Listener Feedback #171

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-411.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-411-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We've got questions from our audience. We're going to answer those, talk a little bit about the math around NSA's 5ZB, also some more revelations on SSL security. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 411, recorded July 3rd, 2013: Your questions, Steve's answers, #171.

It's time for Security Now!, the show that protects you and your loved ones online, your privacy, and also gives you deep insight into how computers work, how technology works, how the Internet works, with this guy here. Yeah, he's the Explainer in Chief, Mr. Steve Gibson. Hello, Steven.

Steve Gibson: You know, Leo, I wondered whether maybe we'd gone a little, you know, there's the expression "jump the shark," or off, over the top or something last week. But I got a lot of tweets from people who said, and even mail in the mailbag that I found today when I was putting our show together, that they really liked the heavy-duty, wind up the propeller beanie episodes. And so we haven't, I mean, certainly we have a widely distributed demographic. And it must have been that some of what I was...

Leo: No. No, no, no. No, no. You can assume this is extremely narrow.

Steve: Well, within the extremely narrow demographic, there's a spread of people.

Leo: I guess that's true, yes.

Steve: It must be that there were people who were like, huh? What?

Leo: What is he talking about?

Steve: Yeah. But I got a nice tweet, for example, from a Jerome, who said, "Loved SN-410 on Intel architecture. For me, having to rewind is a sign of a good podcast."

Leo: That's right. He wants more. I would say that that's a safe bet. Our audience is smart. We don't - there's plenty of general interest and crappy technology programming. I've got to play for you at some point this CBS This Morning woman who did the screed against passwords, and said passwords - she literally said passwords were security theater.

Steve: Oh, goodness.

Leo: And gave us the worst way of making up passwords I ever heard of. And this is on CBS This Morning. So no, no, I think people who listen to this show are pretty serious. Pretty serious.

Steve: They're ready.

Leo: They're ready. I don't think you should ever feel like you have to hold - don't hold back.

Steve: You know, I never - we never dumb it down.

Leo: No.

Steve: It's the serious stuff. And as you said, it does keep our listeners engaged.

Leo: Our listeners are well above average.

Steve: Yeah. They are. They're great, as I can tell from the questions that they submit. We're going to go through actually 11 because some will be quickies. And there's a combo one, I think, between 4 and 5 that basically asked the same question. First one was rather elaborate, and the second guy summarized it much more succinctly. I thought, oh, well, these both have to be voiced.

Leo: [Laughing] Great. Great, great, great.

Steve: Just to show two different approaches.

Leo: Did you see the Mother Jones article today about Edward Snowden? Now, this goes back a little ways, too, because we talked quite a bit about your theories about PRISM, which in every respect, the more we learn, the more accurate I think your theories have been.

Steve: Actually, there was a phrase in a piece on CNET. Declan McCullagh, who we've...

Leo: He's been doing a great job, yeah.

Steve: Yes, whom we've quoted often. There's actually a paragraph here where he said: "Documents leaked by former NSA contractor Edward Snowden confirm that the NSA taps into fiber optic cables upstream..."

Leo: "Upstreaming," they call it, yup.

Steve: "...from Internet companies and vacuums up email and other data that flows past.

Leo: They have a name for it. They call it "upstreaming."

Steve: Upstreaming.

Leo: So there.

Steve: So anyway, so actually this is the first time I have seen what I've proposed as what was going on written somewhere. So, and I wasn't aware of any additional documents which further confirm it, but maybe...

Leo: Oh, no, they released two new slides on Sunday at the Washington Post.

Steve: Oh, okay. Ah.

Leo: And they had the upstreaming information.

Steve: Okay.

Leo: So the more we know, the most accurate you are. And then the other thing that's an interesting question is, how come Snowden knows all this? And I think Kevin Drum, who's writing in Mother Jones, nailed it. He thinks that the evidence, and even the verbiage that's used about him, shows that he was a hacker for the CIA and the NSA; that his job was to build a target list of vulnerabilities for cyberwarfare. And I think that that sounds about right.

Stuart Staniford wrote: "I speculate this is going to turn out that Snowden was an electronic intruder on government payroll. His last job was working at an NSA threat detection center, suggesting knowledge of computer security. He previously worked for the CIA, including overseas, suggesting a cyber offense role."

Steve: I guess I just wonder, that seems like kind of more of a big deal than we have been led to believe. But he was also stationed in Hawaii, which sort of seems like, okay, why was he out there and not, like, at the mothership?

Leo: Pacific intercepts? I don't know. I truly think that he probably - that they're downplaying his role. The federal government wants to downplay his role. It's in their interest. I think it's fascinating. Anyway...

Steve: I do think that the guy, I mean, I watched that first introductory video of him several times. And he's clearly no dummy. I mean, many of the talking heads and the politicians have gone out of their way to paint him as a high school dropout. And it's like, well, folks, Jobs, Wozniak, and Gates all dropped out of college.

Leo: The best hackers are, that's right.

Steve: So, yeah, okay.

Leo: Yeah, that's not a - no strike against him. So let's get into it.

Steve: We've got news.

Leo: We've got questions, but let's get into the news, yeah.

Steve: So actually it's this article I wanted to share with our listeners, about the first half of it. Declan's article that I quoted from has the headline that - it actually leads into next week's topic nicely, and many people have picked up on variations of this. And the title of his piece was "Facebook's outmoded web crypto opens door to NSA spying." And then the subtitle was: "It's relatively easy for the National Security Agency's spooks to break outdated web encryption after vacuuming up data from fiber taps, cryptographers say. But Facebook is still using it."

Okay, now, it's absolutely unfair to point at Facebook because most people are still using it. So, first of all, that's the first - I guess Facebook being Facebook is a target. Maybe it

makes for a better headline. But the fact is very few people have yet moved their public key length up to 2048. Most, I think, are probably still back at 1024. And that's essentially the substance of this.

But there's some interesting details here. He says: "Secret documents describing the National Security Agency's surveillance apparatus have highlighted vulnerabilities in outdated web encryption...." Now, okay. It is certainly noteworthy, though, that the documents that Snowden released are highlighting these, meaning that this is certainly not falling on deaf ears back in Virginia.

So, he said, "...highlighting vulnerabilities in outdated web encryption used by Facebook and a handful of other U.S. companies." Okay, it's a very large handful, as I said. And then there's the paragraph that says, that I mentioned before: "Documents leaked by former NSA contractor Edward Snowden confirm that the NSA taps into fiber optic cables upstream from Internet companies and vacuums up email and other data that flows past - a security vulnerability that HTTPS web encryption is intended to guard against.

"But Facebook," and he says a few other companies, but I don't think that's accurate, "still rely on an encryption technique viewed as many years out of date, which cryptographers say...." Well, for example, all of Google's certificates are 1024, and they've got more than anybody else.

Leo: Oh, well.

Steve: "[F]ew other companies still rely on an encryption technique viewed as many years out of date, which cryptographers say the NSA could penetrate reasonably quickly after intercepting the communications." And this will be the topic of next week. We're going to delve into the - back into the security setup protocol of SSL/TLS in order to look at this question of what can you get knowing what from captured communications, and how does the so-called "perfect forward secrecy" help to prevent that.

So it says: "Facebook uses encryption keys with a length of only 1024 bits, while web companies including Apple, Microsoft, Twitter, Dropbox, and even MySpace have switched to exponentially more secure 2048-bit keys." And so, okay, there's a handful who have, but they've only done so recently because this is only recently - essentially, this is another side effect benefit of the fact that certificates are expiring constantly. Remember that they only have a one, two, or three-year life; EV certificates only a maximum of two years.

So I've grumbled about that in the past, saying, well, it's a pain in the butt, and it's expensive because basically you're just continually shelling out cash to the people who are signing your certificate to say yes, we've proven yet again that you are still you. The flipside is that this deliberate expiring of the certificates does allow for this kind of rolling upgrade. That is, if at a certain time processing power has increased to the point where a 2048-bit certificate is no longer considered burdensome to use in establishing a connection, well, then, let's get that. I mean, you could still ask for 1024. In some cases it's believed to help with compatibility.

But everyone now is using 2048. And, for example, GRC is. I was at 1024 until just the holidays, when I replaced all the servers and updated everything. So this is really just sort of happening. We're in the cusp where we're beginning to switch over.

And continuing from the article: "Eran Tromer, an assistant professor of computer

science at Tel Aviv University who wrote his 2007 dissertation on custom code-breaking hardware, said it's now 'feasible to build dedicated hardware devices that can break 1024-bit RSA keys at a cost of a million dollars per device.' Each such dedicated device would be able to break a 1024-bit key in one year, he said.

Leo: [Laughing]

Steve: Okay, so, yeah, exactly.

Leo: Okay.

Steve: Now, this is something that scales, though.

Leo: Right.

Steve: So $10 million...

Leo: So for a hundred million you could do it in three days.

Steve: Yeah, well, exactly. Exactly. Perfect. So realistically, quoting still: "'Realistically, right now, breaking 1024-bit RSA should be considered well within reach by leading nations, and marginally safe against other players,' Tromer said. 'This is unsatisfactory as the default security level of the Internet.'" And that's really true. It is time to give up 1024-bit keys, especially given the fact that we recognize security as a function of how much you want to break it, and unfortunately there's very little doubt anymore about what the interest is in breaking security.

Leo: Well, one could presume that this Utah data center would have one half billion dollar custom ASIC facility that would crack the stuff in less than a day.

Steve: This next line, yes, the next line in the article is "The NSA's budget is estimated to be at least $10 billion a year."

Leo: Yeah.

Steve: So they'll have to put up their walls and put up their power plant and buy their 5ZB of storage. But then they're going to say, okay, wait a minute, let's start building cracking boxes.

Leo: We need one of these.

Steve: And we have to assume...

Leo: It's on the list.

Steve: Yeah, we have to assume. And in fact I tweeted maybe three weeks ago, I made a tweet that was sort of prescient in this regard. It said something like "Capture everything. Brute-force decrypt selectively."

Leo: Yeah. And presume that, in time, you'll get the capability to break other stuff.

Steve: Yes, yes.

Leo: And that's why they're saving the PGP, Tor, and all that stuff because they figure, well, we can't crack it now. Now, if you go to 2048, that's more than doubling the amount of time. That's exponential; right?

Steve: Oh, whoa, whoa, Leo, it's exponentially more difficult, yes. It's like, okay, now we're in a whole new ball game. Continuing: "Facebook declined to comment on this article." And I don't, I mean, what could they say? Okay, well, we're not alone is what I would have said, but maybe just "no comment" is better. "A person familiar with the company's encryption development plans, however, said the social network is working on switching over to 2048-bit relatively soon.

"Encryption that's used to shield the privacy of web browsing is known as RSA, a form of public key cryptography based on the fact that it's immensely difficult to factor large numbers. As microprocessor speeds continue to advance, however, RSA keys with lengths that were previously viewed as secure have fallen to brute-force attacks." And that's really the meat of this article. It goes on to talk about previous cracking and lengths and so forth.

But, so essentially, I guess I would walk back the screaming headline a little bit and say that, not just Facebook, but Google and many other companies who are using 2048 - sorry, still using 1024-bit keys, only because two years ago or three years ago that seemed fine. Well, I imagine everybody will be moving now as their keys expire. And in fact I did see, I think it was Adam Langley quoted here. He's a neat guy. He and I corresponded, I mean, he's literally maintaining the source code of the security side of Chrome. And it was with Adam that I exchanged email when I wanted GRC to be built into Chrome's SSL-only list so that Chrome will refuse a non-SSL connection to GRC. I think it was here. Oh, yeah, here.

"Langley, the Google software engineer, said his employer could devote some of its massive computing" - oh, this is not what I wanted to read - "could devote some of its massive computing resources to breaking a 1024-bit RSA key, if it chose to do so." Okay, well, thanks for that. He says: "'It could be done today. We could do it if we really wanted.' But, he adds, there are better ways to spend millions of dollars in a way that will 'advance the state of the art of cryptography research.'"

And actually we'll be talking a little bit about that later. I'm sure in this article Langley was quoted as saying of Google that - oh, yeah, here it is. He said - oh, here. "Google

also uses 1024-bit keys, but in 2011 it implemented a clever trick called 'forward secrecy,' meaning a different key is used for each encrypted web session instead of a single master key that's used to encrypt billions of them." Okay, and that's not quite true because it requires cooperation. So we'll be talking about that.

"The company said last month it will switch over to 2048-bit keys by the end of this year." And we've already spoken previously about Google's early warning of their intention to do so, and that in that move of going to 2048-bit keys, remember that they said do not use certificate pinning, as it's called, where you're locking on to specific certificate serial numbers. Those are going to break. Do not assume we're going to be using the same hierarchy of signing. Do not assume we're going to be using the same certificate authority as our root, and on and on and on. I mean, basically, anyone who's not been playing by the rules to take shortcuts, get prepared to be unhappy because we're going to change everything. So we know that that's happening.

And now, says Adam: "'We would have preferred to move sooner; but, operating at the scale we do, client compatibility is always an issue. Everything on the planet seems to connect to us.' Langley added: 'We would have totally eaten the cost and the speed years ago if we could have done it without worries.' As an additional precaution, Langley said, Google usually rotates its RSA keys every two weeks." And it says in parens here: "(Facebook does it once a year, and is also planning to make forward secrecy a default for users, which few other companies do. Once Facebook switches to 2048-bit keys and forward secrecy, its users will be better protected against NSA surveillance than almost any other company.)" And that's of course only true until everybody else catches up. So, interesting little piece. And I thought that needed some clarification. And it will, it gives us a little built-in tease for next week's techie topic.

Leo: Good, good, good.

Steve: There was another one I thought was interesting, Dan Goodin, who we've also often covered, writing for Ars Technica. The headline was "How the U.S. (probably) spied on European allies' encrypted faxes." A bunch of our listeners picked up on this and made sure I was aware of it. And the subtitle here was "Grainy image stokes speculation of old-school, Tempest-style attack." And since this feeds into the concept of side-channel attacks, and we've talked about this sort of Tempest stuff before, I thought this was fun.

From the article, Dan writes: "U.S. intelligence services implanted bugging tools into cryptographic facsimile devices to" - oh, and by the way, the EU is really unhappy about this news - "into cryptographic facsimile devices to intercept secret communications sent or received by the European Union's Washington, D.C. outpost, according to the latest leak from former National Security Agency staffer Edward Snowden. Technical details are scarce, but security experts reading between the lines say the program probably relies on an old-school style of espionage that parses electric currents, acoustic vibrations, and other subtle types of energy to reveal the contents of encrypted communications."

Remember we've spoken of many things like this once. There was some question of the lights blinking on router switches, someone claiming you could read the traffic off of the blinking light. There have been experiments about, like, recognizing what people were typing just by listening to the sound of the keystrokes on the keyboard because there's something, keys are individually distinctive. And so once you - if you recorded enough of a single keyboard being typed on, then ran that through frequency analysis of the language in which it was being typed, you could discern which was the E, the T, the S and so forth, and then begin to figure out what they were doing. So it's all very fuzzy,

but arguably better than nothing, again, if you are really determined, and you have a lot of money and time to figure this out.

Anyway, continuing: "The bugging method was codenamed Dropmire, and it appears to rely on a device being 'implanted on the Cryptofax at the EU embassy in D.C.,' according to a 2007 document partially published Sunday by The Guardian. An image included in the document, presumably taken from a transmission traveling over a targeted device, shows highly distorted text that can just barely be read by the human eye as the letters 'EC,' followed by 'NCN.' The fax device was used to send cables between foreign affairs ministries and European capitals, according to Sunday's report."

And then later down in the article, this is the line that I loved: "Markus Kuhn, a computer scientist and senior lecturer at Cambridge University, wrote in a blog post published Monday," so two days ago, "'Having done many experiments to eavesdrop on office equipment myself, the noisy image at the bottom third of the sample picture looked instantly familiar.'" Oh, I just get goose bumps when I read that. It's like, oh, I mean, that's exactly what you would think. An independent academician has done this, too. And when he looks at what that is supposed to be in the lower portion, he says, yup, that's exactly what you would get from this technology.

And continuing Markus's quote, it says: "'It is what you might get from listening with a radio receiver on the compromising emanations of a video signal from a page of text.' Three security experts Ars spoke with agreed with Markus's analysis. They said it makes a strong case that the attacks targeting the EU encrypted fax devices were relying on what's known as side-channel attacks, which target weaknesses in a specific cryptographic implementation rather than the underlying cipher or mathematics it's built upon."

So this is another - it's a classic instance of the cryptography is not the weakness, the system is the weakness. And since the system involves at some point preencrypted plaintext, if it emits radiation or sound or light that in some way represents what you're sending prior, you know, like when it's being digitized, when it's being scanned, and if you can capture that - and there's going to be noise, there's going to be background noise, all kinds of interference. But if you can find the signal in the noise, that is, if the signal-to-noise ratio is such that you can extract it. And again, we're very good at doing that. We solve CAPTCHAs, god help us, that computers can't because we're good signal-to-noise discriminators. So anyway, really interesting piece, I thought.

Just in other little quick news, Ubisoft has lost their password database. I would imagine, I got a lot of tweets from people saying, hey, just got a letter from them...

Leo: Now, we should - I'm going to say this real quickly. This is not YubiKey, this is a game company called Ubisoft.

Steve: Good.

Leo: Because it's an audio podcast, people might...

Steve: Thank you.

Leo: And we talk a lot about YubiKey. That's not the Y-u-b-i, this is U-b-i soft.

Steve: Right, yes, good.

Leo: It's a game company.

Steve: Thank you, Leo.

Leo: Yeah, yeah, because Stina would be mad if we...

Steve: Yes. They posted: "Hello All" - and we don't want Stina to be mad. Now she lives nearby. "Hello All. We recently found" - this is Ubisoft. "We recently found that one of our websites was exploited to gain unauthorized access to some of our online systems. We instantly took steps to close off this access, to begin a thorough investigation with relevant authorities, internal and external security experts, and to start restoring the integrity of any compromised systems. During this process, we learned that data were illegally accessed from our account database, including usernames, email addresses and encrypted passwords. No personal payment information is stored with Ubisoft, meaning your debit/credit card information was safe from this intrusion.

"As a result, we are recommending you to change your password." And then in the blog posting, "Click here to change your password. Out of an abundance of caution, we recommend that you change your password on any other website or service where you use the same or a similar password." And I would argue that that's now become de rigueur. It's standard operating procedure, first of all, not to use the same password across multiple sites. But certainly, if you learn that on one site there's been a problem, then the argument is you really need to change it everywhere and take that opportunity to use a different password on different sites.

There was a slide that was actually tweeted to me. And Leo, you might want to bring that, click that link...

Leo: Sure.

Steve: ...and bring up the image because it's interesting. I asserted, back when I was talking about my theory for PRISM, that the problem was - the argument, for example, about why would tapping Google help, because all Google Gmail browser access is going to be over SSL. Well, unfortunately now we know that it's over 1024-bit SSL through the end of the year. But my point was that SMTP itself, the protocol, is almost never encrypted.

Well, it turns out that people have been looking into that since. And of the four major web-based mail providers - Google, Hotmail, Yahoo!, and AOL - only Google offers SMTP server-to-server encryption. So that makes my case. Essentially, if email goes anywhere, that is, to any - since encryption has to be supported at each end, then Google to Hotmail, Google to Yahoo!, Google to AOL, and any other combination of that, will always be decrypted.

Leo: But Gmail to Gmail would be safe.

Steve: Well, yes. There's an interesting question about General Petraeus's strategy in communicating...

Leo: He blew it. If he's just stayed within Gmail he'd still be the Chairman of the Joint Chiefs of Staff or whatever the hell he was, yeah.

Steve: Yeah. So Google to Google, if we assume that they transit on the public Internet over SMTP between data centers, I mean, we know very little about Google's internal infrastructure. But essentially, only if it was Google to some other server also supporting secure SMTP, would they negotiate security after establishing a connection. But, for example, no Hotmail, no Yahoo!, no AOL email would ever meet that level, and neither sending nor receiving to or from a Gmail user.

So today, unfortunately, there's still lots you can get from tapping a fiber optic link. Even though a lot of traffic is encrypted, once it emerges, it will be decrypted for you.

Leo: The good news is more and more people are using Gmail.

Steve: Yes. Yes, right. So Gmail to Gmail is probably going to be...

Leo: I'm sure that stays within the Googleverse.

Steve: I would think so. And a couple, just as a further sort of random follow-up to my talking about PRISM a couple weeks ago, a number of people said, well, wait a minute, you know, Facebook, you're always staying in Facebook. So when you go to Facebook, I mean, Facebook's brought up security. Assuming that you're browsing in Facebook and posting in Facebook, then all of the data is secure. Well, except anything coming out of Facebook mail-wise is not, or going in is not. But more importantly, and this is significant, I believe the slide that we saw with that weird-looking timeline, kind of a green arrow with yellow ovals on top of it, what I remember making a note in my mind was that Facebook was added in '07.

Well, the point was Facebook, as we know, only very recently, it was last year in 2012, got serious about HTTP security. They were only being secure during logon, thus the reason that Firesheep was such a problem for them, because their insecure token was being sent back and forth, and we were immediately seeing all these Facebook users pop up. If you fired up Firesheep at Starbucks, that's what you saw was Facebook profiles. So this is an interesting point because the FBI has referred to often the "going dark" problem, as they call it.

So what we have is a situation exactly like that, where a substantial amount of time and effort and money and trouble was invested in '07, if we're right about what PRISM is, in tapping Facebook and as establishing fiber optic taps into the Facebook data centers. And back then it was a treasure trove because all of the pages, except for login, were in the clear. Well, the NSA could not have been happy when Facebook did what was a benefit

for its users last year, which was to get serious about SSL and first offer the option, and then switch it to default, which of course is where we are now.

So it is unarguably the case that that tap that was good for five years has largely gone encrypted, which doesn't mean they're not still sucking stuff in, but it's not as easy as it was before. So my point being this is a moving target. And they got five years of data. Maybe it did them some good before Facebook said, well, I think maybe we should be encrypting.

Leo: I have to think that anybody who's doing seriously bad things at this point is not trusting Facebook to communicate with their cells in the United States.

Steve: Yes. Yes. Although...

Leo: Seriously. Come on.

Steve: ...from a metadata standpoint, all of this adds up for, like, networking. For example, you might have a network of bad guys talking about their favorite sandwich meats...

Leo: Right.

Steve: ...on Facebook, and then doing all of the - being sneaky when they talk about anything that involves bad stuff. But their network has still been established using metadata to link them together when they had their guard down. So it's still useful.

Leo: They never have their guard down. All you have to do is look at "Zero Dark Thirty." Osama bin Laden was living in a compound where there was no Internet access at all.

Steve: Right.

Leo: He knew. Why do you think they live in caves in Tora Bora? They know Internet access is a bad idea. They ain't using Facebook to - I firmly believe the government doesn't do this to capture terrorists. They do it because it's a great way to get even.

Steve: Leo, they do it because they can.

Leo: They can.

Steve: Yeah.

Leo: Useful. And I'm not convinced it's of any help in fighting the real bad guys.

Steve: Well, and the arguments that Udall has been making is that for years in the oversight committees they were pressing the intelligence services to demonstrate that they succeeded with the program, and they were unable to.

Leo: No, of course not. They do it because they can. You're exactly right.

Steve: So, yeah. So you hear these, you know, on the Sunday shows, oh, it's foiled countless plots. Well, wait a minute. Oh, and I love it, too, when they say, oh, but the number is classified.

Leo: We can't tell you.

Steve: What do you mean? Why can you not tell us what number, how many?

Leo: But it's a large, large...

Steve: Yeah, yeah, yeah.

Leo: I just read an article, a blog post by a law professor who estimates that we break three federal laws, everybody breaks three federal laws a day because the federal laws are so broadly written and stupidly written, particularly in electronic communications and so forth, that we - everybody. And so they're collecting all this. And if they ever should, heaven forfend, decide they don't like somebody, they've got plenty of ways to go after them.

Steve: Well, I just - when I heard the intelligence guy say, well, a great many plots have been foiled, but the number is classified, it's like, what? A number is classified? What, 34? 27? 42? How can a number be classified. Anyway...

Leo: No. Because the number is zero, is why.

Steve: Well, and the point is, because we don't have to tell you. We will not tell you anything we don't have to tell you. So it's like, oh, okay, fine. Now, okay. I don't get this. This seems like the strangest thing ever. So we'll try to figure out - I've got two topics under the category of Browser Watch. The first is Firefox v23, which, by the way, is the next biggie, slated for next month, August. This is the one I've been waiting for because, what was the phrase, "God willing and the creek don't rise," I think? They're going to be, Firefox will be disabling third-party cookies by default. But something else happened in the v23 development channel which really sparked off a firestorm. They have removed the checkbox to disable JavaScript, and they reenabled JavaScript if it was previously disabled.

Leo: That's cold [laughing]. That is cold.

Steve: Wow. So updating to v23, the next one, because I just got 22, and I restarted the browser, silently enables previously disabled JavaScript - no warnings, no pop-ups, no notices at all - and removes the ability to disable it from the UI. So this caused a big ruckus. And, I mean, it's like, what? What? Huh? And as you can imagine. So the argument seems to be that sophisticated users are sophisticated. So they will know what to do.

Then the counter argument, of course, is wait a minute, but if they turned it off, they would expect it to still be off. It's been silently reenabled. Then the response then is, well, about:config, and then you dredge around in about 20,000 configuration settings. You use the search because you have no other choice. And somewhere you will find JavaScript:enable, and you change that from a one to a zero. But of course that's not user-friendly because it's not in the UI.

So then the argument comes back, well, there are much better tools available - in other words, NoScript, and NoScript is referred to by name in this back-and-forth in the text of this discussion. So if people want to turn JavaScript on and off, that's what they should use.

So I quoted two paragraphs here from this discussion: "The ability to share your experience, including turning off JS, is offered in many different ways." I'm sorry, "the ability to shape." I think I said "share." "The ability to shape your experience, including turning off JavaScript, is offered in many different ways. Not everything needs to be in the primary browser UI. We did not actually remove a choice, just reduced the visibility" - like, to nothing - of that particular choice. That does not go against either of these principles in the manifesto." And then they said, again, quote, a little bit lower, "Note that, if we removed the preference from the UI, but left JavaScript disabled, this would make life really hard for non-expert users" - and I'll add parenthetically who were expert enough to turn it off, apparently - "that accidentally changed this..."

Leo: You know, if you were told, turn off Java, and you hit JavaScript, and then the web broke, I can see that's not an unusual circumstance. Some of this may be just to avoid tech support calls, you know.

Steve: Actually, that's exactly - there was an argument in there that the IT people will be happy because they get tech support calls when people turn off JavaScript, and then things break and don't work. So anyway, for what it's worth...

Leo: I think that's not unreasonable. We're smart enough to know, oh it's turned on. Or use NoScript.

Steve: And it is the case that, well, okay, first of all, you're not using IE. So you've made your first big step towards...

Leo: Right. But again, a lot of people do this stuff because they hear people tell

them, oh, you've got to use Firefox, and don't forget to disable Java. And then they disable JavaScript. The web is broken.

Steve: Good point, good point.

Leo: I think that that's not...

Steve: I mean, you have to argue, I mean, your argument is sound, and that is that there's a big ambiguity between Java and JavaScript.

Leo: Right, right.

Steve: So much so that people thought one is a scripting version of the other, which we've disabused everyone of.

Leo: Is Java turned off by default?

Steve: Versioning is definitely verified now. I don't remember whether it's off by default.

Leo: And Web8013 makes a very good point, that you shouldn't just turn something back on by default because that's not appropriate either.

Steve: I know. Give them a popup and say, hi, we noticed that in moving from 22 to 23 we are removing this from here. We moved it over there. And by the way, how do you want your new default to be?

Leo: Right. That would be fine.

Steve: Yes. No one would argue with that.

Leo: Although, again, confusing for Grandma or Grandpa. And that's the problem.

Steve: Good point. And now they're doing silent updating. So it's like, wait, whoa, whoa, whoa, what are you doing, updating my browser?

Leo: Huh? But to silently turn back JavaScript, turn JavaScript back on if someone had turned it off, it does seem like...

Steve: I think that's wrong.

Leo: That doesn't seem right, yeah. Okay, I just wanted to mention, somebody in the chatroom said that Doug Engelbart has passed away.

Steve: Oh, wow.

Leo: Yeah. 88 years old.

Steve: Oh, well, he had a good run.

Leo: It's a great run. The inventor of the computer mouse while he was at SRI. I remember interviewing him on The Screen Savers 10 years ago, and just a legend. So, yeah. Just thought I'd mention that.

Steve: Second browser update is, okay, I don't think this is going to turn out well, but it'll be interesting to see. Chrome is experimenting with a new protocol that Google has put together called "QUIC." We are to pronounce - it's an acronym which we are to pronounce "quick," of course, stands for Quick UDP Internet Connections. Quick UDP Internet Connections. And I'll just read briefly from Wikipedia, since there's already an entry there, and I have not gone into it for a deep dive, although I expect I will because that's what we do here.

"QUIC supports a set of multiplexed connections between" - connections, remember, that's an important word because UDP is a connectionless protocol. TCP is a connection-oriented protocol. So what essentially they're doing is TCP over UDP, which is - and therein lies a story.

"QUIC supports a set of multiplexed connections between two endpoints over User Datagram Protocol (UDP) and was designed to provide security protection equivalent to TLS/SSL, along with reduced connection and transport latency, and bandwidth estimation in each direction to avoid congestion. The protocol handles packet loss well. Besides packet-level forward error correction, QUIC aligns cryptographic block boundaries with packet boundaries, so the impact of packet loss is even lower." I mean, these are all really exciting things.

"QUIC also allows higher level application protocols, such as SPDY, to reduce or compress redundant data transmissions, such as headers. One of the motivations for developing QUIC was that in TCP the delay of a single packet induces head-of-line blocking for the entire set of SPDY streams; QUIC's better multiplexing support means that only one stream would pause. As improving TCP is a long-term goal for Google, QUIC aims to be nearly equivalent to an independent TCP connection, but with much reduced latency - goal is zero roundtrip time connectivity overhead - and better SPDY support. If QUIC features prove effective, they could migrate into a later version of TCP and TLS."

So, okay. So my misgivings are that UDP connections are so different from TCP connections. They're different with proxies, which won't pass UDP, but do pass TCP. They're different with firewalls, same goes. They're just - it's very different. Also there is so much technology which has been built into TCP over time, incrementally and carefully. Now, it's all open, public domain open source. So the guys who were wanting to implement the critical aspects, retransmission and queuing and rate management and

buffer, we've talked extensively about buffer bloat and so forth. They can take all the lessons learned.

And it is the case that the problem with TCP is remember that it guarantees in-order delivery. The exact sequence in which you put things into your end is guaranteed to come out the other end. Well, that means that, if a packet gets lost along the way, then the other packets that are in - that have already been sent in send-ahead, they will start being saved at the receiving end. And when the receiving end doesn't acknowledge the packet that was lost, the transmitter, the transmitting end, which has been saving up these packets it's sending until it gets the acknowledgment, it will resend the lost one.

Well, so that does stall the entire flow. And if you've got multiplexed connections over a single connection, TCP connection, they all get stalled. And also the idea of aligning encryption boundaries with packet boundaries is brilliant because you cannot do that in TCP because there's no notion of packets in TCP. In TCP, the application simply sends a stream out. There's no packet-level awareness. But that's not the way UDP operates. The application itself generates and sends individual UDP packets. So it can explicitly align the encryption boundaries with the packet boundaries.

So anyway, really interesting. And I guess I wonder a little bit about what the gain will be because remember that modern TCP there is some overhead, the whole - the triple handshake roundtrip and then some - it starts slow, and it ramps its speed up in order not to immediately over-congest the connection, so it sort of - it feel its way forward. All of that happens per connection. And a browser will normally look like an octopus, sending connections out all over the place if it's populating its page with ads, for example, from far and wide.

And boy, when I look at what NoScript, sometimes I'll go to a site, it's like, okay, I need to enable scripting here, and I look at the amazing list of sites that are being blocked, it's often like 25 or 30 different places that have - they're all wanting to get their scripting into this page. I just think, okay, it's nice, if nothing else, to know.

So anyway, it'll be interesting to see how this develops. Google will always err on the side of caution. So it may be that they will - their servers will support QUIC. Chrome will support QUIC, but fall back to TCP, obviously, if it can't establish a QUIC connection. And then they'll see, they'll do big studies and statistical analysis and figure out where there are problems and where it's working, and they'll move forward if they can. So certainly from a techie standpoint I'm intrigued by this. And if more comes of it, we will certainly understand it all on this podcast.

A little bit of errata, one piece: Last week, when everyone's propellers were wound up, I was talking about sector sizes and said that the sector count was five bits. Well, of course I know better than that. It's six. Somebody said, eh, Steve, you got your math wrong. It's like, oh, duh, of course. Five bits is 32. Six bits is 64. So just to correct the record.

Leo: Of course.

Steve: Now, Leo.

Leo: Yes.

Steve: I am 64% of the way through Stephen King's "Under the Dome" novel.

Leo: And you know that because you read it on the Kindle.

Steve: I know that because I look down. Yes, I read it on a Kindle. I actually read. I'm not having it read to me. I am reading it myself.

Leo: Wow. How ambitious.

Steve: And, boy. It's funny, my favorite quote was, "Hard as this novel is to put down, it's even more difficult to lift up." Not on a Kindle, of course.

Leo: No, yeah. I can only imagine.

Steve: But I guess in the pulp, in the wood pulp version, this thing I guess is monstrous. I am really loving it. And I have to say I tweeted, I guess it was yesterday or the day before, I read through a really chilling part. And Stephen King has a message here. I strongly doubt it's going to come through in the TV version. First of all, the TV edition, as always is the case, is so inferior to the novel. I mean, people are doing ridiculous things on TV. And it's sort of fun to see the characters that have been chosen for the characters in the novel. Essentially the only thing that is carrying over from Stephen King's novel is the concept of being under a dome.

Leo: Yeah, yeah.

Steve: And the roles of some of the characters very roughly, the names. I mean, but it's radically different. So, I mean, actually I'm going to stop watching the TV series.

Leo: Yes.

Steve: Because it's too confusing for me to say, wait a minute, Rose didn't do that. Oh, that's right, that was in the novel and not in the TV. So I'm going to - actually I think I'll be finished with the book by the time we speak next week because I hadn't started it when we spoke last week. But the only little takeaway, and the thing that prompted my tweet, was the observation that people give up freedom in time of fear.

Leo: Yeah.

Steve: And we gave up freedom after 9/11.

Leo: Yup.

Steve: Because fear was stoked, and present, also, but stoked. And those who had a reason to want more power obtained it during the window. And, I mean, Rahm Emanuel was quoted famously as saying, "Never let a good emergency go to waste" or something to that effect. I mean, use these things. And that's what politicians do. And so I feel very much like Stephen King has created a microcosm with authority, outside authority cut off. And then he watches what happens inside. And it is really interesting. I mean, I'm loving the book. I can't put it down.

So I don't know, for people who are interested, you know, people - I saw criticism, I don't know if it was of the book or the movie, that people were not acting in a reasonable way. Well, it's absolutely true in last Monday's episode. One of the sheriffs just went off the deep end. It's like, what? I mean, it was ridiculous. And, well, I can't say anymore without spoiling. But anyway, so that isn't in the book. I'm very pleased with the behavior of the people in the book. And in fact one of the things that Stephen King is known for is building really understandable people. And, Leo, this doesn't often happen, I'm encountering words I don't know. I ran across "commodious."

Leo: Yeah.

Steve: And I thought, what? He was in the backyard, and she had a - I think it was...

Leo: Commodious kimono?

Steve: No, a commodious backyard. It means roomy and comfortable.

Leo: Yes.

Steve: And it refers, like as in furniture, or like a commodious couch, or a building could be commodious. And then this morning was "comestibles."

Leo: Oh, yeah. Food.

Steve: And it's food.

Leo: Yeah.

Steve: And I thought, okay, wait a minute, maybe Stephen is reading through the entire dictionary, and he happened to be...

Leo: He's a literate guy.

Steve: He happened to be in the C's. I mean, "commodious" and "comestible," and those are the two words I have never seen before? It's like, okay, wait a minute. He happened

to be in the c-o-m section of the dictionary he was reading when he was writing this. He thought, oh, I'll just drop "commodious" in here, and "comestibles."

Leo: This is - you had never read any Stephen King before. Is that right?

Steve: No. No, I haven't.

Leo: I think you'd like his other stuff.

Steve: Yeah, well, and he's advertising on the show "Dr. Sleep" or something. Looks very creepy. Something about...

Leo: Yeah, I know that.

Steve: ...somebody's demons, they were with him when he was young, and he grew up, and so did they. It's like [voice shiver], okay. Okay. So Simon Zerafa, our frequent contributor, offers - he's been throwing out all kinds of attempts at NSA humor. And one really landed well. So I thought I would share that one. He said, so the same NSA guy, our NSA guy who has his favorite bar, he walks into his bar and orders a beer, says "I want a beer." The bartender asks, "Domestic or imported?" NSA guy says, "What's the difference?" So I liked that.

Leo: Oh, I get it.

Steve: Okay. Not as good as the first one, but...

Leo: No, I get it.

Steve: You should have seen the other ones. Anyway, no. So Kevin in New York, I'll share this briefly, says "SpinRite fixes an old 1GB flash drive." And actually there's a lesson here for our listeners that I wanted to share. He says, "In testing how well SpinRite can work on flash memory, I tried it on an old 1GB Sony MicroVault flash drive. Basically, the issue was corrupting data, e.g., placing any large file on the flash drive would cause the checksum to come back differently. It was no longer reliable. So I decided to run SpinRite on the flash drive using Level 4 since I didn't care if it killed the drive since it is slow, and" - I love this phrase, I just, like, roll over in my, not yet in my grave - "and 1GB is no longer very useful." Okay, what land do we live in now where 1GB would...

Leo: It comes in a cereal box, that's right, yeah.

Steve: ...no longer be useful. Anyway, so - and the point of Level 4, of course, is that it inverts the data, reads it back, inverts it again, writes it and reads it back to verify it,

doing like essentially a full refresh. And that's controversial on non-spinning mass storage because non-spinning mass storage is known to have a write-cycle life. Anyway, so he says, "Anyway, running it on Level 4 COMPLETED FIXED," in all caps, "the checksum issue, though SpinRite reported no trouble, and I can now copy a large ISO" - see, that's useful - "file to the flash drive..."

Leo: That's true. For a CD, not a DVD.

Steve: For a CD, yeah, "...with no problem. I still do not trust the flash drive for anything important" - that's wise - "but it was interesting to try it and see what it would do with the flash drive."

Now, I wanted to walk back a little bit the prohibition about SpinRite on solid-state storage at anything more than Level 1 or 2 because, okay, it's just two writes. It writes it upside down, then it writes it back right side up. So it's like, yes, you don't want to do it continuously. You don't want to put your swap file on a flash drive.

Leo: No, yeah.

Steve: Where the light's constantly blinking on your hard drive. But if you want to, like, do a better job of cleaning up a drive which is a little dodgy, and/or maybe if Level 1 or 2 doesn't fix the problem, it's not like Level 4 is going to just fry it, I mean, just because SpinRite's got superpowers or something. It's just all it's doing is writing it all upside down, then right-side up. All the ones become zeroes and zeroes ones, and then vice versa to put it all back. And that's what this drive needed. So that was cool.

I just thought I would give a little weekly update on the progress on the next version of SpinRite, since sales has not been killed. I really thank our listeners for that, especially because everybody who is buying it will all get 6.1 for free. Last podcast was of course about the weird anomalies in the Intel architecture that allowed large access, 32-bit access in real-mode, which is normally limited to 16 bits. That's all implemented. It's all working now.

I now have a completely mature, built-in memory manager, extended memory manager in SpinRite, which is able to go out and explore all of high memory. It maps out all of the regions because there's various weird little chunks that are taken out with the goal of finding for itself 32MB to establish a maximum size transfer buffer because all of our new drives are able to transfer 64K sectors in a single burst, so SpinRite will be able to do that also. It's also compatible with external memory managers. If you've already got an extended memory manager, it'll see that and use that instead.

And next on the list - and this is all just finished yesterday, we got everything wrapped up and tested. Next is it's also completely enumerating the PCI bus, finding all of the disk controllers of every type that there are. However, there are three types. And it will only be able to work in super high-performance mode with one of the three, that is, the so-called "native PCI mode." You could have your controllers in compatible IDE mode, which doesn't give access to all of the upper memory regions, or in AHCI mode, which is the Advanced Host Controller Interface. That's the next-generation, super-advanced one.

The good news is, from what we've seen so far with, like, one exception in a hundred people and many hundreds of systems this has been run on now, SpinRite, I'm

expecting, will be able to change the controller into the mode it wants. All of, like, for a long time any of the advanced ones have always been able to be set back to this native PCI mode. So that's the next round of work we'll be doing, and I'll have an update on that next week.

Leo: How exciting.

Steve: Yeah. Oh, and there have been some people who are really anxious for the Mac version and have been asking, like, for a pre-release version that runs on their Mac. Where I'm really going is to support UEFI booting, which would allow USB sticks, USB drives to boot by holding down the Option key at boot time. That's still a ways away. But the much easier solution is the way people have been using SpinRite with limited success because of the keyboard problem, and that's on a CD. You can hold the "C" key down, meaning "C" for CD, and the system will boot in a PC-compatible mode from the CD. And that I think we'll have pretty soon.

My goal is to get all of this new hardware-level, data transfer, large buffer stuff written and solid, I mean, absolutely solid. I certainly won't let anything out that would be a problem. And then I'm going to do - my plan is to do an interim release, not 6.1, but just a development-level release, but which works and is way faster and compatible and all that. And at that point it's already got the Mac keyboard stuff fixed. So you could burn it on - you could take the ISO and put it on a CD and run SpinRite on your Mac.

So I think we'll have - as long as you've got a CD drive, that's my point, is that, like some of the Macs now, my Air, for example, I had to buy a CD for it separately. So I know that older Macs, of course, do have CD drives, but some of the newer solid-state drive Macs don't. So not long. Maybe a few weeks, if all goes well.

Leo: Hey, I just wanted to make a little bit of a note that Twitter has - this is something our audience would be very interested in, I'm sure, and the chatroom reminded me. Twitter has announced that it's going to allow advertisers to target you, if you are a Twitter user, based on your activities off Twitter.

Steve: Ooh.

Leo: Third-party browsing. They'll also use email addresses to target the ads you see. You won't see more ads on Twitter, they say, but you may see better ones. Fortunately, Twitter has put in some opt-out boxes in your settings so you can implement Do Not Track.

Steve: Okay. So what this means is that they will - the advertisers will bring their knowledge of you from when you're not on Twitter to Twitter.

Leo: Right.

Steve: So that the ads you see convey that knowledge.

Leo: And that Twitter will be giving ad partners more information about you based on what it knows.

Steve: Oh, goodness.

Leo: So, I mean, this is no different than what Facebook does at this point. But it might be something people - our audience, given, I know, their strong feelings about Do Not Track...

Steve: Yeah, I also saw - did you see that thing about Bing and Windows 8 or Windows Blue or whatever?

Leo: Yeah. Yeah, it's going to put ads in the search results.

Steve: It's like, oh, my goodness. You're going to monetize my Windows search results?

Leo: Yeah, isn't that - well, the first step was putting Bing results in your search results. And that should have been a red flag.

Steve: Yup.

Leo: It's like, why would you do that? Oh, now I know why.

Steve: Uh-huh.

Leo: Uh-huh. Oh, lord.

Steve: See, this punishes the few people who actually do upgrade because most of us are just going to stay with 7, Leo.

Leo: Yeah. I think that's a pretty clear message for a lot of people.

Steve: Yeah.

Leo: Are you ready for questions, Steve?

Steve: Let's go.

Leo: Let's go: 11 questions, thoughts, and comments. This is Listener-Driven Potpourri #171, starting with John Shattuck, Information Security, in Washington state. John queries: I just listened to the thing on PRISM. What makes you think the NSA hasn't required that Google pass along, and MSN and Yahoo for that matter, their 128-bit encryption keys? My guess is they've been given the keys under the ask of national security, the same way they've been given or they've taken a piece of real estate to build data centers in their buildings. Could they just say, hey, give us the SSL keys?

Steve: I don't know that they could not. I mean, I think yes. I mean, I think they, unfortunately, can compel a company to do anything they want.

Leo: Almost anything, yeah.

Steve: And, now, it's not 128-bit encryption keys. Those are negotiated on the fly. We're going to get into this seriously next week, this difference between symmetric and asymmetric, short keys and long keys and so forth. It's complicated, but that's - we explain that stuff here. But...

Leo: So they'd be 1024 keys.

Steve: They would be, yes.

Leo: Or 2048 keys.

Steve: They would, yes. In the case of Facebook and Google, famously, they would be there. Basically the private key, which they never let out of their control, which is the only way we have of authenticating them, essentially, their security certificate. I don't know that anything prevents our governments from saying it's a matter of national security, are you not patriotic, and here's a letter compelling you to give up your private key. And by the way, we'll need you to refresh this when you change your private key. I mean...

Leo: Sure they have that information, actually.

Steve: Unfortunately, unfortunately, this is the world in which we now live. I guess, you know, people have been saying, "Oh, Gibson, you're nave. We've been living in this world for a long time. What was it that Orwell wrote in '1984'?" It's like, yes, I know. But we've got a timeline now. And storage has gotten so cheap, as this great EFF guy we quoted last week said, it's now so cheap to store stuff that it makes sense to store everything.

Leo: Steve Good...

Steve: And, yeah - yeah.

Leo: Yeah.

Steve: Sorry.

Leo: Of course we can just presume this is happening. Steve Good in Lexington, Kentucky wonders, do the numbers add up? Steve, thanks for the great information on NSA's PRISM program. I have a question. If the spooks are splitting and duplicating the data upstream bit for bit, how much data is that? I think we quoted a number from IBM, but I've forgotten. We could do the math.

Steve: It's a lot. It's a lot.

Leo: How many days or months can they store in their 5ZB data farm? I know this would require an estimate of the amount of data coming out of Google and other servers that are being tapped. Do the numbers add up? Can they actually store everything forever? Not forever, no. If they could store it all, what kind of system can search 5ZB of data in a useful timeframe? I guess I'm trying to understand, how much is a zettabyte?

Steve: A number of people have been confused by this, so I wanted to come back to it briefly. This system, as far as we know, the PRISM technology incorporates something that - a so-called "semantic analyzer." And this is this thing, this equipment produced by Narus, N-a-r-u-s. And we found the brochure for it that has on its second page a prism as its graphics, showing, like, what it does. And anecdotally we've heard that the much-more expansive brochure has been removed from public access, where they're bragging about much more about what the system does.

The point is that these are like filters which can be tasked, as in giving it a task. They can be tasked with pulling specific information from the flood. So we don't know, we're never going to know, probably, whether in addition to that they're just pouring this into some huge sump somewhere and keeping it all, or they're saving the encrypted stuff for later. We've verified that they feel they have the right, just on the basis of it being encrypted, that's reason enough to be suspicious of it. So they're going to keep that for later decryption, perhaps.

Maybe, then, this Narus thing operates in parallel to, in real time, find things that they're actively looking for that it has been tasked to find. Or maybe it is a filter behind which is storage, so it's only storing things of some specific relevance or interest. And if they're to be believed, it would be selecting things that are believed to not have domestic endpoints on each end, but at least one end is foreign, because that was the authorization that they were given was to do this on foreign communications. So we just don't know.

Leo: So, well, I can do a little calculation - well, WolframAlpha can. So according to IBM, the number of bytes of data created daily is 2.5 quintillion bytes, 2.5 quintillion bytes of data daily. And by the way, they say 90% of the data in the world today has

been created in the last two years alone. So this number is getting bigger faster. But let's say it stayed at 2.5 quintillion bytes. So I asked WolframAlpha to convert a quintillion bytes into a zettabyte. And that's 1/1000th of a zettabyte.

Steve: Whoa.

Leo: Every day.

Steve: Three years.

Leo: So, yeah, something like that. But that's - but that makes sense. That's even commensurate with what they're saying, isn't it. We're not keeping it forever. We're keeping it for a few years.

Steve: Real estate in South Utah, Leo. That's where you want to make - you want to build...

Leo: Hard drives.

Steve: You want to build - you want to buy land south of the current facility because they're going to have to grow that sucker.

Leo: Yeah. So it's 2.5 exabytes created a day of data.

Steve: I'm kidding, by the way, people. Do not go buy real estate. Just...

Leo: Soon the whole state will be one giant data center.

Steve: I heard myself. I thought, oh, no, no, no. Okay, that's not real estate advice from your security person. No. Do not buy real estate in...

Leo: Don't listen to that man.

Steve: No.

Leo: But 2.5 exabytes a day of data is a lot of data. And it's going up exponentially. So presumably, though, their capacity is going to go up. And they said 5ZB. Who knows what the real number is?

Steve: Yup. And we've got some questions coming up about that, too.

Leo: Yeah. I mean, at this point, I don't believe anything those sons of guns say.

Steve: I don't either, no.

Leo: And they even admit, well, we lie. Don't believe anything we say. They've even said that. Because we're spies. Jason in Newcastle, Australia shares some thoughts about "secure" email: I thought you'd like this one, Steve. I've been listening to Security Now! for a few months, and I am enjoying it on my long commute to work. Because my commute is so long, I'm currently in the market for a new house to rent closer in. But don't worry, I'll still listen.

So we followed the usual process, found and inspected a property. The agent told us that there was an online application to fill in. Being a sysadmin by trade, as well as having terrible handwriting, I thought, fantastic. More services like this should be paperless. I spent an hour filling in the form, then I arrived at the part where you need to verify your identity by uploading your scanned documents - license, pay slips, passport and the like. They had two options on the site: upload them over an encrypted SSL connection, or email them [laughing]. Do I even need to say which one I chose?

I spent the time scanning and uploading, fighting the irresistible urge to take the quick and easy way out by taking a photo of them and emailing from my phone. Hey, it's my identity at stake here. I'm not taking chances.

Steve: And they're asking him to prove his identity by sending these documents.

Leo: Right.

Steve: So these are identity-proving documents.

Leo: Yeah, these are the real deal here, no messing around. I finished the application and received a confirmation email informing that I had put in an application for the property. It began by saying "Your secure application has been emailed to <insert agent's email address here>." Need I say more? Thanks again for the great products and passing the time on my drive to work.

Steve: Oh, god. Well, yes.

Leo: SSL to us.

Steve: The lesson here, exactly, the lesson here is, we've spoken of it before, the weakest link in the chain. So, yes, nobody could snoop his connection to their real estate application-accepting website. But then for compatibility's sake they emailed it all off in

the clear to the real estate agent. It's like, here's...

Leo: <Sigh>.

Steve: ...the proof, yes.

Leo: <Sigh>.

Steve: Good story, Jason. Thank you for that.

Leo: Chris Lionetti, Bellevue, Washington, wonders if the numbers add up? I'm a reference architect for NetApp, and I used to build datacenters for Microsoft for five years. Datacenters and storage are my life.

Steve: Cool.

Leo: There's a problem - so, yeah, this guy has standing.

Steve: Yeah.

Leo: There's a problem with the NSA's new datacenter math. Let's assume 4TB HDDs. Let's assume they use the most dense rack storage available. That's about 60 drives per Rack U of space.

Steve: Now, wait. Now, stop right there. 60 drives per U? How do you get 60 drives in one U of rack space? That's...

Leo: I don't know.

Steve: I don't know. That's amazing, Chris.

Leo: I wonder if - yeah. That doesn't sound right.

Steve: I know.

Leo: In a 40RU rack I could fit 600 HDDs. So that rack's got 2.4 petabytes. A common datacenter room would - just imagine the cooling. A common datacenter room - and the noise.

Steve: And the power.

Leo: And the power - would contain 90 racks long by eight rows wide. That room would contain 1.7 exabytes. This ties in nicely with the calculations we just did, that there's 2.5 exabytes of data created every single day. Using 65 megawatts, I could power about six of these rooms. That's closer to 200K square feet. The info on the datacenter states 100K square feet of DC space. Really? I thought it was bigger than that. Oh, well. That would give me 10 exabytes. I think the information you have claiming it was a zettabyte must have been wrong. Five exabytes sounds more realistic, and that's still huge. Just want your podcast to be accurate. You are still talking about one million spinning hard drives. For comparison's sake, Microsoft's San Antonio datacenter is 150 megawatts, half million square feet.

And another email from Paul in Dallas simplifies the math even further: I love the show, blah blah blah, very high praise. In last week's podcast concerning the NSA and PRISM, you said X number of zettabytes of storage capacity. Surely you meant petabytes or exabytes. To have a zettabyte of storage would require one trillion terabyte hard drives. That's just not possible, given the cost of material needed. Have I done my math incorrectly?

Steve: Okay. So, great points, representative of our sharp listeners, many of whom said, uh...

Leo: Good critical thinking. I like that kind of critical thinking.

Steve: Yup. Now, first of all, I doubt that it was a typo because nobody's ever heard of a zettabyte before. So it wasn't like some editor said, eh, what's a big thing? Oh, a zettabyte. So all we know, first of all, this number comes from the NSA. This 65,000...

Leo: This was their press release, my friends.

Steve: Yes, this is them saying this is what they're building. Now, one thing occurred to me as I'm looking at the cognitive dissonance that this does set up, of course, is, well, are these hard drives? That is, is a hard drive today the most dense way we have of storing something?

Leo: Right.

Steve: Because hard drives are inherently online. But nothing says these 5ZB might not have basically a massive hard drive cache on the front end and something archival on the back end. I don't - I have not bothered to go into the current state of long-term, ultra-dense, ultra-large archiving. But we know, for example, that Google with their S3 service allows you to tuck data, like, further away somewhere, and it takes maybe a day to get it. But they've done something with it. Maybe they just unplug their hard drives, so they've got to go get one and plug it in. That's probably what's going on.

But for what it's worth, I mean, we could either say we don't believe the NSA, or we can

say, well, maybe we need to think out of the box a little bit more, that it's not just - everyone just wants to multiply hard drive size. But that assumes that's all we have to work with. And maybe there is - I haven't looked for a long time. I mean, back in the old days, IBM had all kinds of bizarre technology. They had, like, spools of magnetic tape and a robot arm as a big library system. And the robot arm would swing around, grab a spool, pull it out, and stick it into a reader. So there were a limited number of reader/writer stations, but a vast wall of spools. So you couldn't get to them all at once the way you can with a huge hard drive array. But you could get to them eventually.

Leo: So this is a slide from the Army Corps of Engineers about the plan.

Steve: Ah.

Leo: And it gives us - doesn't say "data storage," but maybe our audience can crunch some numbers here: 65 megawatts, 60,000 tons of cooling equipment, four 25,000 square foot server facilities. So only 100,000 square feet.

Steve: Total.

Leo: Total. So, yeah, I'd have to think there must be something more dense than hard drives. Maybe not as fast. But more dense.

Steve: Do we know that there's not a basement?

Leo: What's in the basement?

Steve: Maybe there's an elevator that goes way down, Leo.

Leo: Yeah.

Steve: Don't know.

Leo: Salt mines, baby. It's a great mystery. 'Tis a puzzle.

Steve: So thank you, listeners, for your sharpness.

Leo: Yeah, good math.

Steve: Yup.

Leo: And I really like it that people are using their critical thinking. I looked at that number, and I just go, yeah, that sounds right.

Steve: Okay, big, wow.

Leo: They said it. Must be true [forced laughter].

Steve: Yeah.

Leo: Apparently they're planning on scaling it to yottabytes. Alex wonders, should we be creating our own certificates? Steve, forgive me, I'm no expert in the area of certificates. I want to understand. That's why your podcast is the best. So far I know I can obtain a certificate from, say, from CA Cert or my iCloud email cert provided by Apple, somehow by default. This is nice, so I can provide verification of my person. But I'd need a recipient to set up something similar on their end; and with that, I and a recipient can send encrypted email to each other.

But there is a middle man in this. That's the authority. I guess my question is twofold: Can I trust the authority? Might their certificate be compromised somehow, or even stolen, allowing for spoofing of an identity? In the spirit of suspicion, can I revoke the certificate at any time? Secondly, can I create my own certificate? The answer is probably yes, but how can anyone verify my certificate as trusted? I'd be interested in Scenario 2 because I control the certificate and when it gets revoked and the frequency.

Please feel free to edit this question if you intend to use it on your show. But after the NSA situation, I think it's good policy to practice as much encryption as possible. With the Internet today, god knows how important your insights are. One last thing: If I do encrypt, how strong should the encryption be? After all, the NSA has teraflops at their disposal. Hey, the new Mac Pro has teraflops. That's nothing [laughter].

Steve: So zenaflops, Leo, zenaflops.

Leo: Yeah, yeah, yeah.

Steve: Okay. So this is a great question. And I want to step back from it a little bit and sort of look at the meta question, which is - and all of our dialogue up to this point, even just in this podcast, has been sort of nudging us in this direction, which is that the one weakness of the public key encryption technology, the so-called PKI, Public Key Infrastructure, is our reliance on a certificate authority and our need to trust that authority. The authority signs our certificate. And if we trust the signer, then we trust the signing.

So the problem is, if we build a system based on that, that is the weakness. And as we saw, if the NSA or law enforcement compelled Google to give up their private key, if they compelled VeriSign to give up their private key, the trusted root certificate authority, I mean, why stop at Google? Go another link up the chain. Get the root authority private

keys.

The point is, I don't want to get people worried about whether their tinfoil hats are tight enough or not. But we're at a point now where we're seriously reconsidering the trustworthiness of the public key infrastructure, you know, on a theoretical basis. And again, people can say, well, Gibson, it was never trustworthy. It's like, okay. But we now know, we now have more reason for tinfoil than we had before.

So my bottom line for Alex is the only way that we can have security moving forward, if that's what we really want, is to no longer use the public key infrastructure. And that's essentially what I was referring to last week when I was talking about the Threema communications tool for Android. And I don't remember if it's iOS, also. I think it is, but not yet BlackBerry. There, you use - you have, like, three levels of authentication. The highest one, where you get three green dots, requires that the two devices be set face to face so they can simultaneously cross-snapshot each other's key. And the point is we only require trusting a third party when that's not possible.

When the two parties, the end parties, cannot meet physically, we have to trust a third who has essentially met them each. So the third party has met Alice, and the third party has met Bob, and is able then to assert to Bob and Alice that they are each who they claim. Well, if we can't trust the third leg of that stool, then Bob and Alice have to meet. When they meet, they can securely exchange keys, and then that's what we're reduced to, essentially. That's where we are today is, rather than trusting a third party, we can arrange to essentially cut any third party out of the loop. And I think we're going to see utilities more and more in the future like Threema that say, okay, this is what we do now.

Leo: I don't know about Windows. Mac has an easy way to generate your own certificates, self-signed certificates. You can create one. You can create a certificate authority, even, or create a certificate for someone else as a certificate authority. So you can set yourself up as a CA. This is with Apple's built-in Keychain Access that's in every single Mac that's shipped. There also is, of course, I use PGP or OpenPGP. The GNU Privacy Guard is my choice because I like open source software for this kind of stuff. You really want to use open source.

Steve: Yeah.

Leo: And the notion there is, when you use that, is that you create your own keys. No one else has access to your private key except you. And then you send your public key to a key server. And then what you want to do is you could either have a signing party, this is the thing you talked about where you get together in physical presence, and you sign people's keys, because the more people who said, yeah, yeah, that's Leo's key, the more likely it is in fact Leo's key.

Steve: Right.

Leo: And so what I could do is give out the hex number. It's, I don't know, it's 10 digits or 10 hex digits that is my key and say, that's me. I could do it out on the air, for instance, say please sign that key. That is me. And then people would sign it. So

that keeps a government party out of it entirely.

Steve: Right.

Leo: No one has your private key if you generate your own certificate, or you create your own PGP key.

Steve: And, yes, no single centralized authority is making any representations.

Leo: Right, right. That's the negative is that you have to have - it's what they call in PGP terms a "circle of trust." But I think that that's, increasingly, we've got to do more of that. I think. The fingerprint, yeah, that's what that, whatever, 10-digit code is.

Christian Loris, Melbourne, Florida, says maybe General Petraeus gives you more conclusive proof of Steve's PRISM Theory: In your assertion that the NSA didn't need to be directly in bed with Google or Facebook, let's consider the story of General Petraeus's covert communication techniques with his girlfriend. Going on the assumed facts that SSL is secure - he was using Gmail, I think; right?

Steve: Yes.

Leo: ...and that PRISM is picking up the streams of unencrypted email and other traffic outside the major providers, General Petraeus communicated with his mistress via a shared draft folder on Gmail. That's pretty clever, isn't it. He didn't even mail it.

Steve: Uh-huh.

Leo: He had a shared - he just would create a draft and save it, and they both could log into that server. He knew that almost anybody communicating with Gmail's website is not suspicious to the NSA collection efforts. Too many people use it to make it interesting in most circumstances. Gmail's SSL is secure and/or difficult to break unless the NSA has a very specific set of traffic it wants to crack - or has their private keys. NSA/PRISM would need some very specific information before getting a warrant that Google would be happy to comply with. General Petraeus and his mistress would securely connect to Gmail, leave emails for each other in the draft folder of their shared account. Boy, that's clever. I'll have to remember that next time I want to have a girlfriend.

Steve: Yeah.

Leo: The message would never leave Google's datacenter and never be seen by

PRISM.

Steve: Yeah.

Leo: This pretty much backs up the fact that many of the Googles or Facebooks may not have been complicit in the spying. At the time Petraeus was caught doing this, it was also commented this was a common technique used by drug dealers and terrorists to stay off the radar. This points to an awareness of the types of collection efforts that might be in use by law enforcement or the NSA. Yeah, you'd think Petraeus, what was he? He was head of the FBI; right? What was his job at the time? I can't even remember.

Steve: I think he was in Afghanistan. I think he was...

Leo: Yeah, no, but then he came home, and he was...

Steve: Oh, yeah.

Leo: Yeah. No, director of the CIA. He was the CIA director.

Steve: Oh, well, okay.

Leo: So presumably he knows and knew...

Steve: What's going on.

Leo: ...all about PRISM; right?

Steve: Yup.

Leo: And knew what the capabilities were and thought - he probably thought, though, that he wasn't a subject. Anyway, at the time I thought this was silly for them...

Steve: He didn't want to get - he didn't want - yeah, go ahead, I'm sorry.

Leo: But now, in light of everything we've learned, it actually makes sense. So he was doing what he thought would be safe, and obviously - now, as I remember, I think he turned over his Gmail when he was being investigated. I don't think...

Steve: When he turned over his resignation.

Leo: Well, but, yeah. I'm trying to remember the whole - it is germane that the director of the CIA considered this a relatively secure method.

Steve: Mm-hmm.

Leo: And did not send emails. He said, no, we're not going to send emails back and forth. That would be a bad idea.

Steve: Yes, no, exactly. Notice what he wasn't doing. He wasn't doing what everybody else was doing.

Leo: Right.

Steve: Except the terrorists and the drug dealers.

Leo: Drug dealers.

Steve: And so, again, we're the last to know. Everybody else already knew all of this that was going on.

Leo: Apparently. It's the honest folks.

Steve: Yup.

Leo: Andrew Stevenson, Dorset, U.K. points out a great resource: Steve and Leo, the Electronic Frontier Foundation has put together a nice list of alternative programs that can be used to help thwart tracking by the NSA's PRISM program: prism-break.org. Interestingly, near the bottom of the list, when detailing alternatives to iOS, it simply states that it is insecure since it contains hardware tracking. Do not use. I like the name "prism-break." But it does make presumptions about what - presumptions about how PRISM works that we just don't know.

Steve: Well, it's an interesting page. And it's a little sad in places. For example - wow, they just changed it. Huh. They've changed it from this morning when I was looking at it because it made a sweeping statement about, under web browser category, it says Apple Safari, Google Chrome, and IE as proprietary. And over under Notes it said you can't really use any of these browsers because we have no idea what's going on in them. And it was like, whoa. Wait, maybe it was operating systems.

Leo: Yeah. Huh. Cloud storage? So they've got three columns, or two columns: Proprietary and Free Alternatives. And presumably the free open source alternatives are - I don't know. Stop reporting online - it's not really clear what they're saying at this point. Are they saying that the free alternatives are safe?

Steve: Well, they've got a column of proprietary and different categories, then a column of free alternatives, which they're endorsing in lieu of these proprietary ones, and then notes to, like, embellish, like here are the concerns and here are the issues.

Leo: So instead of using PayPal or Google Wallet, use Bitcoin or other alternative crypto currencies, for instance.

Steve: Right, right.

Leo: And I guess when they use the word "free" they're not saying "free" as in beer, they're saying "free" as in freedom. And they do say that iOS is insecure. There is no free alternative to iOS [laughter].

Steve: Ouch. Yeah. Anyway, I do commend our listeners to it: prizm-break.org. Good stuff.

Leo: Wow. It's a good name. Jack, Fairfax, Virginia, wondering whether "Chrome convenience" is going just a bit too far: Steve, not sure if this has come up on Security Now!. At work, my PC is pretty much locked down, and so many sites were nonfunctional under Internet Explorer, the admin agreed to install Chrome. I promptly added NotScript and WOT extensions. When I got home, I was at first quite pleased to find that the same extensions had auto-magically installed when I fired up Chrome on my Mac. If you have syncing turned on, that's what will happen, by the way.

While I appreciate the convenience of extensions replicating across installations and the consistency of experience this offers users, I had a tinge of concern that I had made a change at work, but it affected my computer at home. I wasn't aware of the behavior. Well, dude, you turned it on. So it was a surprise to me - this is often the case. People turn things on and forget.

Steve: And so it's not on by default?

Leo: No.

Steve: Okay.

Leo: In this case, a pleasant one, which is what Google is of course intending. But

couldn't the same mechanism be a foot in the door for someone up to no good? Yeah, it asks you if you want to have syncing turned on. It walks you through this. In the converse - but probably he hadn't installed Chrome in a while; right? In the converse situation, could someone sit down at my Mac at home while I get a beer and install a nefarious extension which quietly replicates to my desktop at work? I realize all bets are off if someone gains local access to a device. But in this case, is the remote system at risk, as well?

Steve: So I think they're saved a little bit by the fact that it's off by default. But, I mean, he's certainly right. This is another classic example of convenience versus security. I mean, it's absolutely convenient to have your Chromes syncing through the cloud, even to the extent of installing extensions on other Chromes that you install on one. But, I mean, yes, this could be used as a foothold to, like, get something nefarious in a work environment. Essentially, the problem is that Chrome in this instance is creating a bridge between two different security perimeters. You've got your high security perimeter at work and your relatively low security perimeter at home, where you're having beer and your friends are over and they're screwing around with your Mac. And there's...

Leo: Really easy thing to turn off.

Steve: Good.

Leo: And it's completely granular. You can say sync bookmarks, don't sync extensions; sync settings, but don't sync themes. You can totally do it.

Steve: Good. Good for Chrome.

Leo: Yeah. I think that this is an example of the problem exists between computer and keyboard or whatever that is, PEBKAC. And I sync everything because I feel fairly secure in the knowledge that...

Steve: You have control of your environment.

Leo: Yeah. And these extensions - Chrome is pretty good about not installing dangerous extensions. Although I can't say that there's definitely no dangerous extensions installed.

Wolfgang Muenst in Munich, Germany offers a note about our podcast length and composition [laughing]: Just heard on a recent episode of Security Now! some listeners want to shorten the show or keep the general stuff out. PLEASE DON'T! I totally enjoy the way it is, actually prefer to broaden my horizon every now and then with content I've never come into contact with before. Thanks to you and Leo. Hopefully, we'll see at least another 400 episodes of Security Now!. Thank you, Wolfgang. Great.

Steve: Yeah. And this does echo many sentiments that I saw after that discussion that you and I had about this last week or the week before, a lot of people saying, hey, no, don't change it, it's what we want. You guys rambling around a little bit and talking about other stuff, that's good, too. So thank you, everybody.

Leo: Question 11 from Adam, Washington, D.C. He wonders, when are you going to be in Petaluma again? Steve, I'm a high school student, rising senior - congratulations, Adam - from Washington D.C. who, over the past two years or so, has become an avid listener of Security Now!. This gives me hope for the youth of our nation. Since I discovered the show, my interest in the field of Internet security has grown immensely, and it's thanks to you that a lot of my knowledge about the finer details of technology has expanded. So I have become just giddy - giddy - about understanding security-related issues. Thank you very much for the hard work you put in each week to create such a fantastic netcast.

My family and I will be heading to California this summer. Oh, I guess he's in Washington state. No, it says Washington, D.C. We'll be heading down to California this summer for a vacation, and I've convinced them to take a detour on our trip down the coast to visit the Brick House on Wednesday, August 21st, in order to see Security Now! live. I heard at the end of Episode 408 on PRISM you might be in-house one of the weeks in August, and I was wondering if you might have any more details on what week that would be. Getting to meet you in person would be unbelievable, but I'd still be thrilled by the experience of seeing the show live and in person.

Regardless of when you'll be in town, let this message serve as a huge thank-you for all the time you've dedicated to the show. I know it's a lot of work and can guarantee I'm not the only one you've inspired with your service. Adam. Man, this kid is eloquent.

Steve: He is.

Leo: What a good writer. Wow.

Steve: Yeah.

Leo: So when are you going to be in town?

Steve: Unknown. I will aim for the 21st. I'm coming up to hang out for a while with Jennifer, my girlfriend, and she is going to be up visiting people in Northern California. And so I'm going to definitely try to make it a Wednesday or a Sunday, so either Security Now! or TWiT. But I just don't know yet. Jenny is great on plans and running around, but...

Leo: Not that one.

Steve: I never really know until she says, "Didn't I tell you that?" "No, honey, I didn't

get the word." So as soon as I know, I will let everyone know because it'd be great to see listeners when I'm going to be around. So other people have asked, and I just - I don't know yet. But as soon as I get a date, I will, on the podcast preceding it or more, I will say. So thank you very much. And Adam, thanks for the great note.

Leo: Jenny, Jenny, make up your mind. Figure out when you're coming down here. Up here. So, Steve, that concludes this reading, this dramatic reading of 11 questions from our vast audience.

Steve: Until next week, when we're going to plow into the technology of handshaking and SSL connections and what can be done, the details of perfect forward secrecy which really is getting a lot of attention now relative to the NSA's presumed interest in sucking up the content and decrypting it at a later date.

Leo: Well, friends, this is the deal. You must go now to GRC.com. You must browse through the vast stacks of information. You must purchase a copy of SpinRite, the world's finest hard drive maintenance and recovery utility, and knowing that you will get a free upgrade to the Mac-compatible version whenever it is available.

Steve: Yup.

Leo: You must...

Steve: Even before.

Leo: You must, if you have - even before. You must, if you have questions, go to GRC.com/feedback. If you would like to download a copy of this podcast in glorious 16Kb audio, which sounds a little bit like it was recorded in the 18th Century.

Steve: Oh, Leo.

Leo: Down a tube. But small, small. You can get that from GRC.com. He also has - Steve has great transcriptions written by hand by a human, all at GRC.com. We keep full quality audio and video available at our site, TWiT.tv/sn. And you can even watch us do the show live, which we do every Wednesday at 11:00 a.m. Pacific, 2:00 p.m. Eastern time, 19:00 UTC. TWiT.tv is the place for that. Hey, Steve. Have a great week. Have a great Fourth. Are you doing something tomorrow, day after tomorrow?

Steve: I'm going to be coding.

Leo: No, it's tomorrow. That's tomorrow.

Steve: I hope I...

Leo: Coding.

Steve: I hope I'm coding up a storm. I'm going to just code away. Quiet days like that give me more chance to work, so that's what I want to do.

Leo: Coding by the rockets' red glare. That'll be Steve. That's great.

Steve: Thank you, my friend.

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