Listener Feedback #225

Description: Leo and I discuss the week's major security events and discuss questions and comments from listeners of previous episodes. We 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-538.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-538-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. There's of course a ton of security news for him to talk about. And then we're going to try to answer 10 questions from you. And I'll give you a little preview. The first question is so great that Steve will spend the rest of the hour doing that. It's coming up next on Security Now!.

Leo Laporte: It's time for Security Now! with Steve Gibson, Episode 538, recorded Tuesday, December 15th, 2015: Your question, Steve's answer, #225.

It's time for Security Now!, the show - it's my lunch hour show. The show I think - and I bet you I'm not alone. A lot of people go, all right, I'm going to take a break for lunch, put in the headphones, and you listen to Steve, the master of security and privacy, really the master of technology and engineering, talk about a variety of subjects. To me, this is like...

Steve Gibson: I will help you stir your stomach contents.

Leo: We used to, at TechTV, we'd have brown bag lectures from time to time. And I know a lot of big tech companies do that, too, where you bring your brown bag lunch, and you learn something.

Steve: Yeah.

Leo: This is your brown bag lecture for the week. Hi, Steve.

Steve: And I guess I've seen that, like, setting in corporations where everyone has, like, the little transparent clamshell salad, and they open up their meal, and they munch on salad while someone's showing them things.

Leo: Exactly. Exactly.

Steve: So this is a Q&A. It's our 225th Q&A, Episode 538. And this one is special because, going through the mailbag last night, I hit a question that I think is quite possibly the coolest question that has ever been asked, or that I've ever been asked. I mean, and we have had, obviously, many questions. What, 225 episodes, about - for a while we were doing a baker's dozen, but then they started - we didn't have, you know, then there was so much news happening that we didn't have time to do 12, so we cut it down to 10. But so arguably 2200-plus questions.

This is just so good. For a while I thought, okay, I'm just going to cancel all the other nine questions. And but then I thought, okay, no, that's going a little too far. So, but I did bring it down to eight because it's just - and for our listeners, who are by-and-large techie and technical, I'm going to suggest after the pet question is posed that everyone pause the podcast and contemplate the answer themselves. See, you know, see what you think. Because, oh, it's just a great question.

But we have, before that, of course, we've got news. We've got updates on the government or various governments versus crypto. You mentioned on the previous podcast, on MacBreak Weekly, what we'll talk about, another chilling discovery, thanks to the Shodan search engine. We are two weeks away from the sunsetting of SHA-1, and people are finally really beginning to get a little upset because it turns out that 40 million people, estimated, will be cut off from having any security on midnight of December 31st.

Leo: When you say "security," you mean encrypted web traffic, like HTTPS, or...

Steve: Yeah, yeah.

Leo: Yeah.

Steve: Yeah, exactly. So because they are still using browsers, there are, it's estimated, 40 million browsers that do not support the next-generation of certificate signing, which uses the SHA-256 hash. They only understand SHA-1, that of course has been acquiring dents over time as cryptographers have successfully found increasingly unsettling things. Also we talked a couple months ago maybe about how Google was upset with Symantec over some mishandling, Google felt, of some certificates. And Google's announced it's going to yank support for a Symantec root, which is kind of surprising.

There is a bad horror story involving Bell Canada wireless routers that we need to essentially warn Canadian listeners about. And we're going to revisit the question of what do we know about Satoshi, since now we think we know less than we thought we did. And I know you've been following this story closely.

Leo: Oh, yeah.

Steve: So I want you to bring everybody up to speed because I haven't dug into it that much.

Leo: So fun, it's so fun, yeah.

Steve: We've got a master's thesis that was written that analyzed the Telegram messenger's, ahem, homegrown crypto. And if that doesn't tell us in advance what the outcome is, I can't think of anything that would. Then some miscellaneous stuff. And then, like I said, a Q&A with only eight questions because number one is just, oh, it's like, thank you for asking this question. What a - it's a fabulous question.

So the Picture of the Week actually involves this SHA-1/SHA-2. SHA-1 is just one hash. SHA-2 is the family of successive next-generation hashes, whose size differs. There is SHA-256, 384, and 512. So three different SHA-2 hashes, all, even the weakest of which, though, is 256 bits, way stronger. As we know, when bits are doubled, you don't double your strength, you two to the doubling number increase your strength.

Leo: It's 2^128 bit or something like that.

Steve: Yeah, exactly, it's just like, whoa.

Leo: It's a lot better.

Steve: So, yeah, way better. So but this Venn diagram shows, unfortunately, there are some systems, notably Android 2.x, old, yes, but still in use. Windows XP SP2, because it wasn't until SP3 that the security of Windows XP was updated to add SHA-2, that is, SHA-256 support. And one of the main streams of OpenSSL, remember there are, like, there's a v1 point whatever, but there's always been 0.9.8, for a long time that doesn't, that also doesn't have SHA-256 support. So, and then there's a group in the middle that offer both - Safari, Firefox until or up through v36, Chrome up through Chrome's v38, and Opera from 9 on, support both. But then there are some, the really more recent ones. Edge has no support for SHA-1, does support SHA-2; Firefox from 37 on only supports SHA-2; and Chrome from 39 on.

So anyway, so this we'll come back to when we talk about essentially what this means in terms of the way the industry is evolving. And CloudFlare has, as you mentioned, a great blog posting. There's a link in the show notes. But significantly, Facebook announced that they're going to be doing this, too. So we'll talk about that. But first I wanted to talk a little bit, just sort of generically about, you know, the interesting times we're in relative to states and encryption and just the tension that exists there.

I picked up in a website, or I guess one of my listeners actually did send me the link, because we talked last week about Kazakhstan's apparent intent, although the fact that the press release announcing it was pulled without explanation, I haven't seen anything about that since, I don't know if there was a huge backlash or what. But remember that

the idea was that we were going to from the local style man-in-the-middle SSL or TLS interception, which some antiviruses do on users' machines, or corporations do for their networks, up to the next, well, I guess two steps. Because the next step would be ISPs, and that's frightening, in case that ever hits us.

But the one beyond that would be state level. And that's what of course Kazakhstan, we discussed last week, has formally, there was a press release that went out that said anybody who wants to be able to communicate with security, that is, you know, web-style, certificate-based, public-key crypto security, which is to say establish a secure connection to any website outside of Kazakhstan, where the traffic needs to transit Kazakhstan's border, will only be able to do so if they load their device, whatever it is - PC, mobile phone, tablet, whatever - that wants to communicate, with an official Kazakhstan certificate.

And so this is at the state level formalizing this, exactly the same sort of secure connection interception capability. What that means is that, when you attempt to connect to a service, a server outside of Kazakhstan, assuming this happens, Kazakhstan will themselves synthesize a certificate and sign it for that site. And that's what your browser will connect to. Well, because your browser has previously accepted the Kazakhstan national security certificate, it won't care. No warnings, no alerts, nothing. It'll trust it.

And what this means, though, is that Kazakhstan's border will then decrypt all of the traffic, you know, your username and password, which if there isn't additional encryption - and that's something worth noting because now that we've got JavaScript in browsers to the degree we do, nothing prevents some, you know, potentially some other layer of encryption being employed. But absent any additional encryption, your username and password will go in, essentially has a moment where it's in the clear before Kazakhstan's border then connects to the remote server, hopefully also securely, reencrypts your data, which at the border they could briefly examine, and then on it goes outside of the country to the world.

So anyway, this website Defense One's headline that caught my attention read: "Kazakhstan's New Encryption Law Could Be a Preview of U.S. Policy." And it's like, whoa, [choking], what?

Leo: I don't think so.

Steve: No. Now, the good news is the article, they must have gotten paid by the column inch. So I read through it looking for anything, and there was nothing there. It was just like, well, okay, and maybe not. So but anyway, I found it interesting that this is sort of what's in the wind. Something else that's in the wind was reported by Motherboard.vice.com, reporting on a new position that our good friend FBI Director James Comey has taken. Of course, you know, this relates to the existing tension and sort of the unsettled question of encryption in the U.S., which U.S. law enforcement has a need, they believe, in when they can demonstrate it in order to see into connections.

So Motherboard reports with the headline "FBI Chief Asks Tech Companies to Stop Offering End-to-End Encryption." So the short version is, well, okay. You're saying that there's no way to do a backdoor. You can't give us a master key. There's no way. So just don't do it at all.

So Motherboard writes: "After the recent attacks in Paris and San Bernardino, encryption has once again become a political target in Washington. Despite there still being no solid

evidence the attackers benefited from or even used encryption." And in fact we heard right after the Paris bombings that a phone was found.

Anyway, Motherboard says: "In at least one case, they coordinated via distinctly unencrypted text messages. [Nevertheless] law enforcement and national security hawks have used the tragedies to continue pressing tech companies to give the U.S. government access to encrypted communications, even if that means rolling back security and changing the nature of their businesses."

So, and this is based on a Senate Judiciary Committee hearing which occurred last Wednesday, so the day after our last podcast, wherein "FBI Director James Comey went so far as to suggest that companies providing users with end-to-end encryption might simply need to, well, stop doing that." And then, quoting him, he said, quote, and this is Comey: "It's not a technical issue." Okay. We won't take issue with that. "It's not a technical issue," he says, "it's a business model question." And he said: "Lots of good people have designed their systems and their devices so that judges' orders cannot be complied with, for reasons that I understand. I'm not questioning their motivations. The question we have to ask is: should they change their business model?"

So, again, this is not actionable. I'm not suggesting it is. But this is, I just want to keep our podcast listeners up on the machinations and the ruminations. I did also pick up some thread about the present Obama administration's statement that they would, by the end of the year, make some sort of declaration, like readdress this in some formal way. So maybe within a couple weeks we will have at least something a little more firm than this testimony in front of the judiciary committee.

In another interesting twist, Russia has been for some time unhappy with the idea that non-Russian Internet service providers, I mean, like Google and Twitter and Facebook and, you know, like the rest of the world, are storing Russian citizen data outside of their borders. So they put into place some legislation around the middle of 2014 that sort of never got pushed and wasn't acted on.

But they've been saying now, more recently, that by January, I don't remember if it was the beginning or the end, I hope it's the end because apparently people still need some time, but the idea being that they're going to enforce, and I guess they can do so technically, so they're going to enforce a ban on access to Internet properties outside of Russia that store Russian citizens' data externally. Now, they have deployed the same sort of powerful NSA-style encryption technology throughout the country. They have their own central monitoring facility and devices installed in all of the various Russian ISPs in order to give them taps, essentially, to give them access that they require.

So in September, a couple months ago, Apple rented space in Russia to house the data of Russian citizens. And then some other companies, a messaging app Viber, also eBay, PayPal, and Booking.com, have decided to comply, meaning that they will create servers inside Russia where Russian citizen data will reside. But the big three - Twitter, Google, and Facebook - have said nothing. They've remained silent. Although there is a belief that they've got representatives that have been in private talks about what to do about this.

Leo: Yeah. Just because they don't do a press release doesn't mean they're not, you know, talking with the Russian authorities.

Steve: You're right, I mean, they have to be saying, look, this is what it's going to take,

or, yeah.

Leo: Negotiate, yeah. Do you want Facebook in Russia? Well...

Steve: Yeah.

Leo: The real issue is you can store the data - well, you know, I'm not saying - I'm not telling you anything. But the problem is they need to replicate data, and they're going to - it gets replicated globally. There's no way you can say only stored in Russia. Because then I, if I have a friend in Russia, I wouldn't be able to see the data.

Steve: Well, and I'm glad you interrupted me because, I mean, like my flow, because I meant to note that there's a fundamental problem with this.

Leo: Yeah.

Steve: Which is it isn't the way the Internet works.

Leo: Yeah.

Steve: I mean, it's what, you know, it's what Putin presumably, or his...

Leo: But isn't it where you have box in Mountain View, with everything, the whole Facebook is on box in Mountain View. So move that box to Russia, and I'll be happy. Not how it works.

Steve: Yeah, and so, yeah, that's, for me and our listeners, that's the key issue is it's like they're trying to mandate behavior of some specific major providers, but there's lots of other providers. I mean, you know, we sell copies of SpinRite to Russians. I'm sure I've got Russian data on my servers from, you know, their licensed SpinRite owners. So, what?

Leo: That's another issue. You're right. You're right. That's a very good point.

Steve: Yeah, I mean, the problem is this may be what they want, but it just - it isn't, I mean, it's fundamentally, at the deepest molecular level, not the way the Internet is.

Leo: Data doesn't recognize national borders.

Steve: Right. Which is like, good. I mean, it's like...

Leo: That's why we like the Internet.

Steve: It's why it works. It's why it's, you know, it's why AOL, and we're not still dialing into AOL.

Leo: Right.

Steve: This is the way it should be. And they're saying, uh, no. So anyway, I thought that this was interesting. But thank you for giving me a chance to take a breath because this was a point I had intended to make when I was putting this together was, you know, you can ask. I mean, and these guys could even do something to placate. But basically it's just not the way everything else works.

Leo: Yeah.

Steve: And I had here in my notes the story about MacKeeper, this sort of much maligned, and apparently deservedly so, really not a great reputation, Macintosh maintenance, I don't know what, D&C (dusting and cleaning) tool of some sort, I mean, it just...

Leo: It's not, you know, it's like the equivalent of a registry cleaner. You know, it deletes temp files. It's just...

Steve: Anyway, so the story, the part of this that our listeners will get a kick out of, well, sort of, is that a bored IT help desk guy, Chris Vickery, one evening sort of thought, well...

Leo: Oh, I didn't know that. I thought he was a security researcher.

Steve: Oh, no.

Leo: He's just some guy.

Steve: Yeah, he's an IT help desk guy. You know, he answers the phone and tells people, okay, plug it into the green port. So he was just - he was poking around using Shodan. We've talked about Shodan. It's basically, in the way that Google spiders port 80 and 443, which is the HTTP and HTTPS ports, Shodan spiders everything else. So you could think about a web search engine is going to - it just, you know, it follows links and indexes things, the contents of servers that connect, that answer connections to port 80 and 443. Shodan says, eh, there's 65,533 more ports. What's there? Maybe there's some other new stuff there. And so it indexes those.

So Chris, bored one evening, asks Shodan, what do you have on port 27101? And it

returns a bunch of IPs. It's like, yeah, there's things answering 27017. I'm sorry, I got it - now I'm not sure because I have it 101 in one place and 17 in the other. But one of those two ports. I think it's 017, 27017. Turns out the very popular Mongo database, when it sets up, it opens a listening connection, a socket, on that port and will accept requests, queries, database queries, on that port. Of course, you never want to have that exposed. That's in your Intranet behind, hopefully, three or four layers of firewalls and NATs and all kinds of stuff so that there's no way that Shodan, wandering around the Internet, is going to get an answer from your database server. But in this case he found four IPs. Shodan responded to Chris's query.

And by the way, this has been fixed since. Chris acted very responsibly, found four IPs belonging to Kromtech, which is the publisher of this MacKeeper utility. It turns out - and so he was like, oh, interesting, logs into their database server. He's not even a Mac user, but this just turns up. So he figures out, he does a little googling and figures out how to log into MongoDB, does so, and finds 21GB of remotely accessible Kromtech MacKeeper customer data. I mean, just everything that they've got on these people. Now, the good news is he informs them. They quickly closed the public exposure, and they claim that looking at the login records of their database, nobody else ever did that.

Leo: Mm-hmm. And then they called the guy and asked him, and he said, no, I didn't do anything wrong. So I feel better.

Steve: And you and I were talking about this before. This is one of the other problems we are sort of struggling with, sort of as an industry, and conceptually, and of course with pressure unfortunately from the entertainment industry that's got its own agenda in the form of the DMCA, the Digital Millennium Copyright Act, is it can be illegal to look at somebody else's copyrighted cryptography. But the problem, I mean, all the lessons we learn here on this podcast is that you have to have other eyeballs on this stuff. It's Google's people and their Project Zero that are examining, not only their own code, but other code, and finding things we're glad they found. I mean, we want this to happen.

Anyway, so there's a danger that Chris faced because these guys, if they were really evil, could say, oh, well, you know, you connected to our servers. You knew that was wrong. We're going to go after you legally. Good news is that didn't happen. They thanked him. He said, you know, you're welcome, and then presumably he's going to be poking around at Shodan some more. We may be hearing more from Chris in the future. But, boy, the idea of these database servers like that being publicly exposed is horrifying. And then the idea that you could have a global search engine that you can query, says, uh, what do you - give me some IPs for things that answer this port. And then it says, oh, here's a list. And then you sort of go, okay, and go through them.

Leo: It's super valuable though; right? I mean...

Steve: Yeah. Yeah, super valuable for, again, for security research. Also, unfortunately, when you find out that, like, all the light bulbs people have installed have a bad server that allows you to break into everyone's LAN that their light bulb is sitting on, uh, it's not - that's a problem. So a mixed blessing. But if, you know, if Shodan didn't exist, somebody else would just do it themselves privately because it's not a hard thing to do.

Okay. So SHA-1. We've talked about this often. It was in October, a couple months back, that a full round collision was computed using a wall of graphics processing units,

PlayStations or something. And this is not a collision of the whole hash. But what we've seen is we've seen reduced-round collisions, that is, where you, for example, SHA-1 does - it does a stirring of the pot, essentially, 80 times. Well, if you only stir it 10 times, it turns out the bits haven't become sufficiently scrambled to prevent them from being unscrambled. So it's the unscrambling the egg problem. It hasn't really been scrambled enough. It's like, okay, we can make this look like an egg again. Eighty rounds of the full SHA-1 has never been - a collision has never been created.

The reason that's important is that, if you could deliberately synthesize an SHA-1 outcome - or put it a different way. If you could create something different that produced the same result, say that somebody wanted to create a fraudulent certificate, the certificate is signed with a, for example, an SHA-1 signature, for which only someone you trust, like we'll just pick on Google for the moment. They've got the private key. This is the Google certificate. They signed it. And you can verify the signature. So that's why you trust what the certificate asserts.

But if it's possible to create a different certificate that has the same signature, then that different certificate, with the same signature, essentially reuses the signature that Google created with their private key. And so you would trust that, too, even though Google never saw it. Google never signed it. But it's got a valid signature. So that's why the hash collision is a problem. You want to make it - and that's the promise of a hash. The guarantee is you can't do that. There is no way, I mean, if the bits are so scrambled, essentially, there's no way for it to be computationally feasible for two different certificates to deliberately collide, to have the same hash.

But SHA-1 is getting older. Computers are getting faster. PlayStations are always getting faster. So there's a lot of computing power available that we didn't used to have. And we talked about Bruce Schneier's famous guess about, like, what year it would be. And he was right about the shape of the curve. But things have moved a little faster than he predicted. So I think it was 2016 he was - no, no, it was 2020 maybe. And it ends up more like, oh, more like maybe 2016. So Bruce was just, you know - and this was a prediction made, to his credit, a long time ago. So when you consider when he said this, he was shooting pretty accurately in terms of when we would have enough computing power that we would have to seriously look at going to a stronger hash. Which is what SHA-256 is.

So we are two weeks away, and a couple days, from the end of 2015. If anyone looks at GRC's certificate, you will see that all of my certificates, my EV certificates, expire on midnight of New Year's Eve, and they are today signed with SHA-1. And that's on purpose. Many people say, Steve, you know, you're still using old certificates. It's like, yeah, because, first of all, there's nothing that is super important that we're keeping secret in the first place.

But you can only reach GRC over a secure connection. And it is still the case that a significant percentage, not huge, but significant, on the order of about 40 million people, would not be able to connect to GRC for the last six months or so, or at any point, if I had dropped SHA-1 and switched to SHA-2. I have them standing by. DigiCert's been great. They made, at my request, because of GRC's particular needs, SHA-1 certs that would expire on that date because, if they expired in 2016, then Chrome would be warning people that my site is using a certificate still valid in 2016.

So it's like, okay. That forced me to kill them on midnight. And my intention is to switch. I will finally switch over. And I'll actually do it at the beginning of that last week, like the beginning of next week, or the week after - yeah, it's the week after - because we know that if people's clocks are wrong, and people's computer clocks often, they're sometimes

off by a few hours or a couple days. So the expiration of the certificate is judged by the client. So anyone whose clock is off is, like, running fast five days, if I waited till the very, you know, to New Year's Eve, for that period of time they would think the certificate was expired, when in fact it was their clock that was telling them it was already 2016. So I'll swap my certificates.

Okay. So this is the problem we have, is a substantial percentage of the Internet still cannot connect with SHA-256. And that was the Venn diagram we showed at the top of the podcast. Android 2.x, anybody using XP at SP2, and there was a third one that I'm blanking on that was - oh, oh, the OpenSSL 0.9.8 version branch. Okay. So an interesting compromise has been proposed. And I first picked up on this thanks to somebody who tweeted me a blog from last Wednesday by Alex Stamos, who is a security guy at Facebook. And it's not very long and provides some nice background, so I wanted to share it.

He says: "Like many engineering fields, the practice of information security in the real world is all about finding an appropriate balance between two desirable goals. One of the most interesting areas of balance is between making systems secure against new attacks and providing security to the broadest population. This dynamic is readily apparent in the debate around the upcoming sunset of the SHA-1 hash algorithm, and my colleagues and I at Facebook believe that the current path forward should be reexamined. Our friends at CloudFlare have written an excellent post on the subject of SHA-1 certificates, and I would suggest you read their post for a good background on the issue." And for anyone interested, I have one little brief paragraph that I snipped out of it, but of course the link is in the show notes.

"Facebook's data shows" - that is, their own Facebook, Alex's Facebook data shows - "that 3-7% of browsers" - that is, browsers connecting to Facebook servers - "currently in use" - is in use today - "are not able to use the newer SHA-256 standard, meaning that tens of millions of people" - and it's estimated about 40, so four tens of millions of people, he says - "will not be able to securely use the Internet after December 31st. A disproportionate number of those people reside in developing countries, and the likely outcome in those counties will be a serious backslide in the deployment of HTTPS by governments, companies and NGOs that wish to reach their target populations.

"After discussing the issue with my colleagues at Facebook, we came together on the following two points: One, the recent advancements in generating SHA-1 collisions do indicate that the industry should transition to SHA-256 certificates. Two, we support the removal of SHA-1 support from the latest browser releases. Facebook has found success running" - and here it's really interesting.

"Facebook has found success running a large TLS termination edge with certificate switching, where we intelligently choose which certificate a person sees based upon our guess as to the capabilities of their browser. This allows us to provide HTTPS to older browsers using SHA-1" - and by that he means which can only use SHA-1 - "while giving newer browsers the security benefits of SHA-256. We don't think it's right to cut tens of millions of people off from the benefits of the encrypted Internet, particularly because of the continued usage of devices that are known to be incompatible with SHA-256" - like all of these old Android devices that are not going to get upgraded, and they're still in use.

"Many of these older devices," he goes on, "are being used in developing countries by people who are new to the Internet, as we learned recently when we rolled out TLS encryption to people using our Free Basics Platform. We should be investing in privacy and security solutions for these people, not making it harder for them to use the Internet

safely.

"Taking these ideas into account, I support CloudFlare's proposal for a different approach. Namely, the CA/Browser Forum should create a new type of" - he calls it an LV, a Legacy Verified. Remember we've talked about DV, Domain Validation; OV, Organizational Validation; EV, Extended Validation. So here's LV, a "Legacy Verified certificate that should only be issued to organizations that have demonstrated they are offering SHA-256 certificates to modern browsers." So sort of a, you know, a compromise intent certificate. "Such verification," he continues - and I just hit space, and I've just paged down by mistake.

Leo: You know, there should be something like space for page up.

Steve: Yeah. "Such certification can be automated or manual, and appropriate measures can be put in place to reduce the risk of a collision attack. Those protections could include requiring LV" - that is, the Legacy Verified - "applicants to have already passed OV or EV verification, as well as technical best practices such as serial number randomization. If this change cannot be implemented by December 31st" - okay, this is only two weeks from now - "then we call on the CAB Forum to delay the implementation of the SHA-1 rules for the period necessary to establish standards for legacy certificates.

"Facebook has already open-sourced the code we use for certificate switching as part of our Proxygen HTTP library, and all are welcome to use it under the terms of our BSD-style license. This is not an easy issue," he finishes, "and there are well-meaning people with good intentions who will disagree. We hope that we can find a way forward that promotes the strongest encryption technologies without leaving behind those who are unable to afford the latest and greatest devices."

And I have a link to Facebook's GitHub code, which jumps to line 381, showing a little six lines which - and I had already guessed how it had to work based on reading this, and our astute listeners who follow this stuff may be able to also. And that was validated by Facebook's code. Remember that when a browser or operating system operating on behalf of the browser, because some of the, like in Windows, the browser doesn't have the crypto library, and that's the same thing for Chrome on Windows. They both use Windows' native crypto library, whereas Firefox brings its own. Either way, the client of the client-server relationship, in its so-called "client hello" packet, it lists all of the cipher suites it supports.

Well, among the parameters of the cipher suite are the signature algorithms it supports. So this simple and clever hack has the server look and modifies the server's actions upon receiving the client hello packet to scan the list of supported signature algorithms to see if the client is declaring that it supports SHA-256. And, if so, that's the certificate the server chooses to send, to use in its ongoing TLS negotiation. It sends a server hello and so forth back and forth.

If among the certificate suites, I'm sorry, the cryptographic suites that the client says it knows, if there isn't any 256 - and so, for example, for a Windows XP system, no matter running IE or Chrome, that is, Service Pack 2 because they added it in Service Pack 3, but Windows XP SP2 or an older Android - all of their connections will list the cipher suites they understand. There will be no SHA-256 enumerated among them. So then one of these, sort of these SHA-1 fallback servers would say, oh, this client, I'm not able to give it the strongest certificate available, so we'll go with SHA-1 because the alternative is the client has just said, "I can't do SHA-256. These are the ones I can do. Help me out

here." And so this allows the server to do that.

I think it's very clever. The thing that immediately comes to a security person's mind is the potential for a downgrade attack. That is, and we've seen downgrade attacks in many different forms. It's not obvious how you would do that, that is to say, there are other measures that prevent that. For example, remember that one of the things that happens in the finishing exchange is they each send the other essentially the signatures of their entire previous conversation. So if anybody tried to go in there and to remove SHA-256 declarations from the client's hello packet, thus convincing the server that it needed to downgrade to SHA-256 signed certificate, well, then the client would see that what the server saw was different than what it sent and shut down the whole connection. So it looks like it's safe for downgrade attacks.

And I think this is really clever. And I just snipped out one thing from a very long blog posting by CloudFlare. The link is in the show notes for anyone who's interested. They just said: "The seemingly good news is that, globally, SHA-2" - and this is, again, this is the CloudFlare blog - "SHA-2 is supported by at least 98.31% of browsers. Cutting 1.69% off the encrypted Internet may not seem like a lot, but it represents over 37 million people. That's the equivalent of the population of California not having access to encryption unless they upgrade their devices. As SHA-2-only sites proliferate" - and, for example, GRC's going to have to become SHA-2 only because Google has said otherwise we're going to freak people out about your connection, tell them that it's not secure.

So, yeah, I'm switching. I'm waiting for the last minute, but I'm switching. So they said: "As SHA-2-only sites proliferate, if these users on SHA-1-only browsers try and access an encrypted site, they'll see an error page that completely blocks their access." And then they note that China, for example, 6% of users today in China cannot do SHA-256. So the concern has been that, in those areas where there are repressive regimes and tougher economics, so that they just - it's not feasible for them to upgrade their cryptography as the rest of the world already has and will continue to do so, maybe we need to say, hey, you know, here's a solution. Because SHA-1, as I have said, isn't actually broken yet. No one has created a collision that we know of and academically reported it. It just worries us.

So to me, this is brilliant. This is a tremendous compromise. And believe me, I'm not wishing - I'm wishing as much as I could that I wasn't using IIS and Microsoft server platform because I don't have access to this technology until or unless Microsoft decides to add it. And this doesn't seem like the sort of thing that they're going to do. I wish, because there will be, I'm sure, if this happens, more companies who are saying, hey, you know.

Remember we talked about this, oh, boy, I don't remember how long ago it was. It was when Mozilla themselves changed their server to SHA-256, and they lost millions of downloads of Firefox. And what was sad is that, since Firefox does bring its own crypto suite with it, if those millions of users, those millions of downloads that couldn't happen, if they had happened, then the users would be updating to a Firefox that does know 256-bit signing and then be able to use the rest of the Internet securely. But they couldn't get to Mozilla because Mozilla changed their server. So it's not just a theoretical, hypothetical, oh, you know, a tiny percentage. Unfortunately, the Internet is now so important that even a tiny percentage is 40 million people.

Leo: Yeah. Well, what is it, 8% of a billion users in China, or was it 6%, is...

Steve: Yeah, 6%, but still, yes, that's a lot of people in China.

Leo: Six million people; right?

Steve: And their, you know, Chinese citizens would like to have encryption.

Leo: Right.

Steve: You know, in three weeks.

Leo: Yeah, I'm glad CloudFlare wrote this article, yeah.

Steve: Yeah. And it is, as I said, it is a really cool hack, the idea of looking at the client's declaration and dynamically choosing a certificate. I mean, TLS already does that. For example, in SNI, we've discussed this, Server Name Identification. The certificate says this is the domain that I want to connect to. Then the server dynamically chooses the certificate for that domain when you've got - this is not wildcard certificates, remember, I corrected myself on that, but completely different domains, so there is this on-the-fly capability already. Well, let's extend it to cipher suites. It doesn't appear to be a downgrade problem because of, like I said, we will detect any change in the certificate on the fly. And this buys everybody some time.

Oh, and by the way, CloudFlare has deployed it on their entire network. All of their servers, all of the sites that CloudFlare hosts, will not go dark on New Year's Eve for all of these people. And now we know that Facebook won't, either. There is an option in the CloudFlare control panel. If for some reason you absolutely don't want that behavior, you can turn it off on a site-by-site basis. But otherwise it will be a seamless transition. You won't suddenly see your traffic drop on New Year's Eve.

Leo: I think our audience is sophisticated enough that we're not going to have to worry about that.

Steve: Right.

Leo: I hope.

Steve: So Google drops the other shoe. Ryan Sleevi, a software engineer who I follow and keep an eye on, he and Adam of course are very much in the security side of Google. We discussed several months ago, maybe it was a month and a half or so, that Symantec had been found misbehaving with some of their certificate issuing policies. And this is so interesting that, again, I don't want to risk misquoting Ryan. But you come away from reading what he posted thinking, I wonder what is really going on? Because this just seems like, okay, there's more to this story. There's something else happening.

So Ryan posted: "Over the course of the coming weeks, Google will be moving to distrust

the Class 3 Public Primary CA root certificate operated by Symantec Corporation, across Chrome, Android, and Google products. We are taking this action in response to a notification by Symantec Corporation that, as of December 1st, 2015" - so that's two weeks ago - "Symantec has decided that this root will no longer comply with the CA/Browser Forum's Baseline Requirements."

So then Ryan says: "As these requirements reflect industry best practices and are the foundation for publicly trusted certificates, the failure to comply with these represents an unacceptable risk to users of Google products." And as I said, this is why it's like, okay, what's really happening here? "Symantec has informed us they intend to use this root certificate for purposes other than publicly-trusted certificates." Okay.

"However, as this root certificate will no longer adhere to the CA/Browser Forum's Baseline Requirements, Google is no longer able to ensure that the root certificate, or certificates issued from this root, will not be used to intercept, disrupt, or impersonate the secure communication of Google products or users. As Symantec is unwilling to specify the new purposes for these certificates, and as they are aware of the risk to Google's users, they've requested that Google take preventative action by removing and distrusting this root certificate. This step is necessary because this root certificate is widely trusted on platforms such as Android, Windows, and versions of OS X prior to OS X 10.11, and thus certificates Symantec issues under this root certificate would otherwise be treated as trustworthy."

Now, the only way I can read between these lines is Symantec knows they lost control. That is, they're saying they no longer have confidence that they didn't lose the private key for this certificate. That must be what has happened.

Leo: Geez.

Steve: Pure conjecture. But that's, you know, they're saying we cannot comply with these baseline requirements. Well, one of the requirements is you absolutely know beyond a reasonable doubt that nobody has - you've never lost control of the private key. They must have. They must no longer be able to assert that that hasn't happened. And so this is the deeply politicalese of how that statement is made, and then Google saying, okay, we're yanking trust from that certificate. Message received.

Leo: Geez.

Steve: Wink, wink. Yeah. So really interesting. Okay. To all Bell Canada listeners with HomeHub brand, apparently 1000 and 2000 series routers: It has been discovered that, even if you're a faithful podcast listener, and you, as a consequence of that immediately disabled the frightening WPS support in your router's WiFi access point radio, even if you did that, it turns out what's been found is that these HomeHubs will indeed stop broadcasting in their beacon that they are supporting WPS. That goes away. The assertion disappears. Yet authentication doesn't.

And anybody with a little bit of packet-smithery capability can, to anyone of these, request over the radio WPS authentication, using the PIN 12345670. That will succeed. The router promptly responds with the WPA2, the good security, the WPA2 passphrase. And the attacker can then use that passphrase to connect. It takes less than a second, no brute-forcing is required, and it's a huge flaw in these Bell Canada HomeHub series

routers. The news surfaced from a posting on DSL Reports. There was a bunch of back-and-forth. It was independently verified by several people. The poster originally said 12345678. But the old-timers among us will remember when we beat WPS into submission years ago. There was a very clever hack where this eight-digit PIN could be, due to the bad protocol - this was always a bad protocol. The way WPS authentication worked, you didn't need to submit the whole eight-digit PIN at once. You were able to sort of independently check the first four digits.

And you'll remember when we talked about this, Leo. You could independently check the first four digits and determine whether they were correct, and then the next four, that is, the second four. Except that there's also a checksum, and the checksum is just, you know, it's a sum of nines checksum. Which means that the last digit must always be the sum of nines of the previous seven. So it's actually only a seven-digit PIN, which you can crack into a four and a three. Which means 10,000 possibilities on the first four, and then only 1,000 possibilities on the second three, and then you're in. Which is why everybody should turn off WPS authentication. On this family of Bell Canada HomeHub routers, you can't turn it off. You can turn it off, and it says, okay, it's not being offered. But turns out it still is.

Leo: They must have bought them from Linksys.

Steve: Anyway, so the poster originally said the PIN was 12345678, but that fails the checksum. It turns out it's 12345670, which passes the sum of nines checksum, and the router gives you the WPA2 password, which you then use to log in. So anybody can get onto anyone's network who has these HomeHubs. And let's hope that this comes to the proper attention and gets fixed soon.

Now, Leo, your turn. Satoshi Nakamoto. Of course I guess the news had just broken last week. Wired said that they thought they knew who it was, and this was a guy who was participating in a Bitcoin investors conference, and he smirked or giggled or averted his eyes or did something, I mean, he was just sort of acting a little squirrelly. And of course we didn't talk about it, but you probably heard about the Australian government breaking into his home?

Leo: Yeah, like the same day that he was outed, and then said, but it doesn't have anything to do with bitcoin. But I find that hard to believe. But, yeah. So this, I mean, it had just broken, and we talked about it.

Steve: In fact, yeah, we were doing the podcast, and you saw the news, I think.

Leo: Yeah, I saw the story as it broke. And it turned out we saw the Wired piece, but Gizmodo had a very similar piece. And in both cases the information they were working on had been provided by, well, Gizmodo called him a "hacker." Wired called him a "source." But a shady source, at that. And in both cases it appears that it could be that the information was provided by the guy himself. Apparently he's been going around kind of trying to convince people that he's Nakamoto.

Steve: Hey, nobody else is, so maybe he is, you know, or maybe I am. How do you know I'm not?

Leo: It seems that some of the documents might have been forged. The PGP key that seemed like a real smoking gun was backdated and possibly forged.

Steve: Right, right.

Leo: And the blog posts maybe weren't really from that date, but were post-dated. And so I think that there's - it's unclear, but I think increasingly just doesn't smell right. It's yet another one.

Steve: Well, I would say the guy got his just comeuppance by having the authorities break down his door.

Leo: Which explains why Satoshi Nakamoto, whoever he or they are, isn't stepping forward with any speed.

Steve: Not me. It is not me.

Leo: It's not me, either.

Steve: I did not - even though I explained how Bitcoin works a long time ago, I am not Satoshi Nakamoto.

Leo: The thing that's interesting is that there's believed to be a block of 1.1 million bitcoins that must be owned by Nakamoto, and they haven't been touched or accessed in any way in years. And that's what's really interesting because that supports...

Steve: Satoshi, if you're out there somewhere, and your hard drive crashed that had that bitcoin wallet, talk to me.

Leo: Call Steve.

Steve: Because we could do something. I could...

Leo: Because that's worth more than 400 million in today's bitcoins. It was more than that, even.

Steve: Well, and of course it was the tax authorities are the ones who said, uh, you know, are you sitting on a gold-studded couch over there?

Leo: But this is also why I and others have kind of always thought of crypto currencies as kind of, in a way, a pyramid scheme because whoever creates the currency can cash in, you know, when it's easy to generate bitcoin. And it's never easier than the first person to do it can kind of cash in on this thing.

Steve: We're just beta testing it, Leo. We're just - this is the beta test.

Leo: And if it catches on, which most don't, if it catches on it could be worth a lot of money. And Bitcoin did. So it is worth a lot.

Steve: Yeah, there were, like, 10 of them for a while, weren't there. And then they just sort of died off, or just never got...

Leo: Oh, there's Dogecoin. There's lots of blockchain-based coinages. And there's others, crypto currency and nongovernmental currency, all kinds. Canadian Tire has some nongovernmental currency you can use.

Steve: I wonder what kind of router they have up there.

Leo: Might be easy to access.

Steve: So, okay, this story, I just - this was on Fox News. And I just - I got a kick out of it. I found this in the mailbag, and it didn't really fit down in the Q&A. But a listener of ours, Troy Frericks, provided this. I just got a kick. The title of the Fox News story is "Suspected Hit-and-Run Driver Caught in Florida After Her Car Called Cops." So this is, again, sort of a...

Leo: What?

Steve: Okay. The future that we are sliding ourselves into. The story reads: "Police caught a driver linked to an alleged hit-and-run in Florida after her own vehicle called the cops, local media reported."

Leo: I've been in an accident. Quick.

Steve: Come help me.

Leo: AMC Gremlin does what?

Steve: "Investigators received an automated call from the Ford's emergency response system..."

Leo: That's so funny.

Steve: "...offering to let them speak with the driver if they pressed zero, according to [Station] WPBF. So a dispatcher talked to the driver, Cathy Bernstein of Port St. Lucie. She denied there had been a crash and said she hadn't been drinking, [the] police reported." I don't know if she offered that. I wonder if there's a breathalyzer in her sun visor.

Leo: No, I haven't been drinking. What do you mean?

Steve: Yeah. "But cops say they saw significant front-end damage to the vehicle when they went to her home. Bernstein then claimed she had hit a tree, according to police. Eventually she admitted to the hit-and-run, police said, adding that she was actually trying to escape from an earlier crash."

Leo: Oh, boy.

Steve: Her luck was really not with her that day. "Bernstein was arrested, WPBF reported." Yes, because her car turned her in. So, oh, boy, yes, Internet of Things. Let's all connect everything and see what happens. Unanticipated consequences.

So we've talked about Telegram a number of times. And I'm on record a month or two ago, just saying, eh, you know, the security's fine for texting your mom and telling her that you can't wait to see her for Christmas, but that already there's really well-known, well-vetted instant messaging platforms where we know how they work. And the really stomach-twisting thing about Telegram is that it's the most bizarre, homegrown crypto you've ever seen. I mean, it is deeply messed up. And the problem is somebody made it, after all of the way to do it right was already in the public domain. So it's like, okay, wait. What? So...

Leo: Yeah, why bother, yeah.

Steve: Yeah, exactly. Just use one of the many good ways to do it. So Jakob Jakobsen, who just acquired his master's in computer science at the Aarhus University in Denmark - largest university, second oldest university there - did his master's thesis taking a hard look at Telegram. And in the abstract at the top, he just says: "The number one rule of cryptography is never create your own crypto. Instant messaging application Telegram has disregarded this rule and decided to create an original message encryption protocol."

And we've talked in years past about the 13 year old who comes up with a bit scrambling algorithm, he says, oh, I've got this fabulous cryptography system, and it scrambles the bits up really good. No one is going to be able to unscramble them. It's like, oh, okay.

Anyway, so Jakob says: "In this work we have done a thorough cryptanalysis of the encryption protocol and its implementation. We look at the underlying cryptographic primitives and how they are combined to construct the protocol, and what vulnerabilities this has. We have found that Telegram does not check integrity of the padding applied

prior to encryption, which lead us to come up with two novel attacks on Telegram. The first of these exploits the unchecked length of the padding, and the second exploits the unchecked padding contents. Both of these attacks break the basic notions of security and are confirmed to work in practice. Lastly, a brief analysis of the similar application TextSecure is done, showing that, by using well-known primitives and a proper construction, provable security is obtained. We conclude that Telegram should have opted for a more standard approach."

So that's really the whole story. In his table of contents he shows a random padding vulnerability, reply and mirroring attacks in older versions, timing attacks, known attacks on primitives. Then there's experimental validations of the two padding attacks he mentions, and also shows a malicious server attack. So it's a catastrophe. And so essentially that's the story is that, yes, I don't know if he earned himself half a million dollars. Probably not. He probably just did his master's thesis and said, okay, I'd rather have my master's degree than win their bogus award because it's already been demonstrated that people offering awards isn't a proof of security. It's like trying to prove a negative. So the proof of security is having somebody who takes the time to tear something apart and find the problems, which this guy did. So again, I'm not telling anybody don't use Telegram. Just don't use it if you actually do want security because it isn't.

Okay. A couple miscellaneous bits. I mentioned, and it's funny because I was already, like, self-conscious about this, last week I referred to myself as a physicist. And what I meant was someone who likes physics, I mean, who's intrigued by physics, not that I have a degree in physics. So I just - I got a couple tweets from people saying, in fact, one guy said, "I was listening to Episode 537, and you said, referring to yourself, 'being a physicist.'" Then he says, WHAT?! And it's like, okay. So I just wanted to, for the record, yes, I'm not a physicist.

Leo: You are not a physicist.

Steve: I took advanced physics in high school, and in fact the poor teacher gave me my own project. I duplicated Millikan's oil drop experiment and didn't have to go, didn't have to attend class. I just took the final because that was, you know, mechanics. And I just - I understand the nature of mechanics of physics so well that, I mean, I could just do the test on day one. So he gave me a sort of a special project to work on, and I was the number one final in my freshman year at Berkeley in Engineering Physics 5. So, yeah, I mean, I love physics, but I'm not a physicist. So I didn't mean to say that I was. Just that that's what I meant.

A Chris Wronski tweeted something he, again, listening to last week's discussion of the AOL Desktop client and its unauthenticated FDO scripting language, and I was criticizing it for having no authentication, he reminded me that this does come from, it heralds from the days of point-to-point modem connections. And although I did refer to that, I wanted to make a point. It's like, yes, I mean, it's when they went to the Internet, that was time to enhance it with authentication, or scrap it. And certainly there's no better time than now, if anyone is still using that. But I just wanted to raise the point that, yes, correct, once upon a time there wasn't like the kind of packet-switched, inherently man-in-the-middle-capable connections that we have today, thanks to the way the Internet works.

Oh, and I got a nice tweet from the guy behind the GWX, the Get Windows 10 Control Panel. Ultimate Outsider was the site, and he just offered for us to contact him if we ever have any questions or comments, and thanked us for the shout-out. And finally,

someone else asked, he said, Steve, you keep talking about your - or actually I don't keep talking about it. I did mention my sci-fi PDF. And he said, "When was the last time you updated it?" And the answer is the last time I read a really good science fiction book. It's been a while. I read the Expanse series, and it was good, but it wasn't Michael McCollum or Peter Hamilton. It wasn't something that it's like, oh, my god, you have to read these books. That's what's on my sci-fi PDF.

And by the way, the series I could not remember last week when I was blanking on it, and that was the Lost Fleet series by Jack Campbell. That's a pen name, but the Lost Fleet series is, like, six or seven, and still there's even more books, "Beyond the Frontier" and all kinds of other things that he's working on.

I'll just note I tweeted to remind everyone yesterday that we had both "Childhood's End" and "The Expanse" series beginning. I liked the first two hours of "Childhood's End." I thought it was fun. I mean, I don't have super high expectations, especially coming from sci-fi, but this was way beyond the typical horrific quality of things that the Syfy channel produces. So last night was the first two hours. Tonight is the second two. And tomorrow night is the third two. And I thought it was great, for what it's worth.

Leo: Good.

Steve: I wanted to, for those interested in health, I got another comment that I thought I would just dip into quickly because I get this a lot. And it's on the topic of magnesium. Carl Engelbrecht said, "Steve, which magnesium supplement would you recommend?" And I guess I've talked about it sort of in passing. We've talked of course about Vitamin D in the past, and I've touched on other topics. Maybe, Leo, one of these days, when things calm down, when I catch up on my projects, we'll have some more time to sit down and create some more talk about health and my interest in it.

But for what it's worth, I believe that magnesium is another element which we don't get enough of. It's estimated that about 80% of Westerners are not receiving sufficient magnesium. And that's sufficient at the RDA level, which is, I'll remind people, not set for the level that produces optimal health, but it's the level below which disease states begin to emerge. So it's not like that's all you want. You want probably as much as you can get within reason. The problem is that crops are pulling nutrition out of the soil for themselves. NPK fertilizer is what's put back in - nitrogen, phosphorus, and potassium. But magnesium is not added. And so we're beginning to see mineral-poor soil, which is producing mineral-poor plants, and that's where we would normally be getting magnesium.

The trick about magnesium, that is, supplementing magnesium, is that you just can't take a magnesium pill, just like the raw material. So what is commonly done is it is complexed with, it is turned into a complex molecule by adding some kind of salt to it, basically creating a magnesium salt. And, for example, there's oxide, magnesium oxide, magnesium carbonate, citrate, malate, taurate, and so forth. The problem is that those disassociate those dissolve in our stomach, and then we're back to having the other component plus magnesium, and it's not well-absorbed by our intestinal tract. And the consequence of that is, if you take enough of it to be meaningful, you'll end up giving yourself diarrhea because osmosis pulls water into your intestines.

Leo: Milk of Magnesia. You're actually making that.

Steve: Yeah. And in fact that's the stuff, if anyone's ever done a colonoscopy, you go to the pharmacy and get an empty jug...

Leo: You drink it, yeah.

Steve: ...with some powder in the bottom. That's magnesium oxide. And so you fill it with water, shake it up, and start drinking it, a glass every hour or so. And before you know it, there's all kinds of rumblings, and you're all cleaned out and ready to have your innards looked at with a scope. So the point is that's not - you don't want to do that all the time. That's not good for you. But you may want to supplement your diet, if you're a person who likes to supplement, with magnesium.

So one company has figured out how to do this. It's a company called Albion Minerals, A-L-B-I-O-N Minerals. What they cleverly do is they create dipeptides. Instead of a salt, they take two amino acids and bind it to a magnesium molecule. And they prefer glycine because that's the smallest amino acid that there is, in fact it's the smallest one you can have. And so there's magnesium bisglycinate, they call it, or sometimes they use a glycine and a lysine molecule. The point is - and that's actually the one I take. Doctor's Best is the brand I buy. And the result is it is highly absorbable. When you ingest it, they've chosen the strength of the molecular bond such that it does not dissolve in your stomach. It gets down into your intestinal tract. And then you get to use active transport to get it into your bloodstream.

So the upshot is there's many different - Albion supplies the raw material to many different companies who sell it. There's a seal that they use called TRAACS. So you want to see that on the label. And so anyone's magnesium that is sourced from Albion that has a seal TRAACS, that's what you want. And you end up being able to take a lot more, if you wish to, before you see any negative effects on your digestion. And it means that, even if you're only taking a little, you're actually absorbing a much greater percentage of it, rather than it just getting washed through you. So there's our little health diversion.

Leo: Mm, mm, mm.

Steve: And finally, an interesting question that I ran across, also in the mailbag, from Juddson in Pennsylvania, with a question about SpinRite on SCSI RAID drives. He said: "We recently had a major drive failure on a RAID storage array at my place of work. Not sure of the configuration, but I don't think it matters much for my question. Apparently, one of the drives had a failure some time ago, and no one had noticed." Huh. And remember I told the story about that happening with my own employee, Sue.

"So when the second drive failed, people of course noticed immediately. It contained critical financial data and company records, so there was a bit of panic going around. I'm not on the sysadmin team, I'm just a developer on the software team, so I was not directly involved. However, I was aware of the issue, as were we all since this was a corporate server, and that one of the options on the table was to send the entire array back to the drive manufacturer for recovery, with a price tag upwards of $10,000.

"I own a copy of SpinRite," Juddson writes, "and pictured myself swooping in to save the day," he says, "With the assumption that my company would be more than happy to pay for a SpinRite site license, having saved them $10,000. Upon recommending the solution to our lead system administrator, I was essentially chastised for having even mentioned

SpinRite. The drives were SCSI drives, and it seems as though SpinRite, in the eyes of professional data recovery firms, is akin to a homeowner trying to fix their own septic system failure. You're just going to make it worse; leave it to the professionals.

"Is there," he asks, "any way that SpinRite could have helped the situation? I was imagining that we could pull the failed drive out of the array, mount it individually, run SpinRite on it, and return it to the array. Is there any way to do this easily with a SCSI drive? How do you respond to the notion that SpinRite will just make your problem worse, I assume by performing heavy" - I think he said "heaving." Maybe he meant heavy. Yah, so he wrote "heaving" - "read/write operations on the disc and thereby worsening the drive wear? I know that's not your position; otherwise, you wouldn't provide the product. I'm mainly curious as to how to respond in future situations when I experience SpinRite skepticism. As far as the array in question, we apparently were able to recover from a backup, which we initially thought had been lost, as well. I still wish we would have tried SpinRite, just to see what it could have done, if anything. Thanks."

And so, interesting question. And the answer is yes. I essentially did exactly that, although it was not a SCSI drive, it was just a SATA drive that Sue had in her mirrored RAID array. One drive died. She ignored the warning every time she booted her machine until the second one died, and then I got a call. So very much like happened there. And I really, again, I'm reinforcing my feeling that, boy, RAIDs, even RAID 5, it might just consider stopping. You know, like maybe waiting till 2:00 a.m. when nobody's doing anything, when there's no file activity, and then deliberately taking it offline so that it comes to the attention of IT, and they can say, oh, okay, we'll put it back online, but we're also going to rebuild the array with a new drive. That would be nice. Seems like we need something like that.

But anyway, so the trick here would have been to run SpinRite on the SCSI drive. You would have to have a non-RAID SCSI controller, just so that you could plug the drive into the system and run SpinRite on it. But that's all. And then you run SpinRite, it fixes it, as it generally tends to, and you're back in business.

So Juddson, thanks for the question. And, yeah, we can solve that problem, too. And I remember hearing a long time ago - this may be apocryphal, I don't know. But when I had a tech support department with a whole bunch of people, and SpinRite was in its youth, we did hear that professional data recovery services simply saved themselves a lot of time and trouble by themselves running SpinRite.

Leo: Yeah, which may be why they didn't want you to do it.

Steve: Yeah, they'd rather get $10,000, yeah.

Leo: Yeah, there's no way SpinRite could damage data. I guess if there were physical damage to the drive, any use of the drive could make it worse.

Steve: Yes. It's important to understand that all it's doing is using - it's very clever, but it's just reading. I mean, there's maintenance things it does. There's ways it has of reading incomplete sectors. I mean, there's a lot of depth to it, which is why it succeeds so much. But the only concern is, if a drive has really, like, got minutes remaining, then SpinRite could theoretically...

Leo: Every minute counts.

Steve: And it's more of a concern than something we've ever seen, but it could push the drive over the edge. But anything else was going to, too. Just trying to back it up would have done the same thing because SpinRite is just reading the drive.

Leo: You know, it does raise the issue, though, I sometimes tell people, before you start doing any file recovery stuff, you might want to image the drive. But the problem is that even isn't - because if you can't read the drive, imaging it's going to not read those sectors.

Steve: I was just going to say, many people run SpinRite when the drive won't image because the imaging program just stops cold.

Leo: I can't read it, yeah.

Steve: It just says, you know, like it just won't go any further.

Leo: Imaging makes sense if the issue with the drive is, you know, the file allocation table has gotten corrupted or something like that. Then doing a sector-by-sector image of the drive, at least you have that, that you could go back to. But there's so many other kinds of errors.

Steve: Well, and for example, you know, law enforcement, in forensics, the first thing that they do is make an image of a suspect's drive.

Leo: And work on the image.

Steve: So that they're able to have a chain of custody. They're able to say we did not alter in any way this drive. And that prevents the suspect's defense attorney from saying, you know, whatever. And then they work on the copy of the original drive.

Leo: Right, right. Somebody's saying if the drive won't mount, you can't image it. Yeah, if the drive won't mount, you can't SpinRite it, either. You do have to at least get to that point. That's kind of the problem with drive recovery. There's a vast range of issues.

Steve: Although it is the case that a drive that is online, but the OS can't see...

Leo: That's different.

Steve: That won't image, but that will SpinRite.

Leo: That's right, that's right. And what am I telling you? You know that. I'm agreeing. Yes, that makes sense, yes.

Steve: And after this sponsor break comes - I put it number one.

Leo: The question.

Steve: The number one question, perhaps the coolest question of all time.

Leo: Oh, so exciting. Does it have to do with the deity? He's looking around, what deity? Of which deity would you speak? All right, Steverino. I've got a Carnac here?

VOICE: Do a Carnac.

Leo: I'm going to do a Carnac? You didn't tell me what the answer is. Oh, let's see what the question is. I've received in an envelope - does this have to do with I've got the conn? The answer is, "Take the conn, No. 2."

Steve: No. 1.

Leo: And the question is - No. 1. Well, No. 1, No. 2, whoever. The answer is 42. I don't know why I was just handed that. But there are no Star Wars spoilers on this show, and I should have said that right upfront. Did you know there's a Google Chrome extension that will block all Star Wars spoilers? I know people, including our own Patrick Delahanty, our programmer, who are off Facebook until they see Star Wars.

Steve: Wow.

Leo: We're going Thursday, 22 people from TWiT, all of the TWiT staff, we're all going.

Steve: In two days.

Leo: Two days.

Steve: No, three.

Leo: Thursday.

Steve: Thursday.

Leo: Day after tomorrow.

Steve: Nice.

Leo: What? Twenty-six is the answer?

VOICE: No.

Leo: What's the question? Twenty-six of us are going. That's pretty much everybody now. That's more people than work here. Go ahead. I'm sorry, Steve.

Steve: I'm - I've always been more...

Leo: Are you not a Star Wars fan? You're a Star Trek.

Steve: I'm not. I liked the first, the original three.

Leo: Oh, and then it really went downhill. I agree.

Steve: I just, it's like, okay.

Leo: Jar Jar Binks [raspberry].

Steve: And apparently he's still in, like, in where we're going.

Leo: No.

Steve: Oh, yeah, there's Jar Jar is in there. I hope...

Leo: No spoilers. I don't think so.

Steve: I have it on pretty good authority.

Leo: Really?

Steve: Yeah.

Leo: Some people have seen it now because the world premiere was last night in Los Angeles.

Steve: Yeah. So I'm hoping what we're doing is starting a whole new franchise. I mean...

Leo: Like J.J. did with Star Trek; right?

Steve: Yes. Yes.

Leo: Which I loved the reboot.

Steve: Yes.

Leo: No, I - this is a great director, a brand new studio. I think this is going to be great. I like the new stars. Well, I'm not going to say more. Because there are actually some people who will not watch the trailer. They don't want to see...

Steve: I've watched Harrison on a couple of talk shows, and he's just so wonderful. I just love the way he's evolved. I mean, he's just - he's a neat guy.

Leo: I'm trying to [crosstalk].

Steve: And so, and it's fun to have him.

Leo: So will there be a drinking game for the sun flares? How early before the first J.J. Abrams lens flare?

Steve: Is it people's belief that there's, like, some reveal?

Leo: Yes.

Steve: I mean, like, is it just a nice story?

Leo: There will be spoilers.

Steve: Could someone say something where it would just destroy it for you?

Leo: I think so. In fact, I don't even want to say what it is, what the speculation is about because that could be a spoiler. But just between you and me - and there are some people who have not watched the trailers. But people are looking closely at the trailers, and they are seeing things that are missing, and they are questioning why.

Steve: Wow.

Leo: Some say I look at what is, and ask why not? But I think you - anyway.

Steve: You know? And for people who are that much into it, I think that's great.

Leo: Yeah.

Steve: You know? I mean, they're having a ball speculating.

Leo: They're serious, oh, yeah, it's fun.

Steve: I think that's wonderful.

Leo: That's really what this is all about. And there's a big sign on the movie theater that we're going to go see this at. We're going to go see it in 3D. There's a big sign saying you can wear a costume, but no masks or light sabers allowed into the theater.

Steve: And please, no really high hair.

Leo: Yeah, please.

Steve: Because that would be...

Leo: I'm concerned.

Steve: ...an alien with really high hair.

Leo: Because even though we have tickets, which guarantees us a seat, we don't have reserved seats, and there's going to be a long line. The line starts at 7:00 a.m. The show starts at 7:30 on Thursday. And we've got 26 people. I don't think you can go get two seats in the line and then 24 other people show up and say, oh, thanks for holding the place for us.

Steve: No, that would not go over well.

Leo: I don't think so.

Steve: So do you not have a reserved seating theater?

Leo: No.

Steve: Ah. We do. We're big on those in Southern California.

Leo: Oh, I would go for that.

Steve: Yeah.

Leo: I would go for that. Otherwise, because if you don't get in line, you're sitting in the front row. May not be a bad thing.

Steve: Yeah, and in 3D, like, aagh, in the front row. But, yeah, I think Star Wars you have to see in 3D. The good news is Jen loves 3D. In fact, she will not see a flat movie - I know. She will not see a flat movie if it's available also in 3D.

Leo: Really?

Steve: Yeah, she won't.

Leo: So she drags you to these 3D versions?

Steve: Yeah, and sometimes they don't have them at convenient times, like several of our theaters only do them in the evening. And I like to go in the late afternoon to avoid...

Leo: That way you can get to the Denny's Early Bird Special at 4:30.

Steve: That's right.

Leo: Very handy. You know, there's nothing like being a senior citizen. Both you and I are headed that way soon, sad to say. You know what, though?

Steve: You caught up about a week ago.

Leo: I'm 59. But I bought a camera that's older than me.

Steve: Nice. I like that.

Leo: This Leica M3 was made in 1955. I was made in 1956. So I don't feel so old when I use - I love this camera. It's a beautiful thing.

Steve: Actually, technically you were made in 1955.

Leo: No, born November 29th, 1956.

Steve: Okay. You squeaked out of the conception window.

Leo: Yes, now, well, the egg of which I was partially made.

Steve: That's true. That had been around for a while.

Leo: Been around since '33. So technically I've been through this whole century, this last century. Come on, let's get to the big question. You ready?

Steve: I was going to say, we rarely go down the rabbit hole on this podcast.

Leo: Well, Star Wars is worth a rabbit hole or two.

Steve: That's true.

Leo: This comes to us - oh.

Steve: And we may be back to only one question, Leo. We're at an hour and 40 minutes. And I could spend - I'll spend easily 10, 15. So I think we'll just do one, and we'll...

Leo: Don't rush.

Steve: We'll do the other ones next week.

Leo: This is not television, this is our own little fiefdom.

Steve: And I don't think anybody will be disappointed, so here's perhaps the coolest question of all time.

Leo: Comes from where my son Henry goes to college, Boulder, Colorado.

Steve: Yup, where it's very cold right now, I hear.

Leo: Just had a foot of snow. Henry says, "They closed all the classes. Some people got out of final exams, but not me."

Steve: Ah.

Leo: And it's Steven Bussinger of Boulder, Colorado poses the coolest question of all time: Hi, Steve. I realize this is not in your wheelhouse, but seeing as how you're interested in health, and you have a penchant and talent for explaining complicated electronics concepts to the layperson, I was hoping you could shed light on this for me.

I was reading a recent study about the circadian rhythm of cyanobacteria. It was in phys.org, but that's not important. At the end of the article, the study author elucidates a potential explanation for the existence of circadian rhythms. You know what's so weird, Steve? We were talking, I kid you not, about circadian rhythms with Dr. Kiki on Sunday on TWiT, and about the cave study.

Steve: Nice. Oh, yes, right, right, right, where they disconnected people from the 24-hour cycle.

Leo: Right. And I invoked your name because I was talking about the, what do you call it, the bifurcated sleep cycle?

Steve: Yeah.

Leo: Whatever that is that you do?

Steve: Or that people used to do before we had Republican debates and the second installment of "Childhood's End" to keep us awake late at night.

Leo: Yeah, that keeps you awake. Here's a quote from the end of that article: "In the historical development of electrical circuits, engineers found that synchronizing each step of a computation to an internal clock made increasingly complicated tasks possible, ultimately leading to the computers we all use today. Perhaps in the future we'll be able to use synthetic clocks in engineered microbes in a similar way." What?

What I'm trying to understand, and hope you can explain, is why synchronization makes more complicated tasks possible. Do you know what specific circumstance he is referring to in the realm of electrical circuits? I'm wondering if there's an inherent property of synchronization that makes more complexity possible, or at least makes current systems more efficient. Thanks for your help, and thank you for your show.

Steve: So I invite our listeners to hit pause. You heard the question. What is it about the way our systems work? Because when you think about it, all of our stuff has clocks. Every, I mean, everything has a clock which is, you know, and we talked about...

Leo: That's why they say 4GHz processor. That's the clock.

Steve: Exactly.

Leo: The original IBM PC had a 4.77MHz clock.

Steve: Right.

Leo: That's the clock.

Steve: Right. So, you know, it's always present. What is it about that? And so, you know, pause the podcast, and we'll be right here when you hit play again. Don't worry. So pause the podcast and think about your answer. What do you think? And then, okay, one, two, are you paused?

Leo: Pause now.

Steve: Pause now.

Leo: Okay. I'm pausing. Okay. Now, okay, I'm thinking, and I have not read this ahead or anything.

Steve: No.

Leo: I was thinking concurrency it makes sense because synchronization will allow

you to have concurrent processes without a race condition. But this is pre-concurrency. Why would you want it in lockstep? In other words, why would you want an add to take exactly the same amount of time as a move? Is that kind of another way of saying it? Like one operation takes exactly the same amount of time as the second operation? Actually, sometimes not because some operations take multiple cycles.

Steve: Yeah.

Leo: So it's not that, is it. Hmm.

Steve: What happened? And this is the reason I just thought, oh, what a fabulous question, is there was a change in thinking, probably around the middle of last century, like the 1950s, maybe the '40s, maybe the '50s. Anyway, one of the ways it's sometimes fun to attack these questions is to look at an extreme example. Imagine wiring, that is, with gates, you know, like coming up with an electronic solution to a spreadsheet. That is, where you - or an accounting system, that is, where you're like, that's what you're trying to create is an accounting machine, and you do it with wires and gates. I mean, you just sort of say, okay, well, you know, if I connect these things, this battery to this light, it lights up.

So if I'm going to do an accounting machine, I would have to have some sort of a register to put one of the numbers in, and then another register. And then somehow, like, add them using some circuits, and then show that in lights. But that's only, you know, that's not accounting. That's just addition. So my point is you can see how rapidly a problem explodes if the tool you have is like just electricity, or even electronics, but just like circuits. And so the answer to the question, what it was that happened in the historical development of electrical circuits...

Leo: Is it okay to unpause now? Oh.

Steve: ...is the concept of breaking a big problem into tiny pieces. I mean, that was the aha. That was the big thing that happened was, rather than just - because, for example, there used to be analog computers, where you actually had an integrator which was a constant current source feeding a capacitor. And the constant current increased the charge on the capacitor over time, which actually is integration at the rate at which the current flows. And then you'd have two of those, and you'd then have something else that summed the two charges into - and that was addition. But you literally did it with analog flows of current and voltage.

And so the big change - and the idea was that you set this whole thing up, and then you started it, and it worked out the problem. We actually went through a stage like that. But the thing that happened that suddenly required a clock was this concept of breaking an insurmountable, I mean, an intractably large, complex, electrical problem, essentially, into pieces, into events. And I've told the story before about Wozniak's design for the floppy disc controller on the Apple II, where I'll never forget looking - I had an Apple II, of course, because I did the light pen for it. I looked at the controller; and, you know, controllers were complicated things. They were typically, you know, the Apple, you could have a really long board that would even have to get cut off, it had to have its nose

trimmed off so it could slip under the keyboard, that sort of sloped down in the front.

So, like, there were graphics cards you could add that were just crammed with components. You couldn't get any more chips on these things. And here was Steve's design with, like, five chips on it. They were lonely chips. They were like, hardly even worth having a circuit board for. And I kind of wondered why they just didn't build it onto the motherboard, but they wanted, not everybody, back then you used cassettes sometimes, and you had cassette input and output audio, sort of like with a modem, in order to store and retrieve programs. So this was an extra cost option.

And the brilliance of what Steve did was he created essentially a microcontroller out of this just a couple chips. And when I figured out what it was he had done, I was just - I was so impressed with the details. But there was a period of time when circuit designs went from just being a bunch of gates laid out and interconnected to do a job, and then the concept, and people will recognize this, but it occurred earlier than people think, of microcode. That is, instead of just sort of having the gates do their thing, when we had access to even small memories, and that was the thing that was missing for so long, some kind of a memory.

Then what you could do is, just with two chips, you have a memory chip where you - and say that this was a ROM, because these were typically read-only memories. You would have a bunch of addressing pins, which addressed the location in the ROM, and a bunch of outputs. So, like, say, eight input pins, so you have 256 bytes in the ROM, eight inputs, and eight outputs. So you have eight bits come out, and a pattern of eight bits selects the location. Then we also had something called "latches." And so a latch was, again, one chip, where it had inputs and outputs. And when the clock, like, dropped, when the clock level fell, what was at the inputs at that instant was transferred to the outputs. And then even if the inputs changed, until the next clock edge, the falling edge occurred, the outputs wouldn't change.

And so the brilliance, and this is what Steve did, and designers were doing this at the time, was if you took the outputs of, for example, this four-bit latch and fed them into the ROM, and took four of the eight bits out of the ROM and fed them back around into the latch, what you have is a rudimentary computer. You have a rudimentary two chip, and they're not fancy chips, it's just a ROM that stores a pattern of bits and a latch that, on the falling edge of the clock, transfers its inputs to its outputs. Because, if this was a four-bit latch, and you've got four lines going into the memory, well, that's 16 steps. And then the other four inputs to the memory would be four circuits, four signals that you need to sense, and the other four outputs, or you could also use some of the circulated, the ones that are being fed back into the input through the latch. Those go do other things.

And the point is that what you have is something where, when that clock falls, the combination of the current four inputs to the latch, plus the state of the other four inputs to the memory, determines the next step. It is a very crude programmable state machine, and it transformed the way stuff was built because it would have taken a huge number of discrete logic gates and all kinds of crazy complexity with signal paths and delays and just confusion. And that was distilled into something very simple where - and if you got it wrong, you just changed a couple bits in this ROM, and it would fix it.

So we had the first concept of programmable electronics. And from the first instant that happened, we had to have synchronization. We had to have a clock in order to create phases in time so that in this interval of time we're in this state, and then in the next increment of time we move to the next state. So we created a state machine which could then accept other inputs and generate outputs in a way that was driven by data, a

pattern of data in the ROM; and you had a very early, crude, you really would be pressed to call it a computer, but the essence of the idea, the concept.

And what that did was instantly change the way designers created circuits. Not all problems could be solved in this way; but, after understanding the elegance and the power of the solution, it would be the first thing a designer would consider is okay, wait a minute. Or maybe three quarters of it. They could take three quarters of their problem and collapse it into a couple chips and then have to do some other things sort of the old-fashioned hard way. But it was an overnight change that we've never come back from.

The two things it needed, it needed the concept of a ROM, that is, the idea that there was somewhere to store this intent of how we move from one state to the next. Because that's really all PCs are. CPUs are big, very complicated state machines, where they're in a given state, and upon the event of a clock, the current set of inputs moves it to its next state. And of course this little simple two-chip solution itself had no memory. It just sort of had intent. It just expressed its intent and dramatically simplified the way circuits were designed.

Now, of course, generations later, there's a huge amount of context in the processor that controls what it does. So it's gone crazy. But the essence, which is what Steven was asking, was throughout, from the first moment this happened, it was this concept of moving from one state to the next. And it's not clear how synthetic clocks in engineered microbes might work, but it's not that far of a reach because it is an incredibly powerful technology to go to a way of breaking a problem down. It's sometimes like not as quick as if you sort of did a brute-force solution, but you can often trade speed for simplicity. And of course that's that solution.

So I just loved the question, and I thought our listeners would probably get a kick out of, well, first of all, seeing if they guessed what I was thinking as the right answer, and I've no doubt that the mailbag and my Twitter feed will fill up with people saying, you know, how about this?

Leo: Well, so I'm thinking, and a little clarification, I wasn't too far off with the notion of this concurrency notion because, for instance, an Enigma machine, which has no clock, but does calculations, the calculations are like a choo-choo train. There's the first one and the second one and the third one. And you don't need a clock because they just all succeed one another; right?

Steve: Actually, it was clocked.

Leo: It was clocked.

Steve: The Enigma machine was because it went from one state to the next, as quickly as it could. Everything advanced, and then the circuits were checked to see if it had a solution. And if not, then everything advanced again. And if a solution was found, then it just locked up, it stopped.

Leo: You're talking about Alan Turing's solution.

Steve: Oh, you're right.

Leo: But the original encryption machine was just cranked.

Steve: I'm sorry, you're right. I was talking about cracking the Enigma machine.

Leo: Cracking, well, that was going to be my question was...

Steve: Not the encipher machine.

Leo: Encipherment was in fact just a logical progression, didn't need clocking. You do this, this, this, and this. One thing follows another.

Steve: You might argue that - remember that when you pressed the key, it advanced the rotor.

Leo: So that's a state.

Steve: That's a state.

Leo: Yeah.

Steve: And so it moved it the next state. And so it was manually clocked by the operator typing on the keyboard.

Leo: So I guess, I certainly would understand how it would need to be like a choo-choo train. Car two must come after car one, and car three must follow car two. But that doesn't have to have a regular metronomic clock to that. That just means it has to be sequential; right?

Steve: True. And remember that we, for example, laptops, they have a variable clock. They slow down in order to conserve battery and heat sometimes because the clock causes a dissipation of energy.

Leo: So really it isn't so much clock as we think of it as a time.

Steve: Right.

Leo: As it is sequence. It needs to be sequential.

Steve: Right.

Leo: You can't do operations out of order because - then of course, as soon as you do concurrency, and certain some gates have concurrency in the sense that things happen, and then they interact, those would have to - you'd have to make sure those happened, not merely sequentially, but kind of in lockstep.

Steve: Yeah, and that's, if we go a little bit further, that's sort of the way the design has evolved is there are things called "propagation delays" where it takes a while for the circuit to assume its new condition and sort of settle down. And in fact, in my little simple example, where we had the latch whose output goes into the inputs of the memory, and some of the memory's outputs feed back around to the latch, back then the memories weren't very fast. So there was a maximum speed at which you could run that clock because you had to, when the clock occurred, if the latch's outputs would change, causing a different address to be given to the memory, so it would look it up and then send its outputs back around to the input. So you had to - so you'd do the clock, and then you'd sort of wait for things to settle down again to get ready for the next event.

Leo: Another way to think of it be more like a traffic cop saying, okay, wait. We're going to wait for everybody to be done. Okay, now you can go to the next step. Right?

Steve: Yes, yes.

Leo: All operations must complete. We are now in a known state. Next step.

Steve: Yes. Yeah. And, you know, a four-way controlled signal works that way. It moves through stages in a preprogrammed fashion with different amounts of time spent. But these people go, then those people go and so forth. So, and it's really good.

Leo: You need a traffic cop. Otherwise it all happens at once.

Steve: Otherwise it's a big mess.

Leo: It's that a Robin Williams line? Time is God's way of keeping everything from happening at once.

Steve: That's right.

Leo: By the way, and I think we're probably going to wrap at this point; right?

Steve: Yup, yup.

Leo: I did want to mention one thing that just broke, that apparently a security researcher, dfir-blog, has discovered a flaw in Kerberos, the authentication system used in Windows.

Steve: Oh, that's been around for ever.

Leo: Kerberos is really old.

Steve: I think MIT designed that.

Leo: Yeah. In fact, I'm kind of stunned that Windows still uses it. I'm reading an article from The Register, so I'm not sure, I mean, I'm just seeing this right now. It says the vulnerability cannot be fixed. The only solution is to use Microsoft's Credential Guard program to prevent passwords from being stored in memory. I'll have to read the blog post.

Steve: And we'll talk about it next week for sure.

Leo: Yeah, dfir-blog.com, Kerberos attacks. I seem to remember Kerberos attacks before. This is not new. But wait a minute, he says this is not a new flaw.

Steve: Ah.

Leo: I don't know why the media is reporting this as a new flaw. Okay, never mind. Forget I mentioned it. My mistake because I read the Register article. Then I went to the original. Always go to the source.

Steve: Yeah.

Leo: Because remember there have been Kerberos issues. This is ancient.

Steve: Yeah, well, yes. It was one of the very early solutions for doing secure online authentication.

Leo: Right. Okay. Never mind. Forget I mentioned it. Steve Gibson is at GRC.com, a great place to go if you want SpinRite, THE place to go, the world's best hard drive recovery and maintenance utility. Don't listen to those other guys who say, oh, you can't run SpinRite. That's just so they can run it and charge you $10,000.

Steve: Yeah. And, by the way, I'm happy to give people a refund. That's always been our policy. No questions asked. If you're not happy - some people buy it because they

think that it, like, reinflates their car tires. And Sue says, "Uh, no, sir, we'd be happy to give you your money back."

Leo: You can have your money back.

Steve: Normally it works for people, and so we're still here.

Leo: Actually, the interesting tie-in with the Kerberos flaw is it is a clock, it's an asynchronization flaw.

Steve: Yup. I was going to guess because it is a time-based protocol.

Leo: So this is the concurrent proof of why you need a clock, and a good one. We also can find many other fine things at GRC.com like SpinRite, like his SQRL research, soon to be a major motion picture near you.

Steve: Yup, getting close. The protocol's nailed down. I'm going to revise the client, revise the server, and then nail down some things that I just haven't gotten to. And then we'll be doing a fun podcast.

Leo: Lots of health information there, including the original Vitamin D research. Oh, my gosh, it just goes on and on. ShieldsUP!. Go to GRC.com. You'll also find copies of Security Now! there, the audio, the text file.

Steve: We have a menu across the top, just like many sites.

Leo: Look at the menu.

Steve: Just sort of float your mouse cursor and look down in there, and you'll find all kinds of stuff,

Leo: People probably say that. Well, I can't find anything. Where is it? Well, just the menu will - you can see everything's there. It's nicely organized. 16 and 64Kb audio at Steve's site of this show. The transcripts from Elaine Farris, those are great. We have 64Kb audio and a variety of video formats, as well, at our site, TWiT.tv/sn. You should also subscribe. All the podcatchers everywhere, iTunes, Apple TV, Windows, everywhere you can find it, Roku. Just subscribe so you don't miss an episode because you're going to learn. You're going to learn, as we do every week. Steve, have a great week, and I guess we have seven more questions we can do then.

Steve: Yup, we do. I'll add three to that, to the seven we already have, and we'll tackle them next week, along with - lord knows what the week's news is going to bring.

Leo: Steve Gibson. Talk to you next week.

Steve: Thanks, Leo.