does that now, too.

Steve: And of course the problem is that, if it's something where you do something, that is, it's user action, user permission, that's still the way this stuff gets in. And so we're back to training users.

Leo: Yeah, true.

Steve: And the other thing I would say is this is a case where you really want to run as a limited user. If you can lock down the system, like Mom's PC, she really doesn't probably need to be installing anything, ever. And so this is a place where you definitely do not want to run as an admin. And maybe you even want to go further and do something like

prevent Mom from even being able to give UAC permission. I mean, really lock the system down tighter so she just can't go clicking away at things, thinking that she's doing something. Because if it's things like fake AV, remember that one of my favorite things of all, I think it was our friend Brian Krebs who I quoted, he said, "Never install anything you didn't go looking for.

Leo: Right.

Steve: I love that advice. If something offers you software, just say no. Unless you initiated the search, don't install it, period.

Leo: Right. I, for a long time, when my mom was still using Windows - she uses a Mac now. But when she was using Windows, I put her on Windows 2000, a limited account, and I never had a problem for years. She could do anything she wanted. I explained to her how to do antivirus updates, how she had to log in as administrator. I gave her the tool to do that, but she was smart. She only did that when she wanted to update her antivirus. And any other time she was running as a limited user, and it was absolutely secure. She never had a problem.

Question 10 from Mike Norin in North Bend, Washington. He says: ScriptNo - is it needed? This is that Chrome extension that we talked about last week. But he says: Given Chrome's built-in sandboxing capability, do you really need protection from malware on Chrome? And given Chrome's performance at Pwn2Own, and with its regular automatic updates from Google, it seems Chrome is pretty well locked down out of the box. Do I have a false sense of security? Thanks.

Steve: The only thing I would note - first of all, I agree with all that. I hope that Chrome's sandboxing is good and getting better. But we'll also notice that Chrome is being updated all the time.

Leo: Yeah, there's always something.

Steve: And it's not only to increment their version number faster than Mozilla.

Leo: I found ScriptNo such a pain in the butt.

Steve: Yeah?

Leo: Because I kept going to sites, and it wouldn't work, and I'd think it was me. And then I'd remember, oh, I have ScriptNo installed. And I would have to give permission. I just gave up.

Steve: I know. It is a different mode of operating, one I'm willing to do.

Leo: You're used to it.

Steve: Yeah, I'm used to it. And again, Chrome is being updated, but they're being updated because of problems being found. So I love that it's secure by design rather than by hope. But they're still only human.

Leo: And finally, our Unintended Consequence of the Week Report from Bob in Littleton, Colorado: Steve and Leo, I found that the web and PDF transcripts of 339 were cut short at the point right after Steve says, "By default, he blocks the blank." I brought up the simple text transcript for 339. It appears to be complete. By this time you've probably been notified of this. I enjoy every podcast every week. Thanks, Bob. What happened, Steve?

Steve: Oh, this is where Elaine is so fantastic.

Leo: She wasn't fooled.

Steve: I said - and no other transcriber on the planet would have done this correctly. I said, "By default" - we're talking about the ScriptNo add-on. I said, "By default he blocks the [ahem] NOSCRIPT [ahem] tag." Elaine knows that HTML tags are in angle brackets: less than, NOSCRIPT, greater than. So in her transcript she has it correct. And when that loads into the browser...

Leo: It stops.

Steve: It stops because it says NOSCRIPT.

Leo: You have to tell her about HTML entity encoding from now on.

Steve: Isn't that fantastic?

Leo: That is pretty funny.

Steve: I got a report quickly after the transcripts went up, and I thought, what the heck?

Leo: What could have possibly happened?

Steve: Because I've had no trouble at all in all these years. And sure enough, I looked at it, and I just thought, oh, my goodness, the scripting was enabled for my local site because I'm using it on my own browser. So as soon as the HTML hit the NOSCRIPT tag, it dropped everything from there on. And the transcript just stopped. And so all I had to

do was I just turned those into > and <, and it was just fine then. And so it showed them in the transcript without it chopping it off. So, oh, I just loved it. What a wonderful little bug.

Leo: And if Steve ever says "drop table" in a podcast, please do not execute. That's pretty funny. That is really funny. Yeah, so use HTML entity encoding by doing ampersand lt (for less than) semicolon and ampersand gt (for greater than) semicolon. And then you'll be okay.

Steve: And Elaine, you're forgiven because it is that level of technical transcribing accuracy...

Leo: Oh, it's fantastic.

Steve: ...that makes it all worthwhile.

Leo: Not her fault.

Steve: So, yeah.

Leo: Steve, thanks so much. We do this show every Wednesday at 11:00 a.m. Pacific, 2:00 p.m. Eastern, 1800, I guess 1900 UTC because we're not in summertime anymore, on TWiT.tv. You can watch live, always fun. But if you don't get to, or it's a bad time for you, please download the podcast. We have audio and video available at TWiT.tv. You can even subscribe on iTunes or any podcast catcher, downcast on the iPod or iPhone, or the iPad, I should say. I use Listen from Google on the Android side, and just subscribe, and you'll get them automatically, and you can listen to them, which is wonderful. Steve does have a version that's a little unusual. He has a 16Kb audio version. That's the smallest audio we make available, for those of you who are bandwidth impaired or throttled or whatever. And there's also those great transcripts from Elaine, all available at GRC.com. That's where you should go if you want to give Steve any feedback. He's got a form, GRC.com/feedback. You'll also find SpinRite there, the world's finest hard drive utility.

Steve: Yay.

Leo: As well as a lot of freebies because Steve is really good on just making stuff available for free. And, finally, follow him on Twitter so that he can DM you. He's @SGgrc, if you want to follow him on Twitter. Steve, thank you so much. We'll see you next week.

Steve: Talk to you then, Leo.