Listener Feedback #185

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

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-449.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-449-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson, the Explainer in Chief, is here. Our topic of the day: Your questions, Steve's answers. There's a little bit of security news, too. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 449, recorded April 1st, 2014: Your questions, Steve's answers, #185.

It's time for Security Now!, the show that covers your security and privacy with this guy here, the Explainer in Chief, Mr. Steven "Tiberius" Gibson. Hello, Steve. Good to see you.

Steve Gibson: I should move my hand more slowly. It sort of blurs when I do that.

Leo: Yeah, Skype doesn't handle fast motion too well. We want to say that we are recording this on April 1st. It is April Fools' Day. There will be no April Fools' jokes in this show at all.

Steve: Not from here. And that's not a - we're not trying to set you up for one by telling you there won't be any. There actually won't be any. And I was thinking, well, had I planned ahead, April 1st, how often does that happen? Please don't answer that question. I'm sure we have people who...

Leo: We know the answer, Steve. Oh, to have a show on April 1st probably doesn't

happen but once every, say, seven years.

Steve: Yeah. There are - we have a neat podcast. It was a surprisingly quiet week from a security standpoint, which is perfect for a Q&A because normally, traditionally the Q&As have run long. But there was - we do have some additional news from a research project that the world's cryptographers did just to see how bad that NSA/RSA collusion actually was. And it turns out it's far worse than we knew.

Leo: Oh, no.

Steve: Then the question arose in sort of an unfortunate Computerworld column about whether Google's always HTTPS for Gmail was a bad thing or not. An interesting tip for installing impossible-to-enter-by-hand WiFi passwords in visitors' iPhones. The predicted collapse in cloud storage pricing. An odd question, or, I mean, a sort of an up-in-the-air question about advertising for Firefox. A bunch of miscellaneous stuff. And we have 11 listener and follower questions. I mentioned "follower" because I got about half of them from over the course of the week people tweeting things. I thought, well, that's a great question for the next podcast. So just overall, some interesting news. More potpourri and miscellanea than security because not much really happened this week.

Leo: Good.

Steve: So we certainly had three weeks of talking about iOS security. And, boy, I got a lot of...

Leo: Well, you know that next week it'll be the last day for XP.

Steve: Oh, yeah. It's funny, too, because I'm sitting here looking at my end of support Windows 7 whatever you call it applet.

Leo: Do you have a ticker?

Steve: Well, I remember when it was in three digits. I remember when it was, like, 385. And I was thinking - or even higher. And it was like, eh, it's never going to happen. And now it says 06. So, yeah. Anything that has a leading zero, I don't know why. But anyway, yeah. So next week will be that.

Leo: Tick tock, tick tock, tick tock. All right. Good. Well, this is going to be a fun show. I've got the questions in front of me. I've got your show notes. We've got a lot to talk about. So we will get right to it. And once again, don't worry, you don't have to - I think on a show like this the idea of having to parse through everything we say to see if it's made up is not a good thing on a security show. This is not the place for April Fools' jokes.

Steve: I completely agree. And it's just...

Leo: It's not that we don't have a sense of humor.

Steve: Well, actually, we got so much feedback from last week's fun with the cash-spewing ATM.

Leo: The spewing, yeah. A number of people said they spit out their lunch.

Steve: Yes, I heard that also, yeah. I mean, do not drink coffee while listening to the Security Now! podcast when we're on a roll. And so we took care of all the levity last week. And people ought to know me by now. It's really sort of not my - I'm too serious. I mean, I believe things.

Leo: Well, and the topic is serious. We don't want to, I mean, when you're talking security, you don't want any ambiguity or confusion in the content. It should be clear that we're talking...

Steve: So we're not doing that.

Leo: Yeah. We're serious about that. This is not a setup or anything. God, I hate April Fools' Day. Steve Gibson, Leo Laporte. There must be some security news this week.

Steve: Oh, unfortunately there is. So some of the things that we discuss that we consider to be vulnerabilities are sort of annoyingly theoretical. They're like the ivory tower academics have determined that, when the moon is in a certain phase, and the sea seems unusually quiet, if you inject a packet into a certain port at that time, maybe something can happen. I mean, some of these things just seem really obscure. And one of the most controversial recent issues has been this so-called Dual EC DRBG. That's the Dual EC as in Elliptic Curve. DRBG is Deterministic Random Bit Generator. This was that random number generator which for some reason became the default choice in RSA's BSAFE library suite, which covers all kinds of different languages. And it was an article in Reuters that then alleged that RSA had received $10 million in a deal whose terms we still don't know, but that it was from the NSA. And Reuters alleged that it was in order for RSA to make this random number generator the default.

Now, that was when there were four choices. This was the fourth one and was orders of magnitude slower than the others. And even at the time, security researchers were raising red flags because there were just too many unknowns associated with it. It was like, well, okay, why do we need an elliptic curve-based pseudorandom number generator? We've already got three good ones based on solid proven architectures.

Leo: Yeah, especially if it's slower, it's suspect, it's worse. Why would anyone pick it?

Steve: Well, and more than that, I mean, it's one thing to say we'd like you to make it an option. But the way the BSAFE library works is you can request any of them. But if you make no request, then you get the default. And I've often spoken of the so-called, my term, "the tyranny of the default." And that is that what's set as the default is what's most often used. I mean, for example, XP had a firewall in it from the first day. It wasn't until Service Pack 2 turned it on by default that we got any protection because, even though it was there, and everybody was telling people to turn it on, nobody did. So again, default settings might as well be the only settings available, for all intents and purposes.

A group, and I can't even enumerate their names, if you follow that link, DualEC.org, Leo, here in the show notes, that'll take you to a page where on the left-hand side it lists the cryptographers that were involved. And this is a who's who of the academic serious hardcore crypto community. I mean, among them we've got Dan Bernstein and Matt and people who we're often talking about. What these guys did was they said, okay, let's quantify the risk, which the presence of the Dual EC DRBG actually represents. So to do that, because the code that contained this was not open source, they had to reverse engineer and decompile, disassemble the RSA BSAFE package; Microsoft's SChannel, which is the secure channel library in Windows; and fortunately OpenSSL is open and open source, so that one they didn't have to reverse engineer.

What they did was they said, let's assume that the NSA does know something about the elliptic curve that was chosen for this. In order to find out what that means, let's replace the elliptic curve that they may know about with an elliptic curve that we do know about. So the point is that it's very trivial for the cryptographers to choose specific elliptic curves that have weaknesses that they know about, very much the way the NSA may well have chosen this particular elliptic curve that we got from them. And so what these guys did is they reverse-engineered the BSAFE package to find the elliptic curve spec inside and changed it to one whose weaknesses they understood in the same way that the NSA may understand the weaknesses of this particular one. And they did that for the RSA BSAFE package, Microsoft's security suite in Windows, so-called SChannel, and OpenSSL.

So in their paper they said: "Major findings are as follows: The RSA BSAFE implementations of TLS make the Dual EC backdoor particularly easy to exploit compared to the other libraries analyzed. The C version of BSAFE makes a drastic speedup in the attack possible by broadcasting long contiguous strings of random bytes and by caching the output from each generator call. The Java version of BSAFE includes fingerprints in connections, making it relatively easy to identify them in a stream of network traffic."

SChannel, that's Microsoft's security suite, "does not implement the current Dual EC standard. It omits one step of the Dual EC algorithm. This omission does not prevent attacks; in fact, it makes them slightly faster."

And then finally they said: "A previously unknown bug was discovered in OpenSSL that prevented the library from running when Dual EC was enabled." So it wasn't even functional, actually. It was, like, installed, but apparently no one ever tried to use it. They said: "It is still conceivable that someone is using Dual EC in OpenSSL" - although, frankly, this is me saying I think that's probably unlikely - "since the bug has an obvious and very easy fix which was applied in order to evaluate the resulting version of OpenSSL, which the paper calls 'OpenSSL-fixed.' OpenSSL-fixed turns out to provide additional entropy with each call to the library. In practice, this additional input can make attacks significantly more expensive than for the other libraries." So that was a good thing.

But they found something else that was, again, sort of chilling. They said: "Evidence was

discovered of an implementation of a non-standard TLS extension called 'Extended Random' in RSA's BSAFE products. This extension" - so this is an extension to the TLS protocol. "This extension, co-written at the request of the National Security Agency, allows a client to request longer TLS random nonces from the server, a feature which, if enabled, would speed up the Dual EC attack by a factor of up to 65,000."

And I thought, okay, wait a minute. Extended random, what's that? So I tracked it down. There is online an IETF draft, and sure enough, it is coauthored, one of the authors is National Security Agency, the NSA. And under the rationale they explain: "The United States Department of Defense has requested a TLS mode which allows the use of longer public randomness values for use with high security level cipher suites like those specified in Suite B. The rationale for this, as stated by DoD, is that the public randomness for each side should be at least twice as long as the security level for cryptographic parity, which makes the 224 bits of randomness provided by the current TLS random values insufficient. This document specifies an extension which allows for additional randomness to be exchanged in Hello messages."

So to unwrap that a little bit, remember that, when we've covered the way SSL/TLS protocol operates, that each side generates its own random nonce and provides it to the other. And they use that for their ephemeral key agreement technology for building a shared key such that nobody who's eavesdropping who doesn't change the data, that isn't an active man in the middle, is able to get in the way and to figure out the secret that they then are able to share. So it completely makes sense that, I mean, like this statement about why this so-called "extra" or "extended random" would be useful is incontrovertible.

At the same time - and I guess the NSA would be the means through which this is done. But what they realized was, by having RSA support this, a client was able to ask for more randomness from the server and thus obtain much more state information. And as they actually demonstrated, by pulling this off, they were able to get a dramatic acceleration in cracking. And this paper, and I saw you had it on the screen there a second ago, Leo, at DualEC.org, anyone who's interested can read the summary. I found the PDF, and then I couldn't find the link to it again. But it shows the time in minutes required to crack the TLS connections.

Leo: It's kind of stunning.

Steve: It's frightening, yes. I mean, it's horrifying.

Leo: Well, the only one that's any good is the fixed OpenSSL.

Steve: Correct. And as they noted, the fact that additional entropy is added. And the programmers of the OpenSSL library must have done that because they thought, eh, you know, we're not so sure where that came from. But what did it do? Like the worst one looks like, as I mentioned, the C library of BSAFE.

Leo: I don't even know how much time 0.04 minutes is, but it's not a lot.

Steve: No, no. Four hundredths of a minute.

Leo: It's less than a second.

Steve: So less than a second. No, wait, four hundredths of a minute? Not less than a second.

Leo: Yes.

Steve: But not much more.

Leo: No, there's 60 seconds in a minute. Oh, yeah, okay, four one-hundredths is one 25th of a minute, yeah. So it's a couple of seconds.

Steve: Yeah, couple seconds, bang, thank you very much.

Leo: Boom.

Steve: Yeah.

Leo: Now, the best, though, is, what, 2^83 - I can't even read that number. Big number.

Steve: Right, that's a big, big number. So it doesn't really affect OpenSSL because it's broken in OpenSSL, and it's not turned on by default, and if you turn it on, it breaks it. So anyone who thinks maybe it would be a good idea probably just turns it off and goes back to what everybody else is using. Also, BSAFE, they believe they were able to fingerprint servers that were using BSAFE. And looking across the entire 'Net at servers offering SSL, it wasn't effectively present. But the danger is people using BSAFE in their own security suites or in their own VPNs or link packages or things, I mean, the idea is this is a crypto library that you use for building into proprietary crypto systems, which is precisely what the NSA would like to have access to. And even if they had compromised a certificate authority or were running some sort of HTTPS proxy, that wouldn't be able to intercept proprietary crypto.

And so the RSA was where you purchased, historically, I'm not sure that's the case any longer, where you purchased your crypto libraries to build proprietary solutions. And if this Dual EC DRBG was the default random number generator, and somebody was bringing up a proprietary link between corporate infrastructures, for example, you'd use TLS. In which case, wham. Given that it is true that the NSA knows something about the particular elliptic curve that they provided to NIST in order to assemble this, and it's now seeming suspiciously like they do.

And again, there's no reason to believe that this, like this extended random is an additional, deliberate hook. But, oh, my lord, you couldn't design something better to give you this leverage than to say, oh, let's put this into BSAFE because it's an extension to TLS, and we want to be fully compliant, even though it's not a standard. It hasn't even

been adopted yet. But it's a way of the client telling the server, tell me more about your random numbers. And as these guys again demonstrated by actually doing it, it provides basically a 16-bit shortening of security. That's where that 65,000 comes from, 65536 times easier to crack this.

So, Leo, I hate this. I mean, the reason I mess with computers, and I code, is that they obey rules, and you know what's going on, and you have a deterministic world. And so suddenly now this aspect of computing that is so important has been just thrown into a huge gray area, where we just have to say, well, we don't know. I mean, maybe. But we don't know what's going on. It's disturbing.

Leo: Yeah.

Steve: So there was a weird Computerworld article. And it's called the Reality Check column. And unfortunately I think that Robert Mitchell, who writes it, needs maybe his own reality check. Maybe he was at deadline. I can't explain this. But he argues that - he actually does argue in this column that Google turning Gmail on forcefully for everybody is bad.

Leo: What?

Steve: Yes. That's what he says: "You WILL use HTTPS." And so he argues that user choice is being removed.

Leo: Yeah, the choice to be less secure is being removed. That's true.

Steve: Yeah, yeah. And that response time is hindered.

Leo: That's not true.

Steve: And the fact is, it's not true. No. I mean, first of all, nobody is spending more time minimizing response time than Google, with the proprietary protocols they're moving on, SPDY and QUIC and all the things they're doing. And we already know that the only time you get a hit from using SSL, HTTPS, TLS, is during the initial handshake, and that from that point on successive connections have no detectable overhead. And now, with computers as powerful as they are, even that handshake, especially as we're moving away from RSA to good elliptic curve algorithms, that's been reduced by an order of magnitude. So it's just like, what?

And so it's funny because when I went back, after several people tweeted me the column, saying Steve, you were thinking this was a good thing, what do you think, and I read it, and I thought, okay - well, anyway, I went back. And apparently I'm not the only person to think that Robert was a little bit off the mark with this because he did comment that he'd received a lot of similar feedback and quoted one person who just basically said, you know, no. It is a good thing for everybody to have this turned on. I mean, more than by default, it's just got to be. And it's easy to understand what a benefit this is, if it's only marketing, for everyone since last week when this happened to know, that if

you're using Gmail, you have HTTPS. End of story. I mean, that's just - that's comforting that there isn't any way to - that Google will accept a non-HTTPS connection from your browser at no point in any scenario. So, yeah. And that's not...

Leo: But I don't have the choice.

Steve: I know.

Leo: I want the choice. Of what, I don't know, really.

Steve: I got a tweet just yesterday, so I haven't had a chance to dig deep into this, and I provided links in the show notes so anyone who's interested can. Karl Kornel, who tweets as @californiaKARL, K-A-R-L, he sent me, actually he first sent me his own document that he shared with me, or he created a public link in Dropbox, which it's essentially an XML script for Apple iPhones which he designed to allow visitors to instantly configure their iPhone for his WiFi network that has a Security Now!-style password, meaning one he may have gotten from GRC.com/passwords, that it's virtually impossible to enter it into any piece of equipment because the only way you're ever able to do so is by copy and paste. And the problem is, when visitors come over, they want to be on his network, use his WiFi.

So it turns out that Apple has, in support of corporate use of iPhones, a very comprehensive configuration utility which produces these, I don't know if you call them "manifests," but they're like configuration XML files, that allow all kinds of really deep cool configuration, where you can, like, instantly apply email accounts. You can restrict the application. You can lock applications. You can provision WiFi passwords and so forth. So I just thought, hey, this is the kind of thing that's absolutely going to appeal to some percentage of our listeners. And I wanted to pass that on because it's just that there's utilities for Windows and Mac, Mac OS X, that'll allow you to step through this and set these things and then produce these outputs, which you then allow an iPhone or iPad to digest and to lock it down or preconfigure it in various cool ways. So links in the show notes to that. They're nothing that I could ever repeat.

And I've promised that I'm going to update everybody on the current state of Trust No One cloud storage. The podcast we did years ago, where I ran through all of the ones that were either current or coming, like they may have been in beta, was super popular. And naturally there's been a huge amount of change since then. For example, Jungle Disk has gone through some changes, and the world has changed. So I've already been taking notes of all of the services that I intend to review. And I'm getting people who have been tweeting since.

But I noted that, as expected, Amazon immediately followed Google's huge reduction in cloud storage pricing. And not to be outdone, and actually because Microsoft explicitly said they would, Microsoft has also followed suit, followed Amazon right down, in some cases matching or even undercutting Amazon's pricing. So, and as we talked about, remember, last week we looked at the price of, I think it was a 3TB drive, at what that cost per gig and what the service providers were charging. And we saw that even at Google's new super low pricing, they paid for the storage in, like, three or four months. So, yeah, it absolutely makes sense that pricing was ultimately going to come down to this level. And it's because of that, because, I mean, cloud storage is now - it makes so much sense that we really do need to come back and look at the range of solutions that

are available.

Then I picked up sort of a distressing story about - and I guess this is a couple months old, Leo. You may have already seen this or run across this, this notion that Mozilla will be adding advertising to Firefox.

Leo: No, that's news to me.

Steve: Okay.

Leo: I don't use Firefox, so I don't really pay that much attention to it, I guess.

Steve: Right. Maybe you can explain something. Why is it that Google is Firefox's major benefactor, to the tune of $300 million a year?

Leo: See, that's the interesting thing about Firefox. It's actually quite profitable because of the Google search box. And it's not just Firefox. Safari also benefits. Any browser that uses the Google search box gets the benefit of Google ads. So, yeah, it's actually been very good for the Mozilla project.

Steve: And so it's good for Google, too.

Leo: Absolutely. It's a very nice win-win.

Steve: Okay. So here's the deal.

Leo: Now, you don't have - by the way, if you're a user, and you say, I don't want Google, you can change the default search to anything else. And then Mozilla will get no money from that.

Steve: And we know about the tyranny of the default, how many people are going to do that. Mozilla has talked about advertising in the past. And naturally there's been a huge uproar from users who are really worried about what that means. So Mozilla is approaching this again. And Mitchell Baker, who's the chair of the Mozilla Foundation, defends Firefox's new ad program and explains a few things. Okay. So what we know is that what they're wanting to do is to put tiles on the New Tab page. So when you say Open a New Tab, and I do that all the time, I get - I think I get just Google. I think I maybe get a tab - in fact, I've got it right here. Let me see what I have. I click on Open New Tab. No, it's a blank page. So it's waiting for something to happen, completely blank. What they're wanting to do is to put something there.

Leo: Ah. Well, that's not so bad.

Steve: And I have no problem with that.

Leo: If it's not on the page as I use it.

Steve: Correct.

Leo: Yeah, that's not so bad.

Steve: So the idea is when you create a new tab, there will be something there, rather than what right now is just a blank page.

Leo: Now, I should point out the reason they're considering this is because their deal with Google is about to run out at the end of the year.

Steve: Yes.

Leo: And they're concerned that Google won't re-up. And that would be a very significant blow to the Mozilla, I mean, you lose $300 million a year, I can see why they'd be worried.

Steve: Yes. And that's exactly right. In December of 2014, that deal is over. And so I look at this as, okay, do I mind nine large tiles? And they're saying, two of which might be sponsored. In general, they're wanting to give people links to things they really believe would be useful. But they're saying, yes, and we'd like to survive. And from my standpoint, my lord, if it's a matter of losing Firefox or putting up with some tiles on the New Tab before I put in the URL that I want to go to, thank you very much. I'm happy. You can make the tiles smaller and give me more of them, if you want. And no tracking. They are blocking any tracking so that, if you click on that, the redirect that occurs has all tracking information explicitly stripped from it, and you are not tracked. So, I mean, I believe this makes sense as a compromise. And so if at some point my new tab, when I update to v33, I think we're at 28 now, and it shows me some tiles, hey, fine. If it's that or losing Firefox, I'm happy to do that.

Leo: I wouldn't be surprised if they add a switch that can disable that anyway.

Steve: I agree with you.

Leo: That seems like something they would - it is an open source project still. And I should point out that Chrome in effect has an ad for Google. When you create a new tab you do get smaller thumbnails of your most visited pages. But right, front, and center is a big Google search box and a link to Gmail. I mean, that's in effect an ad for Google; right?

Steve: Yeah, yeah. Okay. So, and this is - I'm wrapping up with just a crazy posting. And but there's a message here. And this was someone tweeted it. It's TheVarGuy.com's blog. And he said - the title of this entry was "Why Windows XP's Demise Is Bad for Linux and Open Source." And I thought, what? Anyway, so basically he's explaining that he runs Windows XP in a VirtualBox VM under Linux. And he uses it to edit large book manuscripts which cause Libre Office to collapse. But Microsoft Office doesn't. So he runs Microsoft Office in Windows XP to edit large book manuscripts and to play something called Age of Kings. So then he says that Win XP is so much leaner than Vista or Windows 7 or Windows 8 that moving up will be impossible.

And so this is a perfect example of the point I'm going to continue to try to make as I explain what it means to have the constant patch drip cut off next week to Windows XP. And here's a perfect example. The guy launches XP to run Office to edit a book manuscript. So there's no vulnerability introduced there. None. Or to play some XP-hosted videogame which he is enamored with. Again, it's like, okay, that's not a problem. So all I'm trying to say to people is, relative to this end of the XP patch drip, is use your heads. Think about what this means, rather than assuming that it's the end of the world.

My contention, and as I have said, we're going to have an interesting period of a few years, starting next week, to see if I'm crazy or if I knew what I was talking about, that this is a tempest in a teapot; that it's a good operating system, and that everything that it uses to connect to the world is still being patched. And those are the things that represent the lion's share of the vulnerabilities in the operating system. We'll see. We don't know what vulnerabilities may have been held back as this is approaching. Anyway, I could be wrong. We'll find out.

Leo: It's coming.

Steve: Now, Leo...

Leo: I suspect we'll talk about it next week.

Steve: Oh, yeah. So, Neil Young's Pono.

Leo: Oh, yes. I invested.

Steve: I'm wondering - I wanted to alert our listeners, only as a public service, because the thing has just gone ballistic.

Leo: Oh, yeah.

Steve: This is Neil Young's ultra-high-resolution music player, which he is - it's a Kickstarter project. So people can put - it's P-o-n-o, or put "Neil Young Pono" into Google. I'm sure you'll go there.

Leo: Oh, yeah. Just "Pono Kickstarter" will get you to the Pono. It's Hawaiian for "righteous."

Steve: Ah, okay. They wanted to raise $800,000 in order to create this. They're now at $5,237,804 at this point, with 13 days to go. So you have two weeks. Check it out if you're interested. What it is, it's attempting to be, as the name sounds, an ultra-high-resolution music player, meaning that rather than sampling at 44.1 or 48 kHz, they sample at 192 kHz. And rather than digitizing at 16 bits, they digitize at 24. I was interested in the technology, so immediately dug into the digital-to-analog converter chip that they're using, which actually it's a 32-bit D2A. And my lord, I mean, that's insane.

Leo: It's a good DAC. This company apparently is a very good company that's doing the DAC.

Steve: Yeah.

Leo: It doesn't do any sampling on its own. We should make this clear. It's a player.

Steve: Correct.

Leo: The point is it can play back FLAC files, which is an open-source lossless compression format, in as high as 192 kHz, 24 bit. And the reason Young thinks this is even viable is that many albums are now recorded at high resolutions like that. Of course, when you sample it onto a CD, you make it much smaller. CD quality is 44.1 by 16. And even though it's not compressed. So this is not about compression. This is about sampling rates.

Steve: Right.

Leo: I bought it just because it's 400 bucks. Actually they have one, but they sold out, for 300 bucks. And I want to hear it, if it sounds that good. There are those who pooh-pooh it. I thought you were going to pooh-pooh it.

Steve: I have a link there to a great technical article...

Leo: Yeah, I read that.

Steve: ...that argues that, whereas 24 bits of sampling size may be useful, the article argues strongly and technically and acoustically that the 192 kHz sampling is not beneficial, and in fact may be detrimental. So I didn't go through it in detail, but I just wanted to give our listeners a heads-up because, wow, if it's that popular, we ought to know about it.

Leo: This article is by the guy Monty, who created the Ogg Vorbis compression format.

Steve: So he knows a little something about it.

Leo: He knows a lot about acoustics. On the other hand, and I don't know if he would argue this, there's no question that the compression, including Ogg Vorbis, which is not a particularly, in my opinion, good compression technology, takes a lot of the oomph out of these things.

Steve: Oh, absolutely. Absolutely.

Leo: And it's my opinion that a higher sample rate - you know, this goes to that "golden ears" thing because people say analog is better than digital.

Steve: Well, and also the idea that you're wearing ear buds in traffic.

Leo: Well, that's not what this is for, I've got to tell you. This is not about wearing ear buds in traffic. They don't even sell headphones because they say you need to have very, very good headphones. And it also has two connections, one that's appropriate for your stereo. So you can play your Pono into your Sonos, which is what I plan to do.

Steve: Oh, nice.

Leo: Then I'll have a Pono Sono.

Steve: Okay.

Leo: And that can't be a bad thing.

Steve: And I did want to note that the Typo keyboard is threatened by a judge having granted an injunction which BlackBerry brought. I'm not surprised because the design is a direct ripoff of the BlackBerry keyboard. I mean, anyone who has used a BlackBerry and looks at the Typo keyboard says, oh. I mean, it is. And I had never really looked into design patents. I knew that there was a term because all of my patents have been invention patents. They're software or hardware or something.

But two lines from Wikipedia about this said: "In the United States, a design patent is a form of legal protection granted to the ornamental design of a functional item. Design patents are a type of industrial design right." And the second line is: "A U.S. design patent covers the ornamental design for an object having practical utility. An object with a design that is substantially similar to the design claimed in a design patent cannot be

made, used, copied, or imported into the United States. The copy does not have to be exact for the patent to be infringed. It only has to be substantially similar."

Leo: Well, it is.

Steve: Which is, oh, boy, baby. So, yes.

Leo: That's why you like it.

Steve: Exactly. And in fact I gave one to Mark, my friend, and one to a Starbucks friend of mine. Both are ex-BlackBerry users. Neither can live without it now. And I feel the same way. I mean, it is so much better than typing on the touchscreen. It's funny because I can tell - Mark uses his iPhone on his bike for, like, sports instrumentation. And so he's moving it in and out of the case. And sometimes he tries to send me a text when he's not using the Typo keyboard, and I always know because there's typos.

So anyway, I think it's probably gone. I think any judge or jury looking at this Typo keyboard and assuming that BlackBerry has design patents - as they must in order to have brought the suit and for a judge to have granted a preliminary injunction barring the sale, the further sale of the Typo. And understand, that's a very high bar to meet. No judge does that cavalierly because a judge would only do this if he or she was absolutely, I mean, virtually certain that the final judgment would be the same because they are clearly risking dramatically damaging the company against which this injunction is granted, in this case the Typo keyboard company.

So I don't know if they'll change the design to be non-infringing. Apparently, and actually I got this from your discussion of this on TWiT on Sunday, Leo, yours and John's. Or somebody else brought it up. I think that someone did, anyway. I think it was news to John, so it wasn't he, that they had originally approached BlackBerry and asked for licensing rights.

Leo: Yeah. There was their mistake. They kind of raised some attention there, yeah.

Steve: Yeah.

Leo: It looks just like the BlackBerry keyboard. That's true.

Steve: It is, yeah. It's not as good as the BlackBerry. It doesn't have the width to be as good, and the construction quality is not as great. You saw that BlackBerry is bringing back the Bold, by the way, because...

Leo: Really. They're desperate is why, I'll tell you.

Steve: Yeah. They're rolling back and going to a previous phone that they're now saying economies of manufacture allow them to make profitably, which apparently they weren't

before. And it's outselling the newer junk that they have. So it's like, okay, yeah. I mean, I'm part of the iPhone iOS ecosystem now. There's no way I can go back. I've got iMessage groups of people who all have iOS devices. I love the fact that all of my pads and phone are synchronized. And I wish there was a gateway from my PC. That's annoying because oftentimes I'm wanting to send, like share a bookmark over through iCloud to iPad so that I'll pick it up when I'm next out. But, yeah, I can't go back, much as I like the keyboard. So yes, Leo, I am glad that I have a few spares in the refrigerator.

Leo: Oh, you heard me. I was - oh, I'm embarrassed now. I mentioned that a little bit - I was not mocking you. Well, I was maybe a little on TWiT this Sunday.

Steve: No, Leo. I have...

Leo: Because it's not untrue.

Steve: I have HP calculators. Everywhere I turn there are HP calculators.

Leo: What?

Steve: Because, I mean, I've got them in drawers.

Leo: And of course you loved, you must have loved Dvorak's response.

Steve: Here's two more in cases.

Leo: I wasn't making this up, folks.

Steve: No. No, this is the best calculator that was ever created, and I don't ever want to risk being without it. So I have them all over the place.

Leo: But I stand corrected. They're not in your freezer.

Steve: It's pretty cold here. No, those are not...

Leo: Do you still have the - you used to have Palm 7s or something in your...

Steve: Actually the Palm Tungsten, I think it was. No, it was the one that - anyway, I was in love with that. I read a whole bunch of eBooks on that, and I thought, you know, I never want to be without this. Well, some of these investments don't turn out the way I expect. Some of them do.

Leo: We love you for it.

Steve: I'm very happy that I have all the HP calculators that are available.

Leo: Absolutely. And the Typo keyboard because you can't get it now.

Steve: And the Typo keyboard. I have a - although, here's the problem, iPhone 6 is expected to be substantially larger.

Leo: Yeah, I have to get that.

Steve: And so, so much for the Typo keyboard.

Leo: Well, they haven't gone to trial yet. That was just a preliminary injunction. That doesn't mean that they won't be back.

Steve: Yeah. Does not look good, my friend.

Leo: Doesn't look good? Yeah.

Steve: No.

Leo: By the way, Dvorak's response to that was priceless. And I think you did see it.

Steve: Oh, the timing, the delivery, it was perfect.

Leo: Just like, flat. And then, "Gibson's nuts." Which we made the title of the show, in your honor, I must say.

Steve: Yeah, it was wonderful. So speaking of nuts, let's talk about SQRL briefly. We are now heading toward 52 languages. We're at 51. And I was just asked to add Indonesian to the lineup. So the person, I sent a message back to the person who asked me for Indonesian because that takes us to 52. The user interface is coming to life. And just so that people understand what you'd expect from me, in looking at the translations that we have so far, I see that the Asian logo graphic font, or glyphs, they tend to be extremely dense, but they would do better with larger font area, with a larger font size. But so they would like to have more height, but they don't consume nearly as much length because they're so expressive in the individual glyphs.

So the technology that I have, basically what I've been doing is, as you know from last week, I talked about the binary search and the support for languages and how the SQRL

client will be able to very quickly extract whatever strings it needs in the language that it's set to use in order to use them. So what I've done is, in order to properly render this wide array of languages, I have several stages that the application goes through after startup, before it even appears on the screen. And it's pretty instantaneous. I don't think anyone will notice that anything has happened. And that is, it takes a - in the user interface, and all of the UI panels are there at GRC.com, this is the most interface-heavy app I've ever written. Most of mine are like a single panel that it has a couple buttons on it, and it tells you what you want to know; or the DNS Benchmark I used the rich text format control to create some scrollable dialogue.

But SQRL leaves all of those behind. There's a ton of UI because I want to take people in baby steps through it. I wanted to make it easy to use and also have it explain itself as it goes. So I start off by taking one of the longer strings, which fits within a rectangle in the UI, and render in the target language that string into the rectangle. And Windows, of course, handles line wrap. So that'll tell me how large in the target rectangle that language's equivalent is.

And so the first thing I do is increase the height of the font, pixel by pixel, until that reference string no longer fits in the rectangle. Then I back it back by one pixel. So what that has the effect of doing is it allows a font which is more expressive because it contains more content in a smaller area. It gives it credit for the fact that it doesn't end up being as long by allowing it to be higher and still fit within the allotted space. So that sets the height for the font. Then the client rifles through every single string which is wrapped like that, that is body copy text, to verify that they all fit within the target area that they're allotted. And if any one doesn't, then the entire dialogue expands by 20 pixels. And then we test it again; 20 pixels, test it again; 20 pixels, test it again. So until that particular one does fit.

And then I continue. I don't go back and start, I just keep going because of course all the other ones up to then fit in the regular size. So once we're through with that pass, what we have is we have a font which is as large as it can be, assuming the default dialogue. And if we're looking at a language which is substantially longer than English, and there are a number of them, then the user interface has been stretched horizontally such that all of the strings in that language fit within the designated space. And once that's done, and that is all about sort of the body copy text, the UI also has a headline and a subhead on many of the dialogue pages. And so after the width has been set, then I go back and find the largest font size where all the headlines can fit within the headline size and all the subheads can fit within the subhead size. We lock it all down, and then the first dialogue appears.

So it's handling multiple languages the way you'd expect me to handle them, which is I think it will, right out of the gate, work well and be viewable for everyone. And I forgot to mention that, before doing any of that, I also scale to the font size which the user has set on their system. You know how Windows you can have it set to, like, large fonts if you want everything to be larger. So I first scale everything up to that, then go through all this. So it also honors the user's individual local font size setting. So, and it's working, by the way. What I just described yesterday came to life. So because I have so much UI, and I wanted to do justice to handling multilingual stuff correctly, I built a user interface engine, essentially, that allows me to create a description of the individual UI panels which it then renders, taking the specific language that it's rendering into account. And it's looking good.

And I did get, yeah, I got a nice note from some guy named Andreas in Sweden. On the 18th of March he sent it. And he said "Hi Steve," and he says, "and Leo," just to include you, Leo. He says: "I like to thank you for SpinRite. A while back my MacBook Pro would

not boot up. The funny thing is that I rarely shut down my computer at the end of the day. I usually just put it to sleep. But on this occasion I did shut it down. As I did, it hit me that my last backup was a couple of days ago, and I had some files which had not been backed up. But I brushed it off, thinking I could back up these files once I start my computer the next day. And what could go wrong? It will not be any problem.

"Boy, was I wrong. When I came back and tried to start my computer, it would not boot into OS X. How hard I tried with recovery disks and other tools, but nothing would help. I was able to boot into my Windows partition, so I knew that something was wrong with my OS X partition. In Windows, I was able to purchase and download SpinRite immediately from you, though sitting on the edge of my seat, hoping my Windows partition would not crash, as well. With some tinkering, I got SpinRite going. Once it had finished, my Mac would boot up like nothing had happened, working faster and smoother than before. I was able to back up all my files. And just to be safe, I bought and installed a new SSD. So big thanks for saving my files. Best regards, Andreas."

Leo: Yay.

Steve: And thank you, Andreas, for sharing the story. And for what it's worth, the copy of SpinRite you own will keep your SSD running well, too. Just run it on Level 2 from now on.

Leo: Yeah. That was a great tip you gave us. And since so many people are moving to SSD, I think it's a good thing.

Steve: Yeah.

Leo: All right. I've got questions for Mr. Gibson. Are you ready?

Steve: You bet.

Leo: All right. Let me pull them up. Question numero uno. That's not it. Here we go. Les Ramsey in Dublin, Virginia...

Steve: Yup.

Leo: Didn't know there was a Dublin, Virginia. He wonders whether government authorities can compel CAs, Certificate Authorities, to relinquish certificates. This is relevant to that discussion of HTTPS. In fact, one of the things people said, as they're reading that Computerworld article, trying to come up with good reasons why HTTPS is a bad idea, that's one of the things people say: Well, it could be spoofed.

I listen to your podcast every week. It is my primary source of information and education regarding security. While listening to the discussions regarding iOS security, SSL/TLS, and encryption, I wondered: If the authorities can compel a

certificate authority to divulge customers' certificates, does that thwart security and leave us with only end-to-end encryption as a last resort to privacy? Thank you for SpinRite. I've been a customer since v5 was first available. It's saved my bacon numerous times.

Steve: So I saw this, and I thought, you know, this sort of wants me to do a reality check. Because I wonder if we're just not being foolish. Here we are, all getting ourselves worked up over issues of protocol; and how random are the random numbers; and oh, my lord, this extension in SSL can be used for that and so forth. At the same time, our browsers are trusting hundreds of certificate authorities whom we're assuming, and the browser publishers are assuming, are always acting in our best interests.

And Les's question, and I've seen it voiced in other ways and contexts, but I wanted to note it because how can we imagine, if the NSA has the intent that they've been shown now to have, that they don't have a certificate authority in their pocket, that they're not a certificate authority themselves who are operating a front as a reliable authority, or that they don't have someone planted inside a certificate authority, or that in fact it's not just as easy as having a judge issue an order to induce a certificate authority to produce a certificate for them.

I mean, I guess my point is, as we know and we talked about in the iOS context a lot, this notion that the weakest link in the chain is the one that tends to get exploited. And the certificate authority system relies on the trustworthiness of not just one organization, but all the ones that our browser trusts, including famously the Hong Kong Post Office.

Leo: But wait a minute. Wait a minute.

Steve: You see what I mean? You see what I mean? It's just...

Leo: Yes, yes. But, wait a minute. Help me with this. So I am the NSA. You would use this for a man-in-the-middle attack; right? I would like to be in between Leo and Gmail.

Steve: Correct.

Leo: I couldn't do it as the Hong Kong Post Office. I'd have to get a certificate that said I was Google from the same certificate authority, wouldn't I? Or no.

Steve: Well...

Leo: And I'd have to convince Leo's browser. I mean, how would you use this as an attack?

Steve: Chrome, to Google's credit, Chrome is going a long way to mitigate this because, for example, Chrome cannot have Google certificates spoofed because Chrome knows

Google certificates through the process known as "certificate pinning," where the browser itself knows the serial number of the certificate. As far as we know, there is no way for anyone to create an identical fraudulent certificate. They can create a good certificate, but it will not be identical. So Google, using Chrome and Google is an instance where, due to the extra measures Google has taken, their links are extra strong. But imagine that you were the NSA, and you wanted to intercept your use of Facebook.

So all the NSA has to do is induce any one, I'm not even sure if it's not thousands, I think it's thousands of certificate authorities whom your browser trusts to sign certificates, to issue them a certificate for Facebook.com. So at that point the NSA has a certificate signed by an authority whom your browser trusts. So that when they do a man-in-the-middle attack, that is, when they intercept the connection, they see that you're trying to connect to Facebook.com. So they respond with their certificate, which your browser, even Chrome - because Chrome doesn't have special knowledge about all domains or certificates, only about Google's, where they're pinned. So they respond with a certificate signed by an authority whom your browser trusts, one of any of the thousands of authorities your browser trusts. And no alarms are raised. Nothing strange happens. Maybe it's an EV certificate, and you even get an extra green bar in the browser.

So that's all it means. It means that, while it isn't trivial for mass eavesdropping, it's just - it's impossible to imagine that the NSA isn't just laughing at us all worried about this, the idea that they can't eavesdrop on anyone's connection that they choose to. It strains credibility, or credulity, that they wouldn't be able to mint any valid certificate that they chose to, given the fact that our browsers are trusting all of the certificate authorities that are issuing valid certificates on the Internet.

Leo: Yeah, I don't think you have to worry about the Hong Kong Post Office. I'd worry much more about a CA operating in the United States who would have to respond to a national security letter silently and thoroughly. So, I mean...

Steve: Yes. And be gagged, yeah, be gagged by it.

Leo: So I think it's not unreasonable to assume that the NSA would do that. Now, you're right. It's not good for mass surveillance. You would have to say, hey, I really want to read Steve's email, to do it. And in this case you'd have to get a national security letter. You'd have to go to the ACA. They probably have already got a CA.

Steve: They must have a CA in their pocket.

Leo: In their pocket, yeah.

Steve: Yes. One of these thousands that our browser trusts is not actually, I mean, they're in the CA business because they had to prove and look like a CA and act like a CA...

Leo: Right. That's easy.

Steve: ...in order to get, yeah, exactly, it is because the browsers don't want to unfairly deny a little startup CA from having an opportunity to be in business.

Leo: For all we know, the Hong Kong Post Office is actually in Muncie, Indiana and run by the NSA. That's for all we, you know...

Steve: Actually, no. I know it's not because we've had a listener who sent us photos, Leo.

Leo: Of the Hong Kong Post Office?

Steve: They were standing in front. And he said, "Steve, you're not going to believe it. Here I am, I find myself standing in front of the Hong Kong Post Office."

Leo: I probably trust them more than I do others.

Steve: I wonder if I can go get a certificate.

Leo: I don't know if I trust VeriSign. But you can also get very paranoid about all this. I think that this is the risk of this.

Steve: Well, yes. And so it's important to note that they could not afford to do it on a wholesale basis because, in fact, the more they did it, the greater chance is that somebody would spot it. For example, Google, thanks to Chrome pinning, Chrome certificate pinning, Google has been the one who has spotted when their certs have been spoofed because Chrome sends an immediate message back to the mothership that says, whoa, I just got - someone just connected to me with a bogus Google certificate. And so Google instantly knows that. So because it would be a completely valid certificate from some domain that they're wanting to intercept the traffic from, but it's not going to be the identical certificate.

And that's the whole - my whole fingerprinting, certificate fingerprinting service that I put up on GRC last year, that's what that's all about is the serial numbers cannot be duplicated. So if you're seeing a certificate that doesn't match the one that GRC sees, then there's a reason to worry. But you're right. It's not wholesale, but it's targeted. And we just have to assume that anybody they want to target, they're able to mint certs on the fly, intercept their connections, and decrypt all their communications.

Leo: It just means something we've always known: Don't use the Internet for something you want to keep private. But that's always been the case.

Steve: Yeah. And in fact that's a very good point. We have a question a little bit later about Turkey's attempts to block. And what you'll hear me saying shortly is that the Internet wasn't meant for that. I mean, we are extending it in all kinds of ways beyond what it was meant to do. And it doesn't do very well the things it wasn't meant to do.

Leo: When bad guys wanted to talk to one another, they actually met in person. That was usually how they did it. Then they used the phone, and they realized that's not safe. Brian Weeden tweets, via Twitter: "The White House is looking to replace Obama's BlackBerry with 'secure' Android. But why not iPhone?"

Steve: I thought that was interesting. He links to an article which does show, I mean, famously Obama...

Leo: I don't think he was using a BlackBerry. I beg to differ. I think he was using a very secured Windows 6.5 phone, last time I saw anything. He wanted a BlackBerry.

Steve: I thought he always had a BlackBerry.

Leo: He'd always had a Blackberry, and they took it away from him.

Steve: Ah, he wanted to keep it.

Leo: Well, the last I saw they took it away from him, and they have a secure - I'm looking at this, and I think he's still using it, based on the photo I'm seeing here in this article, a secure Windows 6.5 phone. They say that's a BlackBerry. Maybe it is, but I think that's a misapprehension.

Steve: So I sent three tweets back in response to Brian's note, just to answer him. I said, first of all, I said, "Apple's approach is perfect for the typical consumer, who doesn't want the responsibility for security." That is, basically we've offloaded that to Apple. Apple's going to curate the iTunes store. They're going to do everything they can to keep bad guys out of their ecosystem and out of our phone. It's not our problem. Perfect for the typical consumer.

I said, second tweet, "But Android hardware, which is also lovely, can also fully accept fully vetted software that's fully known, which iOS cannot offer." And my third tweet was, "Or stated differently, there's no way to remove Apple from the iPhone. That's not okay for the President of the United States." And so, to me, an Android phone is absolutely the right solution for that scenario, where presumably the NSA or some group that really understands security is providing that operating system, soup to nuts, in that phone, and can absolutely vouch for how it's operating and what it does. So, yeah, I'm sure Brian tweeted this in the wake of our three episodes of coverage of iOS security. But it's not to be confused with, while the security in iOS is great, it is utterly Apple-centric. And they need something way more secure than that.

Leo: I'm looking. It is a BlackBerry. I'm looking at - there's a ZDNet photo gallery of all the devices the President uses.

Steve: And when you said Windows 6.5...

Leo: Well, see, this was what I remembered when he took office, and that's probably one of the problems here.

Steve: Oh, I absolutely remember he...

Leo: He had a BlackBerry, and they took it away from him.

Steve: Yes, yes.

Leo: And the phone they gave him was one that - this is a military, a division of the military that secures communications in the White House. And they have a Tempest-level security smartphone. But of course it's not a late model because it takes them a while to do all this. And as I remember at the time it was Windows 6.5. That I don't - the black one is a BlackBerry.

Steve: You know, Leo, I've got some Trios in the fridge...

Leo: The President might want. What we don't know is what's - the truth is, I wouldn't let the President use a BlackBerry if he's using BlackBerry servers because then his communications are going into Canada. And I don't care how vaunted the security is, that's not what you want.

Steve: I don't think there's any chance that that was ever going to be allowed to happen.

Leo: I doubt very much that, even if he's using a BlackBerry, that he's using anything..

Steve: Going into Canada.

Leo: ...like that. So I don't - I just - I don't know. I'm looking at the phone, and it's hard to tell. And I think there's the general presumption that he uses a BlackBerry because much of the government does. I don't know if that's a fair assumption.

Steve: Yeah, I do miss mine. But I love all the other toys. I mean, gee, I can't get 2048 with a BlackBerry.

Leo: Looks like he uses a...

Steve: It's hard not to get it for the iPhone.

Leo: Yeah, you need 2048. Looks like he uses a Nortel phone, though. That's the good news. Or is that Cisco? No, that's Cisco. That's a Cisco phone on his desk there in the Oval Office, it looks to me.

Steve: Yeah, you're right, it is. And in fact there was a photo that went out. Kerry and he and one of his national security advisors, I don't remember whom, were all sitting around, it was during the Putin call, I think. So it was like, oh, look at that, Cisco phones.

Leo: Yeah. But again, nothing that they're using is going to be anything like the stuff we use.

Steve: No.

Leo: I would hope. I would really hope.

Steve: At least there's not still a red phone on the desk, is there, the hotline to the Kremlin. I don't think so.

Leo: He probably - they use ham radio. Question No. 3 comes from Paul Cutts, also via Twitter, @pcutts. He says: If Apple's Keychain crypto is no good for iCloud, then surely the whole phone is insecure.

Steve: We discussed at the end of Chapter 3 of iOS Security last week that the really suspicious thing was that, from what Apple wrote, the architecture seemed to be one that would allow, if this P-256 elliptic curve, which it appears Apple uses nowhere but for securing the Keychain in the iCloud, everywhere else they used the good 25519 elliptic curve that everyone is using and that they use everywhere, that that particular use would allow the Keychain to be read but not modified. And so we ran out of time.

So I wanted to take this opportunity to say, for anyone who is concerned by that, Apple does provide in the iOS settings very granular control over what iCloud is used for. And you can turn off your Keychain syncing through iCloud, and only the Keychain syncing through iCloud, and it'll stop. Presumably it will wipe what's there. But I would then change my passwords that were synced on the Keychain after I turn off iCloud syncing, again, for those who, from hearing what we know, have reason to think that it's worth doing that. So it's certain possible for everything else to remain secure, yet this one thing that we know about that seems a little suspicious, eh, just choose not to use that.

Leo: Tweet No. 4 comes from David Peterson, @dpeters11. Regarding EMET on XP, it does help some - what was EMET? It was the Microsoft...

Steve: That's the Enhanced Mitigation Experience Toolkit.

Leo: Right. But ASLR, that is Address Space Location Randomization, isn't available. Oh, yeah, because XP doesn't support it.

Steve: Correct.

Leo: Yeah. Chrome itself works, but I couldn't get Flash working.

Steve: So there are a couple points here. First of all, he was right. DEP, which is the Data Execution Prevention, DEP was introduced in XP and was not used except by Microsoft's own code. And then there was a setting somewhere you could flip to turn it on for all your apps and then find out what broke as a consequence.

Leo: Right.

Steve: And you were also able to turn it on or maybe turn it off selectively. Anyway, there were controls for DEP. ASLR we first talked about in the context of Vista, which was where that was introduced. So David's right. You would be - EMET won't help you manage ASLR on XP, only DEP, but it will do that. And when he said Chrome itself works, but he couldn't get Flash working, I heard a lot of feedback from people who either had extensive experience already with EMET, or were experimenting with it after we talked about it some more. And it is exactly, I think, the way I characterized it, which is to say it's an expert's tool. It is not something for everyone. I mean, it makes NoScript look like a walk in the park. And some people feel that even NoScript is asking too much from people, to say, oh, look, the site seems broken. Oh, I'll turn scripting on. Oh, now it works.

EMET is like that on steroids. And so it's very powerful. But with that power comes responsibility, which is why it's not built in. It's why it's not there all the time. Microsoft says most people just don't want this much. But for our listeners, especially those who want to crank things up, if they plan to continue using XP past its end of patch point, I think it's a valuable tool.

Leo: Here's a longer one.

Steve: I would say plan to spend some time with it.

Leo: Yes. And of course the people we're worried about with XP are not using EMET. A paranoid listener, somewhere in the beautiful American wilderness, poses a question about security certificates: Steve, I was listening to Episode 443, "Sisyphus." I use a small non-profit Italian email provider - there you go - which gives me POP and SMTP over SSL. In order for their service to work, I had to download and install a certificate they provide. Wow, that's interesting. Yeah. I've heard of such things.

So my question is, why should I trust DigiCert, VeriSign, or the Hong Kong Post

Office more than this provider? It seems like a fair bet that, if I download the certificate from their webpage, where instructions on what to do with it also reside, it belongs to them. I don't care who "they" are beyond "my email provider," which is verified by the fact that in fact my email works with this cert. Is there some insecurity issue I'm missing? Love the show; love your software. I wish every developer employed the painstaking diligence that you do to write such beautiful and efficient code.

Steve: Okay. So this is an interesting question because he's saying there's a specific site that he wants to be able to establish SSL connections with. They apparently, for whatever reason, probably cost, choose not to purchase a certificate from a certificate authority that his system already knows and trusts. Instead, they have presumably a self-signed certificate. So they've just signed it, and they've said, here's our certificate.

Leo: That's kind of the equivalent of giving you a password and having you log in by a password; right?

Steve: It is. And so he notes that they're making it available on their web page. I don't know if their web page is HTTPS. That would be interesting because, if they're really so unwilling to spend money on security, is their site even secure? Of course, that would raise flags because it would mean that there was more opportunity for shenanigans. But let's assume that it is that their site is HTTPS, but their email system isn't. So they're able to securely deliver to the user a certificate for them which the user then trusts.

Leo: Okay.

Steve: On its face, I don't see a big problem with that. The reason, though, that that's not the way the world works, is it doesn't scale. And that's the beauty of the certificate authority model where our browsers pretrust someone, and then those someones, the certificate authorities, verify the identity of people who want to purchase certificates from them. So the beauty of that is it scales. That is to say, with a foundation of trust in this block of certificate authorities, we don't have to download and install individual certificates for every website we visited. Clearly, if we did, HTTPS would have a much harder time happening than it's already had. I mean, it's been available forever, and still it's not predominant, although it's certainly becoming that rapidly as we've been talking about. But the idea is there's more responsibility on the part of the user.

As I mentioned, there's the danger of the channel through which you obtain the certificate being insecure. So there's that danger. But fundamentally, if you got certificates from individual places you visited, all other things being equal, you'd have a big pile of certificates. And when you attempted to connect, your browser would verify the identity through that certificate, and you'd connect. So mostly I think the problem is that it doesn't scale. And again, it's worrisome that you're using sort of the same channel to obtain the certificate that you are then using to trust and so forth. So I would say in this instance you're probably okay. And the reason it's not done more pervasively, again, is that it would be too big a pain for everyone. It's better to have certificates pretrusted. Although, as we've just been talking about, with that comes its own set of problems.

Leo: Pretrust is like precrime. Question 6, an anonymous listener from the SQRL feedback page. How does SQRL allow for identity sharing? For example, if a family uses one central Amazon/Netflix/Hulu account for all their media access, how can we share SQRL tokens? We all want access to the same account, but SQRL is specifically designed to be unique.

Steve: Yeah. That's a very good point. The way this is handled is we have assumed in the design that you might have a centrally used machine that the family shares. Might be the family-shared computer. And normally people are individually authenticating with their usernames and passwords as they go places. And probably in a shared environment they're a little more schooled about not staying logged in and about explicitly logging off when they're done using a site. Otherwise somebody else comes along and is able to use it. But switching among accounts, login accounts, that will prevent a single account session from persisting. What SQRL does - and here's where we wish we had...

[Trash truck sounds in background]

Leo: Is take the garbage.

Steve: What SQRL does is allow you to define as many identities as you want. So Mom and Dad and brother and sister can each have their own named identity, and there can be one called Family. And so all they have to do, anytime that you're about to authenticate, SQRL pops up a dialogue where you verify. And prominent in the center of that is the identity which is, if there are more than one, which is currently selected. And there's a link right there to change identity. So nobody will know anybody else's password, so you don't have to worry about forgetting to change the identity. But if you were logging into Amazon/Netflix/Hulu, any of these shared ones, and it was set to your identity in your own session, you could just click on the little link in the UI, change it to Family, and then use the commonly known family password to say, yes, I'm a member of this family, and then you're logged in. So we've got that covered.

Leo: Yay. No. 7 already? Wow. John Clayton via Twitter asks what - you talk about Threema a lot, which is a secure public-key crypto-based text messaging service. Under "What Is a Threema ID" in their FAQ, they reference submitting a public key to a directory. How does that differ from iMessage?

Steve: I thought that was a good question because of course I was complaining about iMessage and the fact that Apple manages the public key directory, and therein lies a potential problem. I should mention that I saw some people as I was going through feedback who were saying, hey, Steve, you're complaining that Apple is doing this in the way they are. But you didn't complain about all the other facts that were in evidence from the document, and all they have to do is change any of that, if they want to. And that's absolutely true.

For me, the distinction is that, without changing the architecture of iMessage at all, Apple - and this was the point that I made - could operationally add a public key and receive messages that they're able to decrypt. So I do think that is a distinction with a difference in terms of the technology that exists in the phone, in the firmware, and of course in the hardware. And in here, this is just the function of the service.

So to answer John's question, what I like about what Threema is doing is that it is inherently transparent. And that's the problem with iMessage. We don't see the public keys of the people that we're messaging. We don't even see how many public keys there are. So there could be an additional one, and we'd have no idea that that was going on. With Threema, we're responsible for managing the public keys of the people we trust. That's a barrier that Apple removes at the cost of security, which Threema makes explicit. And I love it that they make it explicit because that is the sole point of security. Remember that Threema has three dots that are either, what are they, three reds, maybe a yellow, and then - or one red, two oranges, and three greens I think it is.

Leo: Yeah, because a three is good. More is good.

Steve: Yeah. And the idea is that the only way you can get three green dots is if you present your phones to each other, and the phones snap each other's QR code format public key. So there you've had a mating of the devices, and you absolutely know that the public key you've received belongs to the owner. The intermediate stage uses a directory where you're assuming that the directory is good. And as I recall, there are hashes that are shown. So you can get the key. And again, they have exposed everything. So you can then say to your friend, hey, I think I've got your public key. Is this the hash? And your friend says, yep, that's the thumbprint or fingerprint of my public key. So there you're able to, out of band, verify that you have the public key for the person you believe you do.

So that's the difference. It's fundamentally the same technology. But by burying the management of authentication, Apple has made it easy at the cost of some security. Threema absolutely puts it out in the public. And it's why I just think that's, you know, if I really want, if I absolutely care about having secure messaging, and I don't because I have no real need for it, but if I did, I would use Threema, and I would arrange out-of-band exchange of public keys.

Leo: I have a little - the key management is always an issue with public and private key crypto because, for instance, PGP keys are kept on key servers. MIT runs one. And it's propagated all the key servers. And I have made many keys since 1999, when I made my first key. And you're supposed to, or I guess you should, create a revocation capability so that you can, if you want a new key, revoke the old one. But I didn't do that. So I probably have 10 or 20, I don't know, some unknown number of keys still on those servers.

Steve: All still valid, do you think?

Leo: Yeah.

Steve: Do they not expire?

Leo: Well, I never, see, okay. Bad key hygiene. You should set them to expire, and you should have a revocation password or certificate so that you can pull them back. I never did, and I don't, and I have no idea what the passwords are.

Steve: Well, and you're like me. You were into it because it was available, and it was cool. You were playing with it. But you weren't really absolutely depending upon...

Leo: Because no one uses it.

Steve: Right.

Leo: I love it. So what I do is I make - every time I forget the password or whatever, I make a new key. So I have a current key. It's about a year old. But I get email all the time on old keys, and all I can do is send them the new key and say, look, this is my current public key. And, oh, there's another flaw, which is that anybody could create a key with my email address. So I know there are phony keys up there, too. They're no good because I don't have the private key for them. So people do that to mess with you. I don't think there's any security issue involved unless somebody starts - I don't know. I don't know how it would work. I can't think of a scenario where it would be bad. They'd have to get my email and everything.

Steve: Yeah, I've heard you talk - I was going to say, I heard you talking on Sunday about fake Twitter followers.

Leo: Oh, yeah.

Steve: That you think - and so what is that about? Is that like bots create accounts in order to monitor your feed?

Leo: No. It's a very tenuous way, but people do it. More than, I would say more than half of all Twitter accounts are spurious. And they're created often by spammers. In fact, I get a ton of spam. They'll create the account, and they'll put your name in it and hope that you will follow it. I don't know if they hope you'll retweet it. I don't know what the plan is. They direct message you if you follow them back, so I don't. I mean, it's not the best form of spam, but these guys, they've got plenty of time.

Steve: Yeah, and it's free.

Leo: And nothing better to do. So there are certainly inactive accounts. There's probably a lot of those. But a good many of them are spammers. And I see it all the time. I get a ton of Twitter spam. So, but it's nothing to worry about. But it is germane if you want to invest in Twitter.

Steve: Right.

Leo: Actually, speaking of Twitter, this is a good one.

Steve: Yes.

Leo: From e_StrategyPro. Of course Turkey famously has tried to block Twitter. And they did it because they changed - I guess everybody in Turkey uses the same Internet service provider, or it's a government-run Internet service provider. They modified the DNS so that you can't get to Twitter. And you'll see it in graffiti on the sides of buildings, 8.8.8.8, Google's DNS server. So people just fixed it. They said, well, we won't use the government DNS server. We'll use Google. Well, now, he says, Turkey reportedly intercepted Google's #DNS by redirecting 8.8.8.8 to their own DNS. Can direct IP address connections be spoofed?

Steve: So, yes.

Leo: It kind of takes government-level capabilities.

Steve: Yes. Yeah, actually it takes anybody who can interpose themselves in the traffic stream. And certainly a governmental body has the ability to impose whatever restrictions they choose to on the data traffic flowing across their border, or into and out of the country. DNS is one of the oldest protocols. It was created back in the beginning. And it really hasn't changed much. We've talked about DNSSEC, DNS security, the idea that DNS records could be signed, cryptographically signed to prevent spoofing and to allow the recipient to verify them. OpenDNS has the DNS curve system where they essentially establish their own elliptic curve crypto between users of OpenDNS clients and the servers, for the same reason, to encrypt and protect the connection.

So there are things that have been done since. But 99.99999%, I mean, today DNS is not secure. It was designed by the guys that know what they're doing to be super lightweight. So it uses UDP, which is just a single packet. It doesn't use the whole three-packet TCP handshake, nor does it then bring up after that an additional secure connection, SSL, TLS. It simply sends, in the plain, a small little query in a single packet off to the DNS server, assuming that that's where it's going. And a response comes back containing the IP address and some additional information that they asked for. And it works, and it's good enough for everyone to rely on. Because it's not perfect, as I said, there have been efforts to improve its security.

And we really do depend upon the security of DNS. Largely it's secure because normally our DNS queries don't go very far. Most ISPs, for example, host and manage their own big-iron DNS servers. And all of their customers' DNS queries go just to the DNS server and back. That minimizes their transit bandwidth. We've talked about the benefits of caching transit in order to minimize cost to the ISP because these servers cache all of the DNS queries that they look up from elsewhere, and then all of the customers get it from the cache.

So it's very efficient. But it is trivial, and this is the key word, trivial for an entity like the Turkish government to decide, to see what's going on, to see 8.8.8.8 graffiti on the walls, and to instruct the ISP to essentially see a UDP packet coming from there and redirect to their own DNS server, which blocks whatever they want to, and then respond, as if they were 8.8.8.8, back to the system that asked the question in the first place. So unfortunately, nowhere currently is DNS security required unless you go to extreme measures, like with the DNSCurve system and a proprietary DNS server organization, like OpenDNS. Do I mean OpenDNS? Is it OpenDNS?

Leo: Yeah, OpenDNS. You mean...

Steve: Yeah, yeah. It just seemed like that was more of a project name than a company name.

Leo: Yeah, doesn't it. I didn't think of that.

Steve: OpenSSL, OpenVPN...

Leo: But they support, in fact, they were one of the first to support DNSSEC, and they do a good job. Any ISP could do this, in other words.

Steve: Yes. It's, well, it's got to - you need a hierarchy, very much the way we have a certificate hierarchy. But the root is - I think all the root servers are now signed. So we have the beginning of a hierarchy. It just - it's like why has it taken companies so long to go to HTTPS, even though it's been obvious they should for years? It's just inertia, and everyone's busy and has priorities and other things to do. So the bad news is the Internet, and this was a point I made earlier that I said I was going to talk about, was never designed as a means of exerting control. It was actually - I'm not sure that it was designed for the reverse. It's been adopted for the reverse as the great unifier and freer, and nobody owns it, all of that.

Actually, it's just beautiful, simple technology that is incredibly effective. And it's also very prone to abuse. And so if somebody inside Turkey is having a hard time getting to Twitter because the Turkish government just thinks, you know, they're anti-social media, apparently, and think that social media is the scourge, there's really not much anyone can do because, again, the 'Net was not designed to be controllable. But if somebody really wants to exert the control, all the buttons are there, allowing them to do it.

Leo: You've seen, I guess they do this every year, the key signing ceremony for DNS that they do every year, the seven...

Steve: Not DNS. It's the...

Leo: The root servers. I thought it was the root servers.

Steve: Yeah, I'm sorry, yeah, you're right, the DNS root servers, yeah.

Leo: Yeah. I mean, they have - it's actually a physical ceremony.

Steve: Yes.

Leo: There's a physical metal key. Each of the primary 14 key holders owns a traditional metal key to a safety deposit box, which contains a smartcard which activates a machine that creates a new master key.

Steve: Yeah.

Leo: Love this.

Steve: Yeah. It's very cool.

Leo: [Indiscernible] society.

Steve: I think we have two of them in the U.S. It's international. But I think I remember that we have two key holders in the U.S. I don't remember now what the distribution is.

Leo: Yeah. Let me see if I can - one of the key holders is a Russian. There are several nongovernmental organizations. Lynn Lipinski, PR for ICANN, here's a picture, signs the official register of the key ceremony. This is just - this is a great article which was in the Guardian last month. They do retina recognition and, I mean, it is a little Deus Opus-y. I mean, it is a little...

Steve: Yup, and look at those boxes. Ooh.

Leo: Yeah. I don't know. I'll find the information. If you want, the easiest way to do it is "DNS key ceremony" in the Guardian. It was February 28th. Highly recommend it. What a great story it is.

Steve: Yeah, neat.

Leo: Yeah. All right. Moving on. Sorry about that. I just remembered that. An anonymous listener asks about crypto libraries - can you be anonymous if you can't pronounce "anonymous"? - about crypto libraries on the Security Now! page. Steve, can you mention some cryptographic libraries? You once mentioned that good crypto is available for free. He's not sure if that means "free" as in beer or "free" as in freedom. It would be nice to get some pointers. Thanks for your great work. And you've got some notes in here with a number of crypto libraries.

Steve: Yeah, I do. And so I just thought I would - I did a little bit of work, I mean, this just - I pulled it out of the air because I have been living and breathing this stuff. Dan Bernstein, who is a famous cryptographer who's got that great domain, cr.yp.to, he set about designing a cryptographic library with the goal that it would be difficult to misuse. Many of the cryptographic libraries are easy not to - they're easy to, well, to misuse. I'm trying to use a better phrase. But, I mean, it's easy not to get the effect that you want,

even though you think you are, from the library because they're complicated. And because crypto is complicated, and it's easy to make mistakes. So he said, okay, what is it that people really want to do, and let's just facilitate that. So he created a library called - it's pronounced "salt," but it's NaCl is the way you spell...

Leo: I love it. What a geek.

Steve: So nacl.cr.yp.to. That's sort of the template for, like, for where everyone has gone. I call it a template because it only compiles to, I think, like Linux or one of the UNIXes. It is not written so that it runs on 32 and 64 and Little Endian and Big Endian and everything. What happened was the OpenDNS folks wanted to do their DNSCurve. So they started with Bernstein's library and completely rewrote it, fabulously. First of all, all open source and really cross-platform. And this is Umbrella Labs. And so I have a link to the original Dan Bernstein salt because you can just sort of get a feel for it there. But what you want is Sodium, which is based upon salt.

Leo: I love this.

Steve: And free and open source, hosted over on GitHub. It's very portable. It's very portable C with ports to OS X, all of the BSDs, Windows. It's got Ruby and Python bindings. Basically, well, and for example, it's what I'm using for the SQRL client because it's got the - it's using the right curves. It's using Bernstein's 25519 curve that everybody is using. Someone noted that there's a Wikipedia page for Curve25519, and SQRL is now on it. And I noted that I'm in really good company, too, because, like, all of the - like Threema, for example, is there, and a bunch of other companies that are saying, okay, what's the best way to do the strongest crypto today? And it's to use the proper elliptic curve. And Sodium is a beautiful expression of Bernstein's design of a library that is difficult to misuse.

Also, it's worth mentioning, Stanford University's JavaScript Crypto Library, SJCL, which you can just - if you just Google Stanford's JavaScript Crypto Library, you'll find it. I know that some people think that crypto and JavaScript is inherently oil and water. I would argue there are places for crypto running on the browser, on client-side. There are fun things you can do. And they've got a beautiful library in JavaScript.

Leo: Wasn't that - what was that crypt-based or crypto-based, that I/O that you and I were looking at, was a JavaScript implementation of OpenPGP?

Steve: Oh, you mean Keybase.

Leo: Keybase.

Steve: Keybase.io.

Leo: Yeah. And they admit, as we talked about, if you're doing it in the browser,

you've got to hope that the browser's not compromised, and nobody's injecting anything. But it just seemed like such a good idea.

Steve: Yeah, well, there are many things. I mean, the Password Haystacks, the Ultra High Entropy Pseudorandom Number Generator, I mean, I've done a bunch. And the Off The Grid project. That's crypto in the browser, taking advantage of cryptographic pseudorandom number generation and other stuff. So again, with an understanding of the domain, I think it makes sense. And I just didn't want to forget Peter Gutmann's, or is it Gutmann's, never really was sure how to pronounce his last name, he's been in crypto forever, and he's got a fabulous crypto library, just called CryptLib, which is also free and available and open source, that he's been maintaining for years. It's very mature. Although I don't think it offers elliptic curve stuff. So if you want to do public key crypto, the Sodium library that I mentioned from Umbrella Labs is the one. And it's what's embedded in SQRL.

Leo: And all of these are libraries that have glue to all the other languages; right?

Steve: Sodium definitely does. JavaScript is JavaScript. But the other ones, pretty much all languages are able to make C calls. So you're able to, like, invoke a C function through them.

Leo: Pouring rain here, by the way. I don't know if you...

Steve: Oh, is that what we're hearing? I was wondering.

Leo: Can you hear that? I thought you might be able to. It sounds like white noise; right?

Steve: Yeah. We had a tiny little bit at about 3:00 a.m. this morning, and then it's going to come back for a second round. Although we've had our share of earthquakes in the last week.

Leo: Yeah. Oh, I didn't ask you. Did you feel that?

Steve: Things fell over, yeah.

Leo: Really. Oh.

Steve: Yeah.

Leo: No damage, though.

Steve: No damage, no.

Leo: Moving along to Question 10, our second to last, our penultimate question. Robert Lowery in Kansas shares some info about Open Office. Steve, I've heard you mention Libre Office on SN a few times lately, which I have been using for a few years. Then on the last episode I heard you mention that Oracle had let Open Office die, which is why I, too, had switched to Libre Office. However, I discovered about eight months ago Oracle had donated Open Office to Apache in June 2011. I have been relying upon Open Office more and more, and it has rarely let me down when it comes to reading and editing MS Office documents. Libre Office was a fork of Open Office.

Steve: Right.

Leo: And for a while that fork was more active. But that happens a lot in open source. By the way, SpinRite has saved multiple drives for friends, family, and customers. So thanks for a great product.

Steve: So I just wanted to put that in. We had talked about Libre Office. Everyone who has tweeted and uses it says it's great. But it sounds to me like, if you're looking for an Office alternative, and it's also the case that Office 2003 stops being supported next Tuesday, as well, second Tuesday of the month. So if you're looking for an alternative, and you find that there's a document that Libre Office won't open, then you may find that Open Office will. So I wanted to mention that it's still around. I did know that Apache had picked it up. So maybe when I meant that they let it die, I meant it died for them.

Leo: That they let it die, yeah.

Steve: Yeah, exactly.

Leo: It does run on Java. Does that concern you?

Steve: What runs on Java?

Leo: Open Office and Libre.

Steve: Again, if you keep Java out of your browser, you're fine. For example, Crowdin.net, that we're using for the SQRL translations, they provide a Java-based command line system. Basically they have an API to their server. But this allows me to, in my build process, to literally pull the files in real-time from Crowdin so I'm always using the latest versions. Which is handy. And so, and in fact that's why, remember, I installed Java a while ago and was mistakenly impressed by the fact that they said, oh, look, we're not enabling this for your browsers. And I thought, yay. But then a listener said, no, that's just because you had it disabled before, and it remembered that it was disabled before, that they're not actually doing that.

So I have no problem with Java as a very mature, a very nice language. It just never should have been stuck on a browser and allowed to have the rest of the outside world talk to it. So as long as you disable it in browsers, and you can do that with Java itself, restrict it itself, and then also don't install the Java plugin in your browsers. But it's great as a desktop engine.

Leo: And the chatroom is pointing out that Libre Office, the team has made a concerted effort to get rid of Java, and only a small part of it now uses Java.

Steve: Ah, good.

Leo: It'll probably be eliminated. It's all being replaced, apparently.

Steve: Probably rewriting it in C.

Leo: C++, yeah.

Steve: Nice.

Leo: And I don't know what the status of whether Open Office has eliminated Java, too, or are they trying to. It just shows you how long it's been since I've used any of these because I of course remember having to download Java to use them.

Steve: Yeah.

Leo: I have no Java on my machine, and I run Libre Office very well, says Dendrite [ph].

Steve: You can understand why they did it, in the same way that I understand why the Crowdin folks...

Leo: It's cross-platform.

Steve: Yes, exactly. It's truly write once. And, boy, there's a huge incentive for doing that.

Leo: It may also be the case that, while you don't have to download Java, that there's a JRE embedded somewhere in the package.

Steve: Uh-huh. Yes, exactly.

Leo: It's not that big. Just because you didn't install Java doesn't mean it doesn't use Java.

Steve: Yes. Back when I was - I think it was the Eclipse IDE, I think it was the same way.

Leo: That's Java. Yeah, that's Java.

Steve: Yeah.

Leo: Aldo in London, U.K. has our last question. He's worried about full-disk encryption: I have a question about, interestingly enough, full-disk encryption. I'd like to enable disk encryption on my machine - it's a MacBook Pro - but I'm hesitant for two reasons. Firstly, because of the hit to system performance due to increased processing; second, because it may shorten the lifespan of my hard drive - I'm using an SSD - due to the increased number of operations on the data. Could you comment on the effects of disk encryption on system performance and SSD lifespan, as well as whether the pros outweigh the cons?

Steve: I would say do not hesitate. None of those concerns are significant enough to offset the balance of security, even if you're not that needful of security. First of all, although I never understood why, my benchmarks of TrueCrypt showed, and I don't know if you're talking about Mac full-disk encryption or TrueCrypt, but when I benchmarked it, it was faster running under TrueCrypt than it was...

Leo: Why would that be?

Steve: I remember when I talked about that, it's like, I don't know. Some coincidence. I mean, I did it multiple times because I didn't believe my results. I took it off, put it on, took it off, put it on, tried all kinds of different tests. Might have been just a coincidence of, like, the rotational latency and the interacting with the overhead. The point is that, if there is any overhead, it is now, in this day and age, absolutely insignificant.

Leo: It is, it is.

Steve: For one thing, Intel has added AES-specific instructions to the chip, which have been there now for quite a while, so that, if you use AES encryption, I mean, it flies like greased lightning through the Intel. Probably the encryption overhead is insignificant compared to the time you wait for the disk to spin around to get to the sector, so that you absolutely don't feel a difference at all.

And so I would say do some measurements of things. Time things. Do stuff before you encrypt and then encrypt. And I think you'll find you can't tell the difference. And as for an SSD, there actually is, again, absolutely minimal effect. In the case of TrueCrypt, which encrypts the entire drive, it makes one read and write pass through the drive. So

that's not anything that an SSD is going to mind. In fact, to the degree that it's a little bit like SpinRite, it's probably a little good for the SSD to read all of its sectors and then have them written. What TrueCrypt does is it doesn't care what's in the sector. It reads it, it encrypts it, and it puts it back. So it's essentially putting random noise into the sector. But it only writes each sector once, which any SSD is going to be tolerating tens of thousands of writes. So again, don't hesitate. I would say employ full-disk encryption.

Leo: Yeah. I use FileVault on the Macintosh. And I haven't noticed, on modern Macs, anyway, a speed hit. It does take a while to encrypt the first time, if you haven't encrypted it before.

Steve: Well, because look at the size of our drives.

Leo: That's a lot of data. And I particularly use it on SSDs because, as we know, it's not always possible to entirely erase an SSD.

Steve: Yes, exactly.

Leo: Some data leakage may happen. So on smartphones I always turn it on. And again, it takes a while to implement it. But once it's implemented, I don't notice any difference on any device. I think nowadays stuff is pretty fast.

Wow. Speaking of fast, we've come to the end of this edition of Security Now!. Thank you. Steve. Do we know what next week holds?

Steve: We don't. I've got a few things on my list. There is QUIC, Q-U-I-C, I mentioned a while ago. I actually talked about it earlier in this podcast. It's one of the Google initiatives to see about coming up with alternative protocols which are, as the name implies, if it's not SPDY, then it's QUIC. And really interesting things that they did. So we may have ourselves a propellerhead episode. Everyone, I know, people really enjoy the deep technical stuff when we plow into that. And so if nothing comes up that preempts it, we may talk about that because I did the research months ago, and we just got too busy with emergencies. And I want to get back to it because I remember thinking, oh, gosh, Google. I mean, and I even talked about it before.

I remember now saying that one of the neatest things about Google is they really are working to improve the Internet. I mean, it may in some sense be self-serving because, yes, their browser gets the benefit of these things, and they're inherently an Internet-based solution, so things like transaction time and turnaround time and latency really matter. It is frightening when you look at the studies that show how quickly users will give up on a page which is slow to load. I mean, it's amazing how quickly they'll say, ah, forget this, and hit back, and then choose the next link. So anything we can do to reduce the latency is a good thing. And QUIC is a very, I mean, they just pulled out all the stops. It's very clever.

Leo: Steve Gibson is at GRC.com. That's where you'll find 16Kb versions of this little bit of a show, that makes it even a littler bit of a show, as well as text transcriptions,

written by an actual human being, GRC.com. You'll also find SpinRite there. What's the current version of SpinRite?

Steve: 6.0. We are shipping, and of course our audience all knows that, as soon as I get SQRL launched, I'll be going back and resuming the work on 6.1. And my commitment to everyone is, just because I think I should, everyone who ever bought 6.0 gets 6.1 for free. We are thinking, though, that we're going to at that time kill off all upgrades from previous versions because it will have been 10 years that anybody who had any prior version in this last decade could have upgraded to 6.0. So I think, I mean, it just makes things easier for us not to have four people a year do something that complicates everything.

So I think when 6.1 occurs, everyone who has 6.0 will be able to upgrade for free. I know that. That will give awareness of the newer partition table format, the so-called GUID, or EFS format, the GPT, the GUID Partition Table, native operation on a Mac because we will no longer get fooled by the keyboard, which is USB on a Mac. I had that working some time ago. And direct access to the hardware, so huge performance improvement. And frankly, I'm hoping that it continues to recover data as well as the current one does. Not that I don't think it will, but the current one just performs miracles. And sometimes I'm scratching my head, thinking, wow, you know, how does it do that? Which of course all of our users are also thinking.

Leo: Well, we'll look for that with great interest. And of course, as always, it is free, as all upgrades are to SpinRite because he's a generous fellow, that Steve. Future question editions can be addressed by going to GRC.com/feedback and leaving your thought or question, or apparently tweeting Mr. Gibson, @SGgrc.

Steve: I do keep an eye on my feed.

Leo: Follows his Twitter feed with great interest.

Steve: Well, it's a great way, you know, throughout the week people are saying, hey, I just saw this, I just saw that. And these episodes have been built from that, largely. And I think it's really improved their quality.

Leo: Yeah. We also have a...

Steve: Certainly it's extended their length.

Leo: We also have versions of the show, audio and video, available at our site, TWiT.tv/sn. And the best thing, of course, would be to subscribe, and that way you get it each and every week, right after we do it. We do do it on Tuesdays at 1:00 p.m. Pacific, 4:00 p.m. Eastern time. That's 20:00 UTC, live on TWiT.tv. We like it if you watch live, too. I interact with the chatroom a little bit, have a little chat.

Steve: Yeah, they're great.

Leo: Thank you, Steve. We'll see you next time on Security Now!.

Steve: Thanks, Leo.