Coin, CryptoLocker, Patent Trolls & More

Description: Following another week overfilled with interesting security-related news, Steve and Leo spend an hour and a half diving deeply into an updated (and likely very close to correct) understanding of the Coin payment card, news on the CryptoLocker front, a close look at a patent troll case that has so far gone the wrong way, and much more.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-432.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-432-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here, and there's so much security news that we're going to defer the question-and-answer segment for an episode or two and just get to some of the big stories, including a second look at Coin, Bitcoin - the other coin - and a whole lot more, all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 432, recorded November 27th, 2013: Coin, Patent Trolls, and More.

It's time for Security Now!, the show where we explain your security, your privacy, and how the Internet works and all that jazz with this guy right here, the Explainer in Chief, Mr. Steven "Tiberius" Gibson, creator of SpinRite, the world's finest hard drive and maintenance utility; coiner of the term "spyware," author of the first antispyware tool. He's a maven; he's a connector; he's a Grand Poobah. I'm wondering how long I can keep talking without him saying, all right, enough.

Steve Gibson: Okay, Leo.

Leo: [Laughing] Hi, Steve.

Steve: Hi, Leo. So this is another one of these weeks. As I was putting things together, I thought, why do I feel apologetic that we're not having a Q&A, when we have just too much really interesting stuff to talk about? I threw away about half of the things I wanted to talk about - actually I pushed them into next week to see how that goes because there was just - there was too much that happened. And as I was looking at some of these things, there are some things I want to really spend some time on. And I

really think the podcast serves our listeners better if we do fewer things in greater depth...

Leo: I agree. I agree.

Steve: ...that you can only get here, rather than just...

Leo: Shallow you can get anywhere.

Steve: Right.

Leo: Let's get deep. Let's go deep.

Steve: Right. So we got some really just, you know, a great podcast, nominally a Q&A, but, sorry. What's happening is I did follow your advice, by the way, and have completely fouled up my entire Twitter feed.

Leo: I saw all those @replies. Did anybody get angry at you?

Steve: Only a couple people were grumpy because they liked using - they liked to be able to use my feed as sort of an archival...

Leo: There's a setting that they can - oh, well, I'm not going to - they can figure it out.

Steve: Yes. And but the overwhelming majority of people really like the fact that I'm doing @replies because...

Leo: It's so cool that you're responding.

Steve: Well, I mean, I always have been. I've been doing this, but I've been doing it with DMs, so no one was ever able to see the fact that I was actively replying to questions that were coming in. So, yeah. I think this is an improvement. It does foul up that cool timeline that someone was maintaining. So what I'm going to end up doing is, once I get SQRL and SpinRite behind me, something I want to do then is to put up a filter page, essentially a filter page at GRC so that my statements in Twitter can always be found in once place. So that way we sort of solve the problem. As I understand it also, verified accounts have the opportunity of having that setting you were talking about, Leo, but not the...

Leo: Oh, you're not verified. Oh, ho ho ho.

Steve: Nope, never been. So that's the other thing I want to do is see if I can get Twitter to verify me, in which case people would be able to use that setting in order to not see the conversations.

Leo: Right, right. With me they can leave off the @replies. Got it. I didn't realize...

Steve: It's because you, Leo, are...

Leo: ...I have something special.

Steve: You are one of a kind.

Leo: You can always, well, there's also Gina Trapani's ThinkUp, which will allow you to track stuff.

Steve: Well, any client, any client will allow you to do this, too. I mean, it's only people who are looking at the raw feed, looking at my so-called "timeline," that now see all the interaction. But, I mean, really it's been a huge win. And I was just teasing when I said I followed your advice and fouled up my timeline.

Leo: Well, there's a lot on it now, which is good. Which is good.

Steve: And it was always there, it was just all private. And now it's public. And people are loving the fact that they can follow conversations that I'm having with other people. So it's like, yeah, I've been this active, but it was just on the DL.

Leo: Yeah. Sure, yeah.

[Talking simultaneously]

Steve: So today...

Leo: ...bother me, but I guess...

Steve: ...we're going to talk about Coin again, Take 2 of Coin because now I've figured out what they're actually doing. And that picture that you can see on the page reveals it. We've got a big CryptoLocker update. And GRC is now offering CryptoLocker for forensic experimentation and download.

Leo: Oh. You have a few lying around, huh?

Steve: The EFF is tracking who's encrypting what. There were massive man-in-the-middle attacks on the Internet, or not, maybe. Who invented public key encryption? And a look at a very high-profile patent troll case, and much more. So a great podcast.

Leo: I know where you're going with that one. All right. Where should we start?

Steve: Okay. So we've got to revisit Coin because I was wrong in my clever guess...

Leo: Such good detective work.

Steve: ...about the idea that they were proxying. If it was impossible to change in some manner the signal that the strip, the mag strip is generating, then my approach was the only feasible one. Turns out that there is something that you can do. You're not actually rewriting the mag strip because that seemed unfeasible. But I ran across a photo of the very first prototype that Coin's inventor created. And it's in the show notes. You can show it during the podcast now, Leo. And the moment I saw that, it's like, oh, okay. Now I know how it works. Even complete with an Arduino sitting there wired up next to it.

Leo: Coin is really a sideline to what they were planning as a startup. They wanted to license this Arduino technology.

Steve: Oh, okay. Well, it makes sense. I bought one for 50 bucks, prepurchased. You've got 17 days now remaining for their 50% discount. Then the price doubles to a hundred dollars. And I don't have a need for it, which is why I'm not as excited about it as other people. I will probably delaminate mine just because I want to completely end the mystery, and I'll show everybody the inside of mine.

But so the idea is that there is a coil in the card, and probably, maybe, there are some contacts in the card, maybe an inertial sensor so that it knows when the card is being moved and to what degree. But so essentially it's very much like the second Loop product that we talked about last week, where those guys had a magnetic coil, thus a loop, attached to a smartphone, and they would bring it within proximity of the reader, and it worked in 90% of the readers. Basically, it was an induction coil inducing a sympathetic current in the read head, which the point-of-sale terminal could not differentiate from the card being swiped. So this is essentially the same technology, but they've bundled it into a credit card format and given it battery power that lasts a couple years and some smarts.

So the idea is, as you swipe it, it uses an alternating electrical field through a coil to produce an alternating magnetic field across the entire extent of the strip. And that's sort of the clever part is it doesn't really matter physically where the card is. The whole strip is simultaneously generating the same magnetic field. But if it were a fixed set of reversed magnetic domains, then you would have to be moving the card. Here you don't technically have to be moving the card because the whole strip is generating the same field.

Well, now, that, then, as I was reverse-engineering this with scant documentation, then the question is, okay, wait a minute. A card actually has three stripes, that is, there's one large magnetic strip, but within that are three tracks, in the same way that a cassette

tape used to have - it would have two stereo tracks on one side of the tape and two other stereo tracks on the other side, so a total of four tracks. And so you'd listen to stereo on one side of the cassette, then you'd flip it over and listen to stereo on the other side.

So these magnetic tracks are - they're immediately adjacent to each other. They're .11" apart and .11" wide. So the question is, wait a minute. If the card has three tracks, that is, a regular magnetic card has three tracks, but we only have the ability to essentially simulate one, how does the mag strip reader know which one we're generating, and how do we determine which one we want to generate?

Well, it turns out I tracked down the ISO IEC specification. It's ISO 7811-2. And there's, like, 7811 is the overall spec, and then there's dash one through, I think, six or seven, which are sort of the sub-specs because all of this is covered with a set of standards so that all the credit cards are the same size; they're the same thickness; you have interchangeability; there's broad global agreement, thankfully, so that we have, like, one standard for those so-called ID cards.

And in fact, immediately upon hearing my original theory for how this worked, a listener wrote back, tweeted me and said, wait a minute, that could work for major credit card companies, but how would you handle loyalty cards and essentially local use of cards, not global financial clearinghouse? And they were, of course, completely correct because the idea is that this can do all of that. So these three tracks differ dramatically in their density and format, the type of data they contain.

There's an alphanumeric track, so-called Track 1, which is 210 bits per inch of data, carrying 7 bits per character, where that's 6 data plus a parity bit. And that basically gives you uppercase alphanumeric and a whole raft of special symbols, some of which are reserved. And as the spec says, the maximum character count consists - or the stripe contains data, control, start-and-end sentinel characters, and then a longitudinal redundancy check character, that altogether will not exceed 79 characters. So we have an alphanumeric strip, 210 bpi, that can hold 79 characters.

Then there's a numeric track which is a lower density at 75 bpi, so essentially a lower frequency of recording. And that's strictly numeric. That's 5 bits/character, and it's allocated as 4 bits of data plus parity. Of course 4 bits of data only gives you the numbers plus a few special characters, all of which are reserved for control characters. That's the second track.

Then the third is a higher density, but also numeric track. So that's back up to 210 bpi, 5 bits/character, 4 for data, plus one for parity. And that can contain 107 characters. So that's the largest number of characters because they've sacrificed the number of bits per character running at just 5 rather than 7, as is the case for the first track.

So three very different track formats. Each of the characters has a parity. And then the key is there's this - there's error detection. We have, as I said, within each character, we have a parity bit. But then at the end of the entire track, there is what they call a "longitudinal redundancy check," an LRC. And it's nothing but even parity for all of the bits in that bit position.

So, okay. So visualize it this way. Imagine that we took the characters, whether they're 7 bits or 5 bits, and we printed them out in binary, 01, 110, 011 and so forth, and then stacked them on top of each other so that we created a matrix of ones and zeroes, where the characters are on the horizontal, running down vertically. But that would mean that the parity bit would be the parity for the row. Well, then, the longitudinal redundancy

check is the parity for the column. So it's one character at the end which creates even parity for the column. And here's the key: If the terminal doesn't see an error-free read in both horizontal parity and essentially longitudinal or vertical parity, it simply rejects that track. Which means that the transmitting technology is able to decide which one of the three tracks it wants to send, and then it arranges the error correction, both senses of parity, so that it's correct for that track.

Well, there's no way it could ever also be correct for the other two. So essentially all three tracks will receive the same data. But it will only be error free for the one it was designed to be targeted to. Thus this single card is able to generate data for any of those three tracks. And I don't know whether cards ever use two tracks at once, that is, for alphanumeric and numeric at the same time. They wouldn't need to. For example, just Track 1, which is alphanumeric, that could contain your name, the card number, the expiration data and so forth, easily, all together, just on a single track. So it's probably the case that Track 1 solves the problem.

I didn't go any further to look into the international standards of actually what type of service used which tracks. But that answers the question convincingly, I think, about exactly how this technology works. It's basically very similar to what Loop does, but they manage to cram the whole thing into a credit card that lasts for a couple of years, which I think is very cool. And whereas Loop is only able to achieve 90% compatibility, this presumably could be your hotel keycard, your loyalty cards, virtually anything. It's able to emulate the magnetic strip on standard ISO format cards. So I think now we know.

Leo: Yeah. I still wouldn't spend a penny on it; but, hey, whatever.

Steve: Well, it's just, yeah, it's not a problem that I have. I've got one card, and it works just fine. And many people said, wait a minute, mag strips are old news. Chip and PIN is where the world is going, and this doesn't do that. So it's like, yeah, that's true.

Leo: Yeah, although we were also going metric about 30 years ago. I mean, the U.S. has a long history of being resistant to how everybody else in the world does it.

Steve: Yes. And it's like, okay, what - I guess my reaction to all the people who pooh-poohed this because it wasn't Chip and PIN compatible was, well, I don't have a single Chip and PIN card. I don't own one at all.

Leo: Supposedly that's going to happen in 2015. But again, like I said, we also were supposed to go metric in 1972. So I don't know. I don't know if that'll happen or not. It should happen. It's a much more secure way of doing things, it's a much preferred way of doing things, and it's how everybody else in the world does it. But so was metric.

Steve: Well, okay, now, if you did Chip and PIN, that is, if you did - how do you solve the problem of giving the waiter your credit card to pay your check?

Leo: The bill. In Europe, what they do is they come to your - and I think this is also

a very good idea. They come to your table with a card reader.

Steve: Ah.

Leo: And you have to enter the PIN. It's actually - it's called dual-factor authentication, I believe.

Steve: Ah. What a concept.

Leo: You know, to me, and I've said this before, I'm not going to belabor it, it's a scam. They're taking a lot of money. They're spending that money on advertising. You'll see Coin advertisements all over Federated Media sites like 9to5Mac, everywhere. Which it's like a pyramid scheme. I don't think they're ever going to ship a card.

Steve: I am also somewhat skeptical, frankly, having looked at the spec, having looked at their - there was one site that I linked to. Oh, I forgot to tell everybody who's listening that the show notes will return to GRC's Security Now! page. Now that I've sort of screwed up, and I shouldn't really say that, now that I've changed...

Leo: Your Twitter. You screwed up your Twitter. Go ahead and say it again. Now that Leo has ruined your Twitter feed...

Steve: I'm not longer able to say go look at @SGgrc's Twitter feed because now it's all full of my responses to people who are tweeting me. Whereas I used to do DMs, now I'm no longer doing that. So I'm unable to post links in my Twitter feed and use it as my means for communicating show-specific, episode-related links. Now, it may be that the guy who's aggregating my feed, the guy who's at - I've created a bit.ly for it, bit.ly/SGgrc. That used to be really cool because he would aggregate them by Security Now! episode. That's all blown to hell, too, but maybe he'll fix it because he could certainly filter them as I intend to eventually get around to doing at GRC. So but in the meantime, I'm going to now, since you're showing the show notes anyway on the air, Leo, I might as well post the PDF on the site. I used to do it a long time ago.

Leo: That's a great idea, yeah.

Steve: But that way everybody gets the links. So links that I refer to, you can get them by getting this PDF from GRC at GRC.com/securitynow, where everybody knows we have the small versions of the podcast and Elaine's fabulous transcripts.

What didn't fit into the show last week because we had, like, a two-hour podcast, and it ran off the end, but I wanted to mention it because it was a little bit of an update on SQRL also, I got a nice note from a Rick Brooks, who's a listener in Columbia, South Carolina, sharing from October 18th, just last month, his very short note. He said, "Steve, I just purchased a copy of SpinRite 6 to use on a MacBook Air." So we know what

that means. That means an SSD. He said, "...on a MacBook Air that had a dead SSD drive that I had already tried every type of scan I could find and could not get any data. The machine would not boot up, and the Mac drive utility failed to do any repairs. This was a friend's machine, and she had her life on the drive with no Time Machine backup. She was really upset.

"After getting an adapter to convert the drive to use on a SATA interface, I ran SpinRite. I put the drive back in the machine, and it booted up. Wow. I completed a Time Machine backup today, and the laptop is running great. Thanks for the great software. Signed, Rick." And he said, "P.S.: Please get back to SpinRite ASAP. I know you're out saving the world via SQRL, but you've got me waiting on the new release. So please hurry now."

So anyway, I wanted to let everybody know. Some people have asked, where is SpinRite? Well, SpinRite is on hold, that is, SpinRite 6.1 is on hold while I nail down the final details of SQRL, this login technology. That continues to be proceeding very nicely. We have the syntax completely nailed down and agreed to, and we're now working on the semantics side, the specific, like, error message numbers and the details of the interchange between the client and the server. As soon as that gets done, I will write a reference implementation that everyone can use to check their own, in their own other languages and so forth. And then I'm done and back immediately to SpinRite 6.1. Rick, who owns 6.0, and everybody else who owns 6.0, or who purchases it in the meantime, will get 6.1 for free. So, and the good news at that point is he won't have to remove the SSD drive from a Mac in order to run it on a PC. He'll be able to run it natively on the Mac. And again, free upgrade for everyone.

So, but there's only me, and right now I'm spending full time on SQRL. And then I will be back to the next phase of SpinRite 6.1, which will be adding the AHCI compatibility so that it runs across all Intel-based machines, and then we'll get it out the door.

Leo: Neato.

Steve: Yeah. There was an interesting story that got a lot of press and generated a lot of upset that I'm somewhat skeptical of. We've never covered Border Gateway Protocol (BGP) in detail, mostly because it hasn't been necessary. We've sort of been able to explain what it is. What it is, is it's the language which BigIron Internet routers use for exchanging their routing tables and routing table updates. We've talked often about, in a broad sense, how the whole Internet works by just being this loose coupled interconnected network of routers where, when a packet that is addressed to a certain IP arrives at a router, the router looks at its routing table, and basically the router is like an octopus. It's sitting there in the middle with a bunch of links going to different places. And a packet comes into it across one of these links, and the router simply refers to a table that tells it which link to send the packet out of, sort of to send it on its way.

Like the idea is the routing table has a coarse understanding of which link is like the destination lies down. So because we, for example, with IPv4, we've got 4 billion IPs, there's no way to have, there's no practical way to have an entry for every single IP. But we don't need that because we know that IPs are allocated in chunks. So, for example, Comcast will have a big block of IPs. All of their customers are within a big block of IPs. So when a packet with one of those IPs lands in a router far away from Comcast, all that router has to see is that, oh, that's owned by this larger aggregator from which Comcast buys a smaller portion of their IPs. So it goes - so it sort of aims it at that larger aggregator. So my point is that these routing tables are very coarse for a large percentage of IPs, and only fine-grained when the router is physically closer to the ISP. It

then makes finer grained decisions about where to send it.

So the news that came out was that there was, like, massive man-in-the-middle traffic hijacking going on. Now, unfortunately, this was from a company, Renesys, that sells monitoring and sort of like detection protocol and service for detecting this. So putting out this press release was sort of like Symantec telling us how bad viruses are. It's like, yeah, okay, we know that's bad. But it's also a little self-serving.

Ars Technica picked the story up, which is why everyone was aware of it. And our illustrious reporter, Dan Goodin, he said, "The ease of altering or deleting authorized BGP routes, or of creating new ones, has long been considered a potential Achilles' heel for the Internet." And he's certainly right about that. He said, "Indeed," and we'll remember this because we covered it on the podcast, "in 2008 YouTube became unreachable for virtually all Internet users after a Pakistani ISP altered a route in a ham-fisted attempt to block the service," that is, YouTube, "in just that country.

"Later that year, researchers at the DefCon hacker conference showed how BGP," what we were just talking about, the Border Gateway Protocol routes, "could be manipulated to redirect huge swaths of Internet traffic. By diverting it to unauthorized routers under control of hackers, they were then free to monitor or tamper with any data that was unencrypted before sending it on to its intended recipient with little sign of what had just taken place."

So what we have is we have relatively good security for BGP, but not perfect. One of the reasons that we want unpredictable TCP sequence numbers is that routers establish TCP links. Border Gateway Protocol runs over TCP. And one of the ways of hacking into a TCP connection is if you're able to guess the sequence numbers in TCP. Then you can spoof traffic from one or another router, which it will trust because the trust is simply the point-to-point connection between two routers.

But you could, if you can spoof the source IP, which you can do with raw sockets, and if you knew where the sequence numbers were, you could essentially insert your own data into a router's table. And in fact that was done historically. It's the weakness that early TCP stacks had, that they had predictable sequence numbers in their TCP communications that allowed this kind of tampering. So it is inarguable that this is, as Dan writes, one of the Achilles' heels of the Internet even today.

So the evidence, however, because what Renesys wrote was that, since February of this year, 38 distinct events they have detected using their technology, their monitoring technology, in which large blocks of traffic were improperly redirected to routers at Belarusian or Icelandic ISPs. When they inquired what was going on, they initially didn't receive any response. And then later, as they were putting this formal announcement together, they tried again, and the response they got was, oh, yeah, sorry about that, we had some bugs in our routers, and we've updated them, and the problems went away.

So I think that's probably the truth. But whether or not it is, I mean, it seems very unlikely because, I mean, this is very easy to see. If you did a so-called "traceroute" while this was underway, you would see your packets heading off to Belarus and then coming back onto the Internet and going about their business. So it's not like this is anything that you can do stealthfully. This is very obvious to anyone monitoring where packets go on the Internet.

But the takeaway is, yes, we need HTTPS everywhere, all the time, because any nonencrypted traffic is definitely subject to this kind of a BigIron router, not little home routers, but routers that are out there in the middle of the Internet, moving all of this

traffic around. The whole routing table technology, it works, but it was probably meant to be replaced and no one's gotten around to it because it's really not as robust as it could be or, arguably, in this day and age needs to be.

Along the lines of security on the Internet, we've got the news, the good news, welcome news from Twitter that they have implemented forward secrecy for Twitter.com, api.twitter.com, and mobile.twitter.com. And we know what that means. That means that somebody decided this would be a good thing to do and, just as I did last week, they reordered the cipher suites which their servers are offering so that they would preferentially offer the ephemeral Diffie-Hellman key agreement suites over the non-use of that, the non-ephemeral suites, which do use the server certificate in order to encrypt the key.

And what they found was, immediately upon making this change, 75% of Twitter's connections began using elliptic curve Diffie-Hellman key agreement. So that is to say that there were all these clients out there, already using Twitter servers, that were ready to work with that cipher, but they required Twitter to make the change in order for forward secrecy to come up to speed. And it's now only older clients of various stripe which, you know, 25% making connections which are not using ephemeral Diffie-Hellman key agreement. So another company takes security more seriously as a consequence of, I mean, a direct consequence of this general hardening that we're seeing throughout the industry.

Which takes me to my next bit of news, is Microsoft has joined the group of corporations who are visibly tightening their security. Just yesterday the Washington Post carried the story saying that: "Microsoft is moving toward a major new effort to encrypt its Internet traffic amid fears that the National Security Agency may have broken into its global communications links, said people familiar with the emerging plans. Suspicions at Microsoft, while building for several months, sharpened [last month] in October when it was reported that the NSA was intercepting traffic inside the private networks of Google and Yahoo!," which of course we've covered extensively, "two industry rivals with similar global infrastructures. They said top Microsoft executives are meeting this week to decide what encryption initiatives to deploy and how quickly."

And as we'll remember because we've talked about it here, we did see signs of this in the various slides that Snowden caused to be released. We know, we saw mentioned that Hotmail's address books were being collected. There were signs of a Hotmail message referenced in one of the slides. And apparently also Windows Live Messenger was one of the social networking technologies that the NSA were saying that they had access to. So we don't have any timetable yet, but Microsoft has decided, whoops, we need to follow along.

Leo: Of course Google's doing it already; right? I mean...

Steve: Yes. Google, well, Google, they're now working to encrypt their internal links. I think everybody is scrambling to do that. It's not an easy thing to do. But Leo, bring up this chart in the next link here, the EFF? They have produced a really nice summary of who's encrypting what: www.eff.org/deeplinks/2013/11/encrypt-web-report-whos-doing-what.

Leo: This is great.

Steve: Yes, anyone can find it, if you google the phrase "encrypt the web report." So the EFF calls it their "Encrypt the Web Report." And they are maintaining it. They've had two updates so far as of this podcast. And it's a really nice visual grid showing what types of encryption are being done and by whom.

Leo: You use Level 3, don't you.

Steve: Yeah, they're my - I'm in a Level 3 datacenter, yup. They are. And they have been implicated in - essentially they're implicated because they are the largest top-level Internet major bulk traffic carrier. But anyway, so what's interesting to me, for example, I don't know why, but for example in this grid the EFF has a lot of "undetermined" under the forward secrecy column. They show "encrypts data center links," so who does that; who supports HTTPS, meaning any encryption at all; who supports the strict transport secrecy or security, HSTS; who supports forward secrecy; and, for email, who supports STARTTLS.

And what I don't understand is why they've got so many "undetermined" under forward secrecy because it's trivial to determine that. I mean, you could just put the various websites into your browser, watch the protocol, or any of the SSL, like the SSL Labs checking website, and you'll immediately determine whether forward secrecy is supported or not. So that column should easily be filled in with either yeas or nays all across. But anyway, so it shows Amazon, Apple, AT&T, Comcast, Dropbox, Facebook, Foursquare, Google, LinkedIn, Microsoft, MySpace, Sonic.net...

Leo: I think that the reason that they say "undetermined" is because these companies haven't responded to the survey. They're doing this based on asking the companies.

Steve: Ah, okay.

Leo: We've asked the companies what they are doing. And so if the company doesn't respond, they don't know. So they're not doing any research on their own. They're just - this is responses to their survey, EFF's survey.

Steve: So better than nothing.

Leo: Yeah. Yeah, I mean, I think now that we know Level 3 provided backdoor links, or at least supposedly provided backdoor links to Google and Yahoo! to the feds, and Google and Yahoo! are encrypting datacenter links, I imagine everybody will do the same. And if they don't, well, that tells us who do.

Steve: Well, and this is why I was so bullish from day one on the Snowden leaks. It's like, okay, this is not good for the NSA, but we need to know what's going on. And we're seeing the upshot.

Leo: I mean, you don't have - so the links, we've talked about this before, the datacenter links are database replication, things like that. No company like Google, of Google's massive size, is in a single datacenter. So you have to replicate your databases from your datacenter in Seattle to your datacenter in Dallas to your datacenter in Singapore. And those transports go across the, well, I don't want to say the public Internet. They go across leased lines from companies like Level 3. And if the NSA says to Level 3, hey, let's just put a little tap in here, a little fiber optic splitter, oh, and by the way, you can't tell anyone we're doing this, they're going to get everything. You figured this out, I want to give you credit, long before this came out. You said they must be upstreaming.

Steve: This was the way to do it, yes, exactly. If, I mean, and this further demonstrates, I think, that those companies were telling the truth. I mean, there's still this unknown about some of the language in some of the slides which implies that the companies were working, were knowingly working with the NSA. I don't think we are ever going to know definitively what was actually going on. But from the outcry from these companies - and remember we've also talked about some of the reactions that the employees have had. I mean, they were furious when they learned that this was going on. I mean, they were using words we can't say on the podcast. So, yeah. And as far as I know, Google is not yet encrypting their inter-datacenter links. They are on it.

Leo: Oh, I thought they were.

Steve: No. No, because that's a big deal. I mean, they're absolutely working on it hard, as we reported, I think it was last week. But it's going to take a little bit of doing because it's a big problem.

Leo: I love it. And this is always the case, that our local Internet service provider, Sonic.net, my good friend Dane Jasper runs that...

Steve: Yeah. Green all the way.

Leo: Green all the way. And also SpiderOak, which you've recommended before as a Dropbox alternative, green all the way across.

Steve: Yes, yes.

Leo: Yeah, that's good.

Steve: And, but on the other hand, so is Dropbox.

Leo: Yeah, good for Dropbox. Late to the game, but they did it 100%. Of course they still have the keys to your data, so...

Steve: Yeah. They are not TNO. And that's the other thing I wish. Oh. Wouldn't it be nice...

Leo: Where's the TNO?

Steve: I know. Wouldn't that be nice, to have a TNO column there. But no.

Leo: Would be nice.

Steve: That would be nice, yeah.

Leo: That's what people have to listen to this show for.

Steve: I think at some point I'm going to have to revisit the cloud storage. I mean, it's a huge amount of work, frankly, to pull all that together, as I did once when we did the big cloud storage provider podcast. And people are now asking me constantly about this or that instant messaging because, I mean, exactly as we predicted, the upshot of these privacy revelations is a huge influx of new secure services that are being offered. So it's like, uh, okay. And people are saying, well, what about this one? What about this one? I mean, the problem is it takes serious work to determine exactly what people are doing. And when you've got someone like BitTorrent not telling you what their protocol is, although there is an open source BitTorrent client...

Leo: For BitTorrent Sync?

Steve: ...for BitTorrent Sync, yes.

Leo: Ah.

Steve: And so that's going to give us - they're reverse-engineering the protocol, and so the open sourceness of the BitTorrent Sync client will give us a wedge into how Sync is working. I don't think they have it done yet, but they're on the way.

Okay. So, patent trolls. We've talked about patents...

Leo: This made me so sad. This is the Newegg trial?

Steve: Yes.

Leo: They had Whit Diffie coming in, saying, "I invented public key cryptography."

Steve: Oh, well, actually, I sent a link when Ars reported that. And I said to my Twitter followers, read down at least until - down to. Anyway, so here's the background. We've got this guy, Erich Spangenberg, who is - he sets up shell corporations. He's got nine of them that he owns and apparently a total of 22 of them owned by family members. And so this one is known as TQP Development. And it owns the rights to one patent, which has long since expired, by the way. It's no longer even a valid patent. So the technology it protects, which was for - it was for modems, back in the modem days, a means of cryptographically protecting the data that modems were exchanging.

And so what's sad here is all of the evidence viewed by somebody, I mean, like viewed by the industry, that knows what's going on, is that the patent was never valid, that what it - the rights that it had granted were already in use and in the public, well, they were already actively in use, thus constituting prior art. And prior art renders a patent invalid. I mean, it's prior art technology cannot be patented. It's why I immediately published the fundamental, and I have continued to publish, all of the protocols that SQRL is using because the act of publishing it renders it unpatentable. It's now in the public domain. Nobody can have it. Which is the way something like this should be. Okay.

So this company, this Erich Spangenberg with his TQP Development, purchased the patent from its originator for three quarters of a million dollars. So, okay, that's a chunk of money. He has then gone around and sued nearly 140 different companies, generating a total of $45,370,000 because the companies have capitulated rather than challenging the patent. His claim is that this patent covers the combination of using the RC4 cipher with SSL. And as we know, until recently, when RC4 has fallen into disfavor, many companies were using RC4 and SSL. So even though this patent is expired, his suit alleges that Internet commerce, that the inventor foresaw the whole future of Internet commerce, back in the - this was a Rockwell modem that this thing was - it was a way of retrofitting firmware in a Rockwell modem to just create a secure point-to-point link. Nothing to do with the Internet. Nothing to do with packet-switching technology, I mean, nothing to do with commerce at all. So he's saying that any large Internet commerce companies were infringing the patent before it expired, and therefore owe them money.

A mutual fund, Dodge & Cox, was sued and paid a little over $25,000. The Pentagon Credit Union paid $65,000. QVC paid $75,000. MLB Advanced Media paid $85,000. PetSmart paid $150,000. PMC, $400,000. Cigna paid $425,000. Bank of America, $450,000. First National, $450,000. Visa paid half a million dollars. Amazon paid this guy half a million dollars. UPS, $525,000. IBM, three quarters of a million dollars. Allianz Insurance, $950,000. And Microsoft paid them a million dollars.

Leo: Notice that all these sums are less than the estimated million and a half it cost to fight it. And that's one of the reasons patent trolling works. These companies, they just made a simple business decision. Well, even a million dollars is less than it would cost to defend it. And I'm thrilled that Newegg, despite the fact that it's cost them millions, and now they have lost for another couple of million, decided to fight it. But you understand why it's a bad business decision.

Steve: Yes. It exactly is. The good news is they have a T-shirt we can buy to help support their cause. But to give you an idea, so now how does this work for the patent troll? They're called patent trolls, first of all, because they're not using the patent.

Leo: Non-practicing entities is the...

Steve: Yes, exactly.

Leo: ...more official term, I guess.

Steve: Yes. This guy, this Spangenberg, bought the intellectual property rights simply to have grounds to sue, not because he was using the patent and wanted exclusive rights to what it was protecting. I mean, it's obsolete, has been for a long time. And there were many other ways to skin this cat than using RC4. So the deal with the original inventor, the inventor gets - get this, Leo - 2.5% of any money recovered, plus $350 an hour as a consulting fee. So far, the inventor has made $588,000, while Spangenberg keeps the rest.

Leo: More than $40 million that he's keeping, that's his.

Steve: Yes, yes, exactly.

Leo: So now you know why people do this.

Steve: Exactly.

Leo: You could swallow a lot of ethics for $40 million.

Steve: So Newegg says no. Newegg has been sued before, and has always said no, they won't do this. So the drama here, which occurred just recently, is when expert witnesses whose job it was to explain the technology to the jury - this was a jury trial as opposed to just explaining this to a judge - they took the stand. First we have Ron Rivest, the "R" of RSA, who testifies via a videotaped deposition about how he invented the RC4 cipher while he was at RSA Security in 1987, two years prior to the TQP patent application. So then we get former Microsoft CTO, the Chief Technology Officer, Ray Ozzie, who described demonstrating Lotus Notes to Bill Gates in '88. And Lotus Notes used the same technology.

Alan Eldridge, who worked on the Notes product, flew down to Marshall, Texas - this is where this was happening, in East Texas - in person, and wasn't paid, because he felt he was doing his civic duty to keep this travesty from happening. He flew down to describe how he put Rivest's RC4 cipher in the Lotus Notes software. So it was in a product in use, practically, and in commerce, all of which invalidates it being the subject of a patent.

Okay. Finally, on Friday of last week, Newegg's star witness, the person we talk about often, Whitfield Diffie of Diffie-Hellman fame, the Diffie-Hellman key agreement, the key cryptographer takes the stand. Diffie's goal was to knock out the so-called "Jones patent" because this was some guy named Jones was the guy who added this. And I have to say, I've looked at the patent, and I mean, it is nice technology. It's not junk. It is honest-to-goodness really good encryption for a point-to-point connection between two modems. It's good. But it wasn't first. And that's the key. And what it patented and what they're suing everybody over was already in use before.

So, and Whit Diffie, there's a picture, I don't think - there's a link further down, Leo, to the Ars Technica story, I think it's the "Newegg trial crypto legend Diffie takes the stand."

Leo: Yeah, Whit looks like a mad wizard, I think.

Steve: He does.

Leo: You shall not pass.

Steve: If you put Endeldorf's or whatever his name is cap on his head, it's totally convincing.

Leo: Endeldorf? Gandalf.

Steve: Gandalf. Oh, okay. I don't know what I'm talking about.

Leo: I like Endeldorf, though. We'll have to create a character. He's the guy, Whit Diffie's the guy on the left, by the way. The lawyer's the guy on the right.

Steve: Oh, yeah. You can tell who's the attorney and who invented public key crypto.

Leo: But, now, I have to tell you, in Marshall, Texas, that is a little bit of a strike against him. He looks like a hippie.

Steve: Well, he looks like an eccentric genius.

Leo: What I've been told about these juries, and one of the reasons these companies pursue this in East Texas, they're very conservative juries.

Steve: Yes. Yes.

Leo: And they really want to help the little guy against the big companies. That's really where this is coming from. So they feel like the little guy is getting ripped off, you know.

Steve: Yup. So the attorney, Albright, for Newegg, says: "We've heard a good bit in this courtroom" - I'm quoting from the transcript - "a good bit in this courtroom about public key encryption," says Albright. "Are you familiar with that?" And Diffie says, "Yes, I am," in what surely qualifies as the biggest understatement of the trial. And then Albright says, "And how is it that you're familiar with public key encryption?" To which Diffie replies, "I invented it."

Leo: Yeah, that's good.

Steve: And then, I mean, you would just think, it was like, okay, this is done. So then the plaintiff attorney gets up, Mark Fenster, who's the lawyer for TQP, and says to Whit Diffie, "You never completed a master's degree, did you." And Diffie says, "That's correct." "Other than the honorary degree, you don't have an earned doctorate or PhD; correct?" And Diffie says, "That is correct." And even though he taught a few courses, "You never had a real professorship; correct?" asked Fenster. And Diffie says, "I never had a full-time academic job, no." So, and then Fenster of course notes that although Diffie was testifying in court for the first time, he had other expert witness work lined up. His rate varies from $500 to $600 per hour, and it's $700 for testifying in court.

And Newegg lost. And this, Leo, is why I've stopped agreeing to be an expert witness. I did that for a while, years ago. And it was this kind of event. I would only testify if I was on the side that, like, should win, because that's me. And the most annoying lawsuit that I was involved in, I testified on behalf of NEC, who had the famous MultiSync display. And they were being sued by Princeton Graphic Systems because NEC's advertising was saying this is the last monitor you'll ever need to buy because, because of the MultiSync-ness of it, it could handle whatever different resolution you gave it, which was phenomenal at the time. Now we just sort of take it for granted. Back then, that was a big deal. So Princeton Graphic Systems was suing NEC for their statement because the PS/2 had just come out, and you didn't need to buy a new monitor. You could use the old MultiSync that essentially just worked because it was smarter.

And so I very carefully tried to explain to the judge - this was not a jury trial. This was just me and the judge, who had a green oxygen tank next to him, and he remembered when horses pulled carriages. So I was trying to explain to this guy why the way this worked meant that NEC's ads were correct. And they lost. And I just was like, okay, I'm not doing this anymore. This is just too annoying, the fact that the system is that broken.

So in this - back to Newegg's trial. The plaintiff was claiming $5.1 million in damages and who knows what. I didn't look at the detail of the suit, so it may have been damages and other forms of upset. They were awarded by the jury 2.3, so a little less than half. But still, $2.3 million. But as I said, Newegg has lost before, and they have always won on appeal. So they are going to appeal this.

Leo: Because that moves it to a different venue, moves it out of Marshall, Texas.

Steve: Exactly. And then saner heads prevail. But anyway, I liked this. I wanted to share the details of this because we've talked about patents and patent trolls, and here's a - it's a classic example. I mean, you couldn't - basically this brought the industry's top gurus out of hiding. Diffie's never testified in court before. This was his first testimony as an expert witness about what he did and when. And it didn't matter. So I think it will because, yes, I think...

Leo: Yeah, I'm glad they're fighting, yeah.

Steve: Oh, absolutely. And it is the case that at Newegg they have a T-shirt you can get which is neat.

Leo: They don't have Extra-Large, though. They only have skinny sizes.

Steve: Ah.

Leo: Whoops. They need more - maybe they sold out already.

Steve: I bet you - I was just - you took the words out of my mouth. I bet the big ones are sold out, yes.

Leo: Yes.

Steve: So over the weekend - over the weekend? I guess it was. No, this week. Where are we? This is Wednesday. I guess it was over the weekend, late in the weekend, and at the beginning of the week, it occurred to me that we've been getting a huge number of questions about CryptoLocker. Will it affect a drive that has, like, what level of drive mapping? If it has a drive letter? What if it's available, but it's not mapped? Does Sandboxie in fact protect us? What kind of a virtual machine do I need, blah blah blah, I mean, there's a huge amount of anxiety because it's a huge problem. And I finally decided that we have a super-savvy audience of listeners. People would probably like to play with it.

So first I put it up in a non-public directory on GRC and announced its availability through Twitter and then sent people back links. And there was a lot of interest there. So I decided to formalize this and to make an old version and a new version available on GRC. So if anyone wants to experiment with CryptoLocker, I mean, this is deadly. This is not neutered, or it's not - it hasn't had its fangs removed. This is the live CryptoLocker malware, both an early version, which is highly detected by existing antimalware software, and the most recent one, which is not yet very well detected. I think it's 7 out of 47 antimalware that virus total tests detect it, but the balance don't.

So anyway, GRC.com/malware. That will take you to a page where I explain what's going on, and the dangers. And I have in text, not clickable links, the location of three different ZIP files because I also threw in the banking trojan, Zbot, or Zeus, that we've talked about often, which is a rootkit trojan that I thought people might want to experiment with also. They are in encrypted ZIP files, so you must use a password in order to decrypt the ZIP file, in order to get access to it. My only, I mean, I recognize this is a mixed blessing. This is dangerous. But based on the feedback I got through Twitter, I know that a chunk of the listenership of this podcast would love to set up a Sandboxie.

Already Jason, who tweets from @aliencg, and there's a link to his report from the weekend [www.aliencg.com/journal/2013/11/24/CryptoLocker], he played with it a lot. And I got a whole bunch of other feedback from people who were enjoying the opportunity. Many people wanted to verify that their antimalware would detect it. The good news is, for example, Microsoft Security Essentials detects this. It doesn't detect it in the ZIP because the ZIP encryption is very good. As we know, something that is well encrypted is pseudorandom noise. There is nothing to lock onto in an encrypted ZIP, which is why this is one of the ways it's being distributed. But the second it emerges from the ZIP, if you've got real-time monitoring on in Microsoft Security Essentials, it just nails it.

I was playing with it myself on a completely isolated computer in order to create these ZIP files and to get their SHA-256 hashes and so forth to put this page together. So I was pleased to see that it is being caught immediately. So you'll need to turn off those defenses if you want to watch it go and see it do its stuff. But I thought it was, on balance, more useful to let people verify their defenses and also experiment with containment. As I was mentioning, Jason verified that Sandboxie does indeed protect from CryptoLocker. What he found was that encrypted copies of the files that existed on his system were appearing inside the sandbox, exactly as we would predict. So any time Sandboxie detected that a right was trying to be made, it essentially created it in the sandbox so that CryptoLocker saw the encrypted file, but nothing outside was affected. And when you deleted the sandbox, you completely deleted all of the encrypted files.

So anyway, GRC.com/malware. If I get in trouble from search engines for having that there, as some people have cautioned me I might, then I may have to take it down. But I hope - I don't think I will because they are not - there are no active links on the page. You have to manually copy and paste and then remove spaces that I've put on either side of the forward slashes. And only then do you get a completely safe to download ZIP. And that you need to use a password to decrypt the contents inside.

Oh, and I say it on the page, but be sure to delete the decrypted executable once you're through messing with it. Do not leave it around. I did not change the name of the EXE because executables can check their own name. And I wanted to leave it as it was received in case the executable did check to see whether its own name had been renamed to something like horrible-cryptolocker-virus-do-not-run, that kind of thing. Which I would have liked to do, but it might have changed its behavior, so I decided not to.

There is a new version of CryptoPrevent over at the FoolishIT.com site. I linked to it in the show notes, if anyone's interested: www.foolishit.com/vb6-projects/cryptoprevent. Or you can probably just put CryptoPrevent into Google, and it will find it there. And interestingly, the file format has been reverse-engineered now, that is, the format of the encrypted files. One of the things that many people have asked is, if you ran CryptoLocker on an already encrypted file, would it double encrypt it? Would it reencrypt it? And it looks like the authors have been careful to prevent that.

The other thing we heard, remember that there was that service they were offering, "service," unquote, where you could pay them more, and you uploaded a copy, or you uploaded one of the encrypted files, and it would then provide you with what you needed in order to decrypt it. Or I guess maybe it would decrypt it itself. I don't remember exactly how it worked. But the fact that they wanted a copy of the file told us that there was a header of some sort on the file which it could use, that is, it wasn't just the file encrypted, otherwise it couldn't do anything with it.

And sure enough, this has now been reverse-engineered. There is a 20-byte cookie in the form of an SHA-1 hash, a 160-bit hash, on the front of the file, followed by that file's AES key encrypted with 2048-bit RSA. So again, this is very good cryptography, unfortunately, which CryptoLocker has employed.

Leo: I think the patent trolls should sue them over that. That's terrible. That's an infringement of TQP's patent. Go after them. Go, boy. Sic 'em.

Steve: That's a good way to spend their [indiscernible], yes.

Leo: Well, we know they're deep - these guys, on the other hand, have deep pockets; right? Making lots of money on CryptoLocker.

Steve: So a pseudorandom key is generated per file, which I didn't realize, but that's, again, that's the way you'd want to do it. A pseudorandom key is generated per file. That key is encrypted using the public key. And that's stuck that the front of the file. So the encrypted files will all grow by, I don't know, like about 256, maybe 266 or so bytes. And that header is crucial for decrypting the file.

So anyway, the point of this is there is now a third-party decrypter. Someone named Kyrus, K-y-r-u-s, has reverse-engineered the CryptoLocker malware to determine how the file format works and has built an open source decryption engine. Now, you still need to go get the key, and you have to pay for it first. But we've heard reports of, for example, the decryption process crashing before it's finished. And we were joking that they weren't as worried about creating a bulletproof decrypter as they were an encrypter.

So this Kyrus-Tech.com does have a free open source decrypter which also fully documents the file format and shows the reverse engineering of the encryption process to create decryption. So if anyone only got a partial decryption of their documents, this would probably handle the balance very well, I would think. And I noted also, since the podcast, the latest versions of CryptoLocker have decreased the ransom to half a bitcoin in response to the...

Leo: Still going up, though.

Steve: Yes, the massively crazy price of a single bitcoin. And I don't know if we talked about it on the air or not. But for the first time today, Leo, it went north of a thousand dollars per bitcoin.

Leo: Why is that, do you think?

Steve: I don't know. I think it's legitimacy. I think it's - only thing I can think is that it is beginning to acquire legitimacy. Because people selling bitcoins would tend to drive the price down. People buying them would tend to drive the price up. And so it must be that people are believing that there is a long-term future, and so they're moving cash into bitcoin, looking at - purely speculatively.

Leo: Yeah. So it could be a bubble.

Steve: Yeah. Although, if you look at it over the last month, it has spiked. And then there has been a drop-off as people have been liquidating their bitcoin assets, driving the price down. But it continues to recover. And I think we're about two days away from a difficulty increase. The whole system is going to reevaluate itself and reset the difficulty level for this.

Leo: So now is the time to get your bitcoin miner fired up.

Steve: But really, Leo, really, think of all of the people who have some serious money because they were mining early, and with any luck they were holding onto their bitcoins. Oh, my goodness.

Leo: You saw the guy whose hard drive had 7,500 bitcoins on it, and he threw it away, and now he's going through the landfill?

Steve: It's a landfill. And the guys at the landfill think, well, it's probably three or four feet down based on when you think it was. And it's like, oh, good.

Leo: Four feet of rubbish. But it's worth, what, three quarters of a million dollars. No, more than that. 7,500 bitcoins, wow. That's $7.5 million; right?

Steve: Ooh, yes.

Leo: I'd look through some landfill for 7.5 million.

Steve: Oh, ouch.

Leo: Wow.

Steve: And did you hear the story? His girlfriend was complaining that the laptop was making too much noise.

[Talking simultaneously]

Leo: 7,500 bitcoins created in 2009.

Steve: Yeah. So back in the early days, when a laptop running by itself had a chance. And remember, I just had my one machine, and like on the third day I came out and said, oh, look, there's 50 it made for me. And of course I reported it on the podcast. Look, Leo, I got 50 bitcoins. They were worth 450 bucks at that point.

Leo: $50,000 they're worth now.

Steve: Wow.

Leo: Wow. Anybody wants to give me bitcoins, I'll take them. We have a bitcoin donation QR code on our front page. I have seven. People have donated seven bitcoins. That's good. That's good money. I'll take it. Seven grand.

Steve: A total of, yes.

Leo: The problem with all of this stuff is you don't know when to see it. Is it at its peak? Is it just beginning? Are you going to be like the guy who bought the pizza pie for what is now worth several million dollars in bitcoins? Or are you going to be the guy who rides it all the way down?

Steve: I think it's a legitimate currency. I think that's what we're seeing. I think, I mean, as I've said before, these radical fluctuations are just because...

Leo: It's very volatile.

Steve: ...it's so young. It's very young. Nothing has really established the price. It just needs more inertia behind it. So, yeah, wow. Fun.

Now, our old friends at Pogoplug - we were using Pogoplug for quite a while - have a new product which is kind of cool. It's called Safeplug, and it is $49, and it's a little box. It's an appliance. It's TOR in a box.

Leo: Oh, that's interesting.

Steve: Isn't that cool? GigaOM carries the story. I looked for it over on Pogoplug's site, and all they wanted to do was get me to log in, and they didn't have any other information that was visible. But I know a few things about it from the GigaOM story. It's $49. It is a little appliance, like a little - one of these like a little Apple TV box or a little random Internet appliance. So you plug it into your router. And essentially it's got Linux running in it, and Tor, all preconfigured and ready to go. So you then - you run your network through it, then to your router, and you're anonymized.

Well, and remember, we need a - that's why I'm a little concerned about them overselling this, because remember that anonymity requires more than just bouncing through Tor because, for example, cookies will deanonymize you. And you need to have cookies in order to maintain persistent state with a site. So you have to...

Leo: You know what would be interesting is if each of these Safeplugs were a Tor node itself.

Steve: Actually, they are. You can also set them up so that they are a Tor node.

Leo: Because the risk with Tor is the feds co-opting a Tor node.

Steve: Yes.

Leo: An entrance or an exit point particularly.

Steve: Yes.

Leo: But you've got the entrance point. It's yours. And if it's a random set of - if they sold a million of these Safeplugs...

Steve: I know. It would massively expand the Tor network and make it so diffuse that it would no longer be feasible to watch exit nodes. They'd all become exit nodes.

Leo: Right. Yeah, just pick a random Safeplug as an exit node.

Steve: Yup. I like the idea. So in order to solve the problem of some sites balking at using Tor, for example, banks often tie their authentication to your IP address. So if you suddenly appear to be coming from somewhere, some other country than you're known to be in, the bank will say, uh, no. So there is software that they include that allows you to whitelist based on URLs. So, for example, you're able to not - you're automatically, your traffic will automatically not go through Tor if you're going to any of your whitelisted sites. And what's very cool is you can whitelist by browser. They use the user agent...

Leo: Oh, that's a good idea.

Steve: ...the user agent header to either route through Tor or not. So you could set up Firefox, for example, to be your Tor-based browser, and Chrome would be direct high-speed access. Because that's the other thing is running through Tor does slow things down. You can't stream video conveniently and so forth because it just - there is a lot of temporal overhead associated with bouncing from one node to the other, as we've discussed before, and all the crypto work that's being done on the fly. So anyway, Safeplug, 50 bucks from the Pogoplug people.

Leo: It also has ad-blocking software built in.

Steve: Yup. Yup.

Leo: Which it would have to, right, because that could be a way of tracking you.

Steve: Yeah. I mean, I would say you want to use a clean browser. You'd want to run

your browser in Firefox...

Leo: Incognito mode, yeah.

Steve: Exactly, in incognito mode, with a fresh instance. And it's going to flush its stuff away and so forth so that you don't - if, you know, depending on what level of anonymity you want. But I think it's very cool.

Leo: Neat.

Steve: Now, I don't know, Leo, how this escaped me. But here's the warning to our listeners who like science fiction. Give yourself three hours. Do not watch the first episode, there's only three so far, in any situation where you do not have time because you will not have any choice. I am stunned by the quality of this. It's got an 8.6 rating on IMDB. Nothing is 8.6 on IMDB. It's Fox's new series, "Almost Human."

Leo: Really. I thought it was for kids.

Steve: It is fabulous. No.

Leo: No, not for kids.

Steve: It's fabulous. It's set in the near future, the year 2048, which of course is nice because that's a nice power of two. So, and it's not feasible that we're going to have androids in 2048, but they do. But oh, my goodness. All you have to do is watch the first one. If you watch the first one, you will have no choice. It is really good. So it's - I don't know how they're going to keep it up. In fact, the reviews I've read of the first one, the reviewers were saying, if they continue this - it's the guy - JJ Abrams was involved, but it's not JJ. It's someone else who's, like, tied to JJ. I didn't do the research to figure it out. But, I mean, it's a win.

So three episodes have been aired. It airs Monday nights on Fox. The three episodes, as you would imagine, are well available on the Internet. You can use the Fox app in order to view them. It's all over the BitTorrent seeding, so it's easy to find these. And, wow. The first one will - you will absolutely be addicted. I'm just - I watched all three last night, deliberately. I wanted to know whether it could be as good as I was reading, and I wanted to know for the podcast. And it kept me up too late because I was like, okay, I just can't stop. So the good news is you're limited to a three-hour binge at this point, and then I think it'll be on everyone's must-watch list. It's just - it's really good. And I'm not going to give any more away except that it's a police procedural with a human and an android, but it's just done really well. The writing is great. It's very clever. I love it.

And while I'm on the topic, in Miscellany, I'll just say that I've heard from other people who are glad that I mentioned the Showtime series "Masters of Sex," which is a somewhat, apparently, accurate story of William Masters and Virginia Johnson, the Masters & Johnson story. But it's turning out to be a really well-written, worthwhile series on Showtime. So I'll recommend that, too.

And Leo, following your recommendation of using the @replies, I did some more research because I wanted to understand what was going on. And I found a man, a so-called "man page." We all know what...

Leo: Which is terrible [laughing].

Steve: Which?

Leo: There's a man page for how to use Twitter.

Steve: Yes, yes, a man page for Twitter. I created a bit.ly link, bit.ly/tweetfmt, so all lowercase, t-w-e-e-t dot - I'm sorry. Bit.ly slash t-w-e-e-t-f-m-t, tweet format. And I found it useful. It's very concise and condensed. And actually I tweeted it, and I got a bunch of replies saying, hey, there's some stuff here I never knew about. So there is some cool stuff there.

Leo: That's cool, yeah.

Steve: And that's our podcast.

Leo: It says a man page. Because that makes sense to people like you. Everybody else looks at it, goes what the hell [laughing]. Steve Gibson is at GRC.com. That's where he puts all his stuff, all his hard work including SpinRite, his bread and butter. If you want to get a copy of SpinRite, I highly recommend it for anybody who uses hard drives. GRC.com. He also has the podcast. There's 16Kb audio, smallest audio version made available, as well as transcriptions in human-readable and human-writeable text at GRC.com.

If you do have a question for Steve, he does questions and answers whenever time allows at GRC.com/feedback. We also have high-quality audio and video at our site, TWiT.tv/sn or wherever your favorite netcasts are aggregated. Check it out at iTunes and places like that. We do the show every Wednesday, 11:00 a.m. Pacific, 2:00 p.m. Eastern time, 19:00 UTC on TWiT.tv. You can watch live. We will be moving come January 8th to - actually to January 7th, to Tuesday, right, at 11:00 a.m. Pacific. 2:00 p.m. Eastern time. So that's the first week of January, our new time. And I think that's about it. I thank you, Steve Gibson, for being here.

Steve: Elaine is going to be digesting turkey, so she said that her transcript may be a day late. Because I want to put all these links up and make them available, as soon as I have access to the audio here in a few hours, I will make a point of getting the entry for the Security Now! page for this podcast, 432, up on GRC, with the return of the show notes as part of our weekly offering so that people can scroll through and follow the links that we provide.

Leo: Excellent. And a correction, it is 1:00 p.m. Pacific, 4:00 p.m. Eastern on Tuesday, after MacBreak Weekly. I got that wrong. 1:00 p.m. Pacific, 21:00 UTC. Starting in 2014. Thanks, Steve. We'll see you next time on Security Now!.

Steve: Thanks, Leo.