Bitmessage

Description: After catching up with a lot of interesting security news, Steve and Leo examine the operation and technology of the new Bitmessage secure and anonymous Internet messaging system.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-420.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-420-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Today he'll talk about Bitmessage. Is it as secure as it's supposed to be? He also talks about the USA and Brazil, another embarrassment. Oracle, doing it all over again. And he has a revelation to make about a new product he's going to be working on sometime soon. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 420, recorded September 4th, 2013: Bitmessage.

It's time for Security Now!, the show where we cover your security, your privacy, all the good stuff that you need to know when you get online. And there's nobody better to do it than our Explainer in Chief and coffee addict pro tem - no, that's wrong - Mr. Steven Gibson of GRC.com. He's the creator of SpinRite, the guy who first discovered spyware, coined the term, wrote the first antispyware. He's also the guy who told Microsoft, "You're crazy," when they did that raw sockets thing, and nobody believed him. Everybody dissed him. Microsoft mocked him until it became a big problem and they had to take it out. I could go on and on. This is a guy, he's been - he's been in the lead, the charge all along, and you've got to listen to him: Steve Gibson.

Steve Gibson: I think, Leo, we may be able to add before long that I came up with the solution for user login and web authentication.

Leo: This would be huge.

Steve: It would be huge.

Leo: This would be huge.

Steve: And I think I've got it.

Leo: All right.

Steve: And we'll talk about that later.

Leo: Did you have this inspiration in the shower?

Steve: No, I was having coffee and reading my morning stuff when I was having some breakfast. And it just kind of, I mean, my total focus is on SpinRite and working on the next release of SpinRite. And this just came unbidden, like. And I went, oh.

Leo: Sometimes that happens.

Steve: And then I thought, wait a minute. And then the more I thought of it, it's been six days now, and I - and I wanted to call it HIPS, which would be an acronym for Hiding In Plain Sight. Because, I mean, it is so simple.

Leo: I like it. I like that acronym.

Steve: And so obvious.

Leo: Yeah. You're like Archimedes in his bathtub. Eureka!

Steve: And as we were saying before, many times when, the way the human brain works, you focus on a problem, you really think about it, and then you just put it away, and other parts of your brain have been engaged. And so when our listeners learn of this, it'll be like, oh, uh, yeah.

Leo: Yeah.

Steve: So, yeah.

Leo: Yeah.

Steve: And so I can't - it's so much better than anything we have now, I can't imagine...

Leo: You're such a tease. All right, stop teasing us.

Steve: ...that it won't - yes, okay.

Leo: Let's get to the security news. You can tease us later. By the way, what is this show about?

Steve: Bitmessage.

Leo: Okay.

Steve: Yes. I want to cover Bitmessage to sort of get it out of the way. It's got so many problems that it probably isn't going to survive, at least in its current form. It could mutate. But many of the people that have successfully attacked it for its shortcomings have also mentioned they're working on something of their own. So, and I appreciate them disclosing that. But so this is an example of what I have expected we would see post-NSA and Snowden revelation, which is a refocusing on this kind of stuff. I think I read that Tor traffic is up 500%. It's like five times more use of Tor.

Leo: Good, good.

Steve: So Bitmessage, everyone wants to know what it is. So this is - I didn't bother to spend the time to dissect the protocol at that level because I don't think it's - I think it's a waste of time. But I want to explain - but my subtitle is "An idea worth learning from." And so the concepts are interesting. And so we're going to talk about it so that everyone's curiosity will be satisfied. You'll sort of know what it is. You could use it if you wanted to. But it's probably better to wait for 2.0, or maybe even 1.0. I think it's at 0.3.5 right now.

Leo: That's usually a bad place to start, yeah.

Steve: Or wait for the one that succeeds it that, like, solves some of the fundamental problems. So it's some interesting ideas that are definitely worth sharing. And we've got a bunch of news this week, as well.

Leo: Very exciting. Very exciting. Leo Laporte, Steve Gibson, Security Now! on the air. And I like the headline of your first story. It says, "NSA and the USA in the D-o-g-h-o-u-s-e." What's going on?

Steve: Yeah. Well, there continues to be international fallout from the incremental revelations as the Snowden documents continue to come out. And I first learned of this from a tweet from a Brian S., is his name. And he sent to @SGgrc, "NSA fallout: Brazil's government seeks to create new surveillance-proof email system. Aim: an alternative to

Gmail." And he sent me a bit.ly link [bit.ly/1e6DJYn] which took me to the article in Portuguese which described this. And a different friend of mine sent also that, but said - he noted that the Google Translate does a pretty good job.

Portuguese: www1.folha.uol.com.br/fsp/mundo/127105-governo-brasileiro-quer-e-mail-nacional-contra-bisbilhotice.shtml

Google Translate: translate.google.com/translate?sl=auto&tl=

en&js=n&prev=_t&hl=en&ie=UTF-8&u=http%3A%2F%2Fwww1.folha.uol.com.br

%2Ffsp%2Fmundo%2F127105-governo-brasileiro-quer-e-mail-nacional-contra-bisbilhotice.shtml

So looking at that and sort of pulling out keywords because it's still a machine translation, what it looks like is that the Brazilian government, in the wake of the Snowden NSA revelations, ordered their domestic postal system to expand an existing email facility which was previously targeted mostly for business into a national competitor to Hotmail and Gmail, specifically to avoid the problem of U.S. NSA surveillance. So this is them saying, gee, now that the emperor's been shown to have no clothes at all, we need to do something about this. So they're just uncomfortable with using domestic Hotmail and Gmail, U.S. services. And I just shake my head. I mean, it's certainly sad that this has happened. But that's the consequence.

Secondly, the Indian government, this was a headline - this is covered by TheHackerNews.com. So the Indian government may ban U.S. email services for official communications. And I'll just share the top of the story. They said: "The Indian government is planning to ban the use of U.S.-based email services like Gmail for official communications to increase the security of confidential government information." Well, one could argue it should have never been unencrypted, heading off to Gmail, for important Indian government work.

But, I mean, continuing: "The recent disconcerting reports that India was being spied upon by American intelligence agencies has opened an all-new chapter in the cybersecurity space, as leaked by former U.S. National Security Agency contractor Edward Snowden, that the NSA was involved in widespread spying and surveillance activities across the globe. The government" - the Indian government - "plans to send a formal notification to about 500,000 employees across [India], asking them to stick to the official email service provided by India's National Informatics Centre," said the Times of India.

"The fact [is] that several government officers in top positions use their Gmail IDs for official communications. Several senior government officials in India, including ministers of state for communications [and a few others that are named in the article] have their Gmail IDs listed in government portals as their official email. So last week India's IT minister revealed that the new policy will enforce rules such as the use of static IP addresses, virtual private networks, and one-time passwords for accessing Indian government email services on all Indian officials who are stationed abroad. All Indian missions will use NIC servers which are directly linked to a server in India, and that will keep government information safe."

So in the past we were seeing rather lax security and not much focus and concern. But governments are responding to what has been learned about what the NSA is doing.

Leo: Of course, they're spying on us, too, so...

Steve: Yeah.

Leo: I mean...

Steve: Yeah, but they're doing it more secretly, apparently.

Leo: Right. That's the big difference.

Steve: They're able to keep it quiet.

Leo: Right. There's no Snowdens working for them.

Steve: Okay. So this next piece helped, I think, to explain a little bit about what we discussed last week with not only David Miranda being detained, but then that weird forced destruction of hard drives in the basement of the Guardian over in the U.K. It turned out that David Miranda, who was, as we know, working as a courier for the Guardian's Glenn Greenwald, shuttling documents back and forth - are you sitting down, Leo? I know you are.

Leo: I'm sitting on my ball, yes.

Steve: Just wanted to make sure.

Leo: I don't know if that's enough. Maybe I need more support.

Steve: Are you well centered? Center yourself.

Leo: Yes, okay.

Steve: He was carrying the secure password for TrueCrypt-encrypted drives on a piece of paper.

Leo: He wrote it down.

Steve: He wrote it down. Now, we've discussed, obviously many times, proper password management. And Bruce Schneier is famous for saying it is better to have a password you cannot remember and must write down than one that is easy to remember that you

don't write down because, if it's easy to remember, it's easy to brute force.

Leo: Right.

Steve: But of course the right solution is to be a little more clever. Leave off something that you...

Leo: The Walter White solution.

Steve: Yes, exactly. Leave off the ending seven characters, which you can remember, which you don't then tell anyone. And then you're mystified why the password that was written down doesn't work. It's like, well, that's what they gave me.

Leo: That's supposed to work.

Steve: I don't know why. I'm not technical. Anyway, so the Register was apparently delighted in reporting this, that Miranda was carrying a hard drive encrypted with TrueCrypt. So they said: "It had been widely reported that Miranda disclosed some passwords to the police at Heathrow under threat of jail. But many analysts had concluded that these were merely those of his social networking accounts and such."

Leo: Here's my Twitter. I'll give you my Twitter.

Steve: If you really want to log in with my Facebook account, then okay. They said: "...which it would be implausible to claim he did not know. Naturally, it was considered unlikely that he would even know the keys to any top-secret encrypted data he might be carrying. This was the view taken by security guru Bruce Schneier, for instance. But now, in a court statement made this morning and tweeted live by Telegraph correspondent David Barrett, the government says that Miranda was actually carrying a piece of paper with a decryption password written on it. This allowed the police to read at least some of the files he was carrying. These include some 58,000 highly classified U.K. intelligence documents."

Leo: Ooh.

Steve: Yes. "In the government's view, this demonstrated 'very poor information security practice' on the part of Greenwald and the Guardian. According to the Cabinet Office official making the statement..."

Leo: Snowden was probably going, "God, I told them." Oh.

Steve: I know, "...it was concern over this apparent amazingly lax security posture by the Guardian that had previously led the government to insist on destruction of any

Snowden files it held, on U.K. territory, at least." And now I have to say, who could blame them? I mean, if you're given this kind of information, you absolutely have to treat it with respect. And if they're traipsing around with encrypted drives and the password written down, not obfuscated, don't swap the first and second half, don't change anything about it, I mean, there's so many things you could do. You could use haystacks. Just add some stuff to it. And then...

Leo: But we knew, because Greenwald couldn't figure out how to use PGP, we knew he was not sophisticated.

Steve: Yeah. But this is just a crime. And, I mean, they've had WikiLeaks support and help. And you'd think that somebody would have said, "Okay, look. Here's the basics." But no. So anyway, I - you can imagine, I mean, we do know that the U.K., that part of what Snowden provided was documents the U.K. feels extremely unhappy about being disclosed. And so it's one thing for someone, for the press to have them. It's another thing for them to be flying around the world with - essentially in plaintext because that's what you have if you're using TrueCrypt and the password's tattooed to you. So, yeah.

Leo: That's really too bad, yeah.

Steve: There was, it was revealed, sort of an interesting iOS and OS X vulnerability. This is, at this point, this is a crashing problem. But we all know that that's the way exploits begin. So Boy Genius reported: "Android might be targeted by hackers and malware far more often than Apple's iOS platform, but that doesn't mean devices like the iPhone and iPad are immune to threats. A post on Russian website Habrahabr.ru" - H-a-b-r-a-h-a-b-r dot r-u is how you spell it, that's for Elaine - "draws attention to a fairly serious vulnerability that allows nefarious users to remotely crash apps on iOS 6, or even render them unusable. The vulnerability is seemingly due to a bug in Apple's CoreText font-rendering framework, and OS X Mountain Lion is affected, as well.

"According to the report, simply exposing various iOS or OS X apps to one of several possible strings of text is enough to trigger a crash. What's more, sending one such string as an SMS or an iMessage to an iPhone, iPad, iPod Touch, or Mac computer can crash Apple's Messages app repeatedly, rendering it unusable. Safari is also impacting by the bug, and naming a Wi-Fi network with one of those strings of text can cause an error while any Apple device is scanning for networks."

Leo: Oh, that's bad. Oh, that's really bad. Oh, that's ugly.

Steve: That's nasty. "The report claims that Apple has been aware of this vulnerability for six months and has yet to patch the exploit in any currently available operating system build. The author does note, however, that beta versions of iOS 7 and Mac OS X Mavericks are seemingly not affected." So for whatever reason...

Leo: Well, that's how they plan to fix it, then, because those are both coming out soon.

Steve: Yes. And what I would say, then, to people, is upgrade. This is definitely - or, if your device begins doing something really suspicious every time you drive by a certain Starbucks, then now you know why, if it crashes when it comes within range of their network, or someone's network. So I thought that was interesting.

And I picked this up also. I thought this was interesting. In what's being regarded as a historic vote, Ars Technica reported that New Zealand just banned software patents.

Leo: Yay.

Steve: Yeah. Big yay.

Leo: Yay.

Steve: So Joe Mullin reporting for Ars Technica said: "A major new patent bill, passed in a" - this was a landslide vote - "117-4 vote by New Zealand's Parliament after five years of debate" - they probably could have cut the debate somewhat shorter, given that it has that kind of majority - "has banned software patents. The reluctant clause" - I'm sorry. "The relevant clause of the patent bill actually states that a computer program is 'not an invention.'"

Leo: Good. It's not.

Steve: And people have argued that you cannot patent math, and computer programs are just math, so why can you patent computer programs?

Leo: Yeah, good question.

Steve: "Some have suggested that was a way to get around the wording of the TRIPS intellectual property treaty, which requires patents to be 'available for any inventions, whether products or processes, in all fields of technology.'" So instead they're just saying a computer program is not an invention. So therefore it doesn't have to - we don't have a problem with this next clause of available for any inventions, whether products or processes. If it's not an invention, then it doesn't matter.

"Processes may still be patentable if the computer program is merely a way of implementing a patentable process. But patent claims that cover computer programs as such will not be allowed." And that's interesting because I probably worked on one of my first patents, oh, 30 years ago? And it was - it used one of the early 4-bit micros, and it was definitely a computer program. But the attorneys I worked with said, okay, the way we do this, Steve, is we describe it as hardware. And I said, but what? It's software. He says, I know, I know. But you can't patent software. Now this, again, this was like 30 years ago.

So the way we do this is, because a patent doesn't have to be - it doesn't have to always reflect what they call the "preferred embodiment." So, but you can patent hardware. You just at that time could not simply patent software. So you'd patent the hardware

implementation, but then you would do a software embodiment of the invention, and that's sort of the way you worked around it. That was, 30 years ago, the way we did these things.

And of course over time it just became just sort of by agreement. Nothing ever really happened, but it just - the patent and trademark office just began becoming more lax in their interpretation. You know, under, I'm sure, plenty of political pressure for big U.S. companies to get patent protection for things that they want to cover with intellectual property law.

So continuing from this article in Ars Technica, it says: "It seems there will be some leeway for computer programs directly tied to improved hardware. The bill includes the example of a better washing machine. Even if the improvements are implemented with a computer program, 'the actual contribution is a new and improved way of operating a washing machine that gets clothes cleaner and uses less electricity,' so a patent could be awarded." So that seems sort of gray to me.

Leo: Yeah, one of the things they're talking about is you have to build a model of it, a physical model. That would be a good choice. Yeah?

Steve: Yeah, that - yeah. So anyway, this is - now, the question - oh, and they are saying they will continue to honor existing patents. But they simply will not issue new ones. So now the question is, here's New Zealand off by themselves with this. What happens with the rest of the world? And in fact they quote this Clare Curran: "One Member of Parliament who was deeply involved in the debate, Clare Curran, quoted several heads of software firms complaining about how the patenting process allowed 'obvious things' to get patented and that, 'in general, software patents are counter-productive.'"

And that's, I mean, that's always been my argument is that there's - in the patent language, the test is supposed to be whether it would be obvious to someone trained in the art. So, like, if you went through college, and you got your degree, and someone said "Solve this problem," that's called engineering, to use what you've learned to solve the problem. And that's different from an invention, which is like, oh, my god. Now, I don't know if what I will be revealing as the solution to web logon authentication should be an invention or not. I'm going to claim no ownership of it because it needs to be free. But maybe, I think it's obvious, but like there's a zipper, and there's Velcro. And so some things are obvious in retrospect, but weren't obvious prospectively.

So it's difficult. But it's certainly the case that what we see with software patents today is a company was just first. And so, because they were facing problems others hadn't yet faced, they're claiming that they invented all these things which anyone else with that problem would have also done. And I guess that's my argument. I think that isn't an invention, if anyone being asked to solve the problem who is a knowledgeable expert in that knowledge domain would have, like, just simply done the work, written the code. And that doesn't deserve protection.

So it is, unfortunately, our patent-granting system which is so broken. They say, well, that seems new. We'll let them battle it out in court. And that's the problem is this litigation is incredibly expensive. And in fact it's why I stopped volunteering to be an expert witness is that I was, for years. It's sort of fun to be involved. But I watched the court do the wrong thing so many times. It's like, oh. It just - it was more frustrating than it was gratifying to help people solve problems.

Leo: Yeah.

Steve: So I don't know what that means. But I think, if nothing else...

Leo: It's a step in the right direction.

Steve: Yes. It's progress.

Leo: It means people are paying attention to it, whatever.

Steve: Now, we have a lack of progress from Oracle.

Leo: Two steps forward, one step back. That's the way life is.

Steve: Ohhh.

Leo: I just hope it's not three steps back.

Steve: It turns out that Oracle is, like, they're reluctantly recognizing that they seem unable to control their own language, or at least the language that they inherited from Sun that is, of course, Java. So now there is a new security warning which Oracle has added to Java, to pop up and warn you...

Leo: Use of this software could be hazardous.

Steve: Yeah, it's like, no you know what. No kidding.

Leo: What does it say? Does it actually say that?

Steve: No, yeah, it actually says that this is potentially dangerous.

Leo: No. That's the solution?

Steve: Well, here's what they did, though. Two of the fields, the Name and the Location, where it came from and who created it, are not protected, and malware can change them.

Leo: [Laughing] Oh, I get it. So it warns you when you download a JAR file.

Steve: An applet cryptographically signed. Yet you can change those fields in the warning.

Leo: Well, they're just text strings [laughing].

Steve: Exactly.

Leo: Holy moly.

Steve: It's unbelievable. So Brian Krebs carried this. He said: "Faced with an onslaught of malware attacks that leverage vulnerabilities and design weaknesses in Java" - which puts it beautifully, Brian - "Oracle Corp. recently tweaked things so that Java now warns users about the security risks of running Java content. But new research suggests that the integrity and accuracy of these warning messages can be subverted easily in any number of ways, and that Oracle's new security scheme actually punishes Java application developers who adhere to it.

"Running a Java applet now pops up a security dialog box that presents users with information about the name, publisher, and source of the application. Oracle says this pop-up is designed to warn users of potential security risks, such as using old versions of Java or running applet code that is not signed from a trusted Certificate Authority. Security experts differ over whether regular users pay any mind whatsoever to these warnings. But to make matters worse, new research suggests most of the information contained in the pop-ups can be forged by malware writers.

"In a series of scathing blog posts," writes Brian, "longtime Java developer Jerry Jongerius details the various ways that attackers can subvert the usefulness of these dialog boxes. To illustrate his point, Jongerius uses an applet obtained from Oracle's own website, javadetection.jar, and shows that the information in two out of the three of its file descriptors, the Name and Location fields, can be changed, even if the applet is already cryptographically signed."

So, quoting from him, "'The bottom line in all of this is not the security risk of the errors but that Oracle made such incredibly basic errors in allowing "unsigned information" into their security dialogs,' Jongerius wrote in an email exchange. 'The magnitude of that fail is huge.' Jongerius presents the following scenario in which an attacker might use the dialog boxes to trick users into running unsafe applets: 'Imagine a hacker taking a real signed Java application for remote desktop control and assistance and placing it on a gaming site, renaming it "Chess." An unsuspecting end user would get a security pop-up from Java asking if they want to run "Chess" and, because they do, answers yes. But behind the scenes, the end user's computer is now under the remote control of a hacker [who], maybe to throw off suspicion, implemented a basic "Chess" in HTML5 so it looks like the applet worked...'"

Leo: Like this.

Steve: Exactly, "'all because Oracle allowed the Name in security dialogs to be forged to something innocent and incorrect.' Oracle has not responded to requests for comment," said Brian, "but Jongerius is hardly the only software expert crying foul for the company's security prompts. Will Dormann, writing for the Carnegie Mellon University's Software Engineering Institute, actually warns Java developers against adopting a key tenet of Oracle's new security guidelines. Oracle recommends that all Java applets be cryptographically signed, regardless of the privileges required by the program."

But get this: "Unsigned Java applets will run within a web page with a scary red warning that 'Running this application may be a security risk.'" Okay. So that's what the Java Runtime presents if you attempt to run an unsigned applet in a web browser. Which is good. Except, "One of Java's most-touted features is a 'sandbox' security mechanism that's supposed to prevent certain functions when the applet is sent as part of a web page. But according to both of these developers, Jongerius and Dormann, Oracle made the default behavior for signed code to be full access to the computer..."

Leo: Oh, come on.

Steve: "...essentially completely negating the usefulness of the sandbox." It's just crazy.

Leo: And it wasn't that way before Oracle? I mean, that wasn't - that's something Oracle did, they added as a feature?

Steve: Yes. These are things that have come along over time. These are their responses.

Leo: Thank you, Oracle.

Steve: In fact, they just - they cannot make it work right. Unbelievable.

Leo: We can't figure out how to do this, so we're just going to punt. Oh, lord.

Steve: A bunch of people noted to me, so I wanted to share the information, that BitTorrent Sync now has an iOS client. So it supports Android 2.2 and higher over on the Android side, and iOS 5.0 and higher over on the iOS platform. Still no information from them on the protocol. So we're still sort of in limbo. It looks good. They talk a lot about lots of bits of encryption and all that. But they won't tell anybody how it works.

So, and every time - I'm on their PR list. So I keep getting very nice updates from their PR guys saying, oh, Steve, it does this and it does that. And I write back. I say, that's nice. Please, the only thing I want to know is have the white paper on the crypto. We have to know how it works. And he says, oh, okay. But also it's pretty, and it has ribbons. And I say, I know, I'm sure it is, but all I care about is the technical details. When you have those, I will happily study it and then tell everyone that you guys did it right, assuming that they did. But until then, we don't know.

Google Authenticator, I know you know this, Leo, made a huge mistake.

Leo: Yeah.

Steve: Ouch. And it's now pulled from iTunes.

Leo: This was on iOS only, by the way.

Steve: Yes. An update for iOS, when updated, wiped out the secure store of all of your Authenticator account.

Leo: It just - it started over.

Steve: Yeah, yeah.

Leo: But, you know, I have to say people should be saving their secret keys. I save it in LastPass. So it's very easy. And the reason I do it, not because I was worried about that, although this is a benefit there, but just because I want to set up Authenticator on other platforms.

Steve: Yes.

Leo: So I just - I save actually the image of the QR code into my LastPass, into my secure...

Steve: Yeah, it requires more steps. And so not everyone was doing. But, yes, you're right. And it would be nice if there were some, like, good backup facility.

Leo: Some automatic one, yeah.

Steve: So many people were getting hurt by this that Google yanked it immediately from iTunes.

Leo: Well, and I think what happens is people - so let's say you're using Google's second factor authentication. Google gives you a QR code. You snap the picture in the - the QR code just has a long number, which is your ID, and so you don't have to enter it in by hand, just snap a picture of it, and Google Authenticator now has it. Then people just delete it. They go on - so there's no other place that it's stored. That's just dumb, frankly.

Steve: Yeah.

Leo: But they should tell you that. They should say, "Store this QR code somewhere securely."

Steve: Yes, because you're going to need it. Print it out on paper and stick it in a drawer.

Leo: Right, right. Now, you can always ask for another one. So I don't know how big a deal it is.

Steve: Although I guess, obviously, you have to authenticate somehow through a different means.

Leo: Right, right. But everyone who uses this has some other means. I just - Evernote uses this now, offers Google Authenticator, which is great. More and more people are doing this. I mean, I feel bad for anybody who's on iOS and lost them all. I use it like crazy with LastPass, with Google, with Outlook. Microsoft uses it with Evernote.

Steve: Yeah. And just wait till I'm able to tell you how it should be done.

Leo: Oh, I can't wait.

Steve: I know. Okay. So there was a hoax that upset a lot of people. I wanted to let everyone know it is not the case that TrueCrypt has a backdoor.

Leo: Oh.

Steve: This weird document suddenly was floating around the Internet that looked really authentic at first blush. It appeared to be - and if you click that link, Leo, bring up the PDF in my notes there, you can put it on the screen - from the National District Attorneys Association. Subtitle was National Center for Prosecution of Child Abuse. And this was a presentation, a slide presentation titled "Computer Forensics for Prosecutors," dated February 18th and 19th of this year, 2013, Portland, Oregon. And it's a series of absolutely legitimate slides that would be part of a presentation being made to - sort of for law enforcement about computer forensics technology. It talks about hard drives and encryption and hashing and is sort of a good grounder.

Toward the end of this really authentic-looking presentation there's a slide that is just labeled, "What's a Backdoor?" And then underneath it says, answering the question, "A method to bypass data encryption or security." And then as bullet points it says, "Does not require the password or passphrase to be known." Next bullet point, "Saves time, cost, and effort to access encrypted or secured data." Third bullet point, "Allows data to be accessed, copied, and even modified without tipping off the owner." And then coming out a level of the outline, "Currently available for major encryption software: Microsoft BitLocker, FileVault, BestCrypt, and TrueCrypt," and then it says, "et cetera. Currently implemented by major cloud storage provider to comply with NCMEC requirements." And

it just sort of goes on.

So people see this and freak out, thinking, I mean, looking, thinking this is like an absolute legitimate presentation. I mean, everything about it up to now, I mean, even through this, looks legitimate. Except the last slide. Down, if you look carefully at the last slide - this is the end of the first part of the presentation, so says "Part 2: Detective Stu Pitt..."

Leo: [Laughing]

Steve: "...will take over for Part 2. And tomorrow Detective Laughlin Foo will conduct Part 3." And the legitimate original document has also been found from which this spoofed one borrows heavily. But the other one is much longer, and clearly many of the same slides were taken and so forth [cryptome.org/2013/09/computer-forensics-2012.pdf]. So this was just a hoax. And Detective Stu Pitt and Detective Laughlin Foo will probably not be delivering Part 2 and Part 3 of the presentation.

Leo: That is so funny. There is one more little revelation somewhere, when I click this. Oh, no, okay. I guess the name of the PDF is "Hoax," but that was probably added after the fact [cryptome.org/2013/09/computer-forensics-2013-hoax.pdf].

Steve: That was added afterwards, yes.

Leo: Oh, okay. That would also give it away.

Steve: I got a bunch of tweets from people, oh, my god, there's a backdoor. It's like...

Leo: No.

Steve: I don't think so. Okay. So I had intended to cover this topic, Tor traffic analysis, so I printed out the 12-page, detailed, small print, two columns PDF and took it with me to a meal. And only when I was sitting, I had sat down and was getting ready, I saw that my toner had just about run out on the printer.

Leo: Oh, so you had a bunch of blank pages.

Steve: I got big huge empty stripes down the - so anyway, new toner is on order. I will just say that this looks really interesting. It was this research on Tor traffic analysis - and remember, we were talking about traffic analysis just recently because it is the Achilles heel, and it also feeds nicely into our discussion of Bitmessage. This paper was put together by researchers at the U.S. Naval Research Lab, which of course was the original sponsor of Tor. They did the original work on onion routing under the auspices of DARPA, and also some people at Georgetown University. So this is - it's a beautiful piece of research.

And just reading from their abstract, they said in the abstract of this: "We present the first analysis of the popular Tor anonymity network that indicates the security of typical users against reasonably realistic adversaries in the Tor network or in the underlying Internet. Our results show that Tor users are far more susceptible to compromise than indicated by prior work. Specific contributions..."

Leo: No.

Steve: Yes. "Specific contributions of the paper include a model of various typical kinds of users; an adversary model that includes Tor network relays, autonomous systems, Internet exchange points, and groups of Internet exchange points drawn from empirical study; metrics that indicate how secure users are over a period of time; the most accurate topological model to date of the anonymous systems and Internet exchange points as they relate to Tor usage and network configuration; a novel realistic Tor path simulator; and analyses of security making use of all the above. To show that our approach is useful to explore alternatives and not just Tor as currently deployed, we also analyze a published alternative path selection algorithm, Congestion-Aware Tor. We create an empirical model of Tor congestion, identify novel attack vectors, and show that it, too, is more vulnerable than previously indicated."

So I will digest this study, figure out what it means, and we may just do a podcast on it because that's significant. But basically it sounds like it's not good news. I don't yet know how bad the news is.

Leo: Yeah.

Steve: Okay. So before we began recording, Leo, you and I had a lot of fun talking about one of our passions, which are TV shows.

Leo: TV, yeah.

Steve: So in a weird - I haven't seen this from him before - blog, Paul Krugman, who is with the...

Leo: Nobel Prize-winning economist.

Steve: Yeah, exactly, who blogs for The New York Times and is a regular columnist for The New York Times. The title of his - and this is a good friend of mine sent this to me, otherwise - and I'm really glad from the description. The title of his blog post was "Send in the Clones." And then he said in parenthesis, "Unserious Entertainment Advice," except he's serious about it. So I want to share this with - this is under - we're in Miscellany, obviously, now.

He wrote: "Hey, if I can post music videos once a week, I guess I can recommend a TV show now and then. Just finished watching our DVRed Season 1 of "Orphan Black," and wow. If you haven't heard about it, it involves a number of women who discover that they are clones, products of an illegal experiment. All of the clones are, of course, played

by one amazing actress, Tatiana Maslany, who not only changes accents, but changes her whole body language when she shifts from London grifter to soccer mom to science geek to murderous religious fanatic."

Leo: Wow.

Steve: "She even does the soccer mom impersonating the grifter and vice versa..."

Leo: Wow.

Steve: "...and somehow makes it clear that that's what's going on. Eat your heart out, Alec Guinness," writes Paul. "And with the magic of modern technology, there are multiple scenes in which, say, three of the clones are talking to each other, and you really do forget that we're watching repeated takes of the same actress. Oh, and Max Headroom appears to be the big villain, although in this show nothing is what it seems." Then he concludes: "I think they're rerunning Season 1 this fall, and there will be a Season 2 next year. Highly recommended." And that's a BBC production. [Begins September 14, 2013 on BBC America.]

Leo: And you can buy it on Google Play, which I'm about to do right now.

Steve: Yes. It really sounds interesting. I grabbed the Blu-ray, the first season on Blu-ray disk, and Amazon said they've sent it to me, so it's on the way. And it is available in your other shadowy sources, as one would expect.

Leo: Also Play Store, so you can get it for download. I'm getting it right now from there.

Steve: I think it's 10 episodes of the first season, and really sounds intriguing. So I'm not representing it one way or the other. Haven't seen it. But I...

Leo: Chatroom agrees. Some have seen it in the chatroom. They say it's incredible.

Steve: Oh, fantastic.

Leo: Yeah, can't wait to see it, yeah. Buying it right now.

Steve: Okay. So a couple days ago I posted something in the grc.spinrite.dev newsgroup, where I am hanging out full time, working on SpinRite. And in a minute I will update everyone on that because I hit a major milestone yesterday. What I posted was this. Subject was "Something I cannot ignore any longer."

"Gang: Three days ago" - so I guess this must have been Monday I posted this - "during

breakfast last Thursday morning, I came up with what may well turn out to be the solution to the whole website authentication problem. It requires no username or password. It's 100% anonymous. It gracefully supports multiple unlinked personas. It's FAR" - in caps - "more secure, quick and easy to use than time- or sequence-based one-time passwords. It inherently thwarts man-in-the-middle-style attacks, and it's comparatively safe to use in public settings where keystroke or other logging might be present. The whole thing is so simple, and almost obvious in retrospect, that I can't believe that no one else has hit upon it before. But I've searched, and apparently it's nowhere but in my head.

"It's inherently open, free, and TNO. It can easily coexist with any other existing traditional authentication system, gradually taking over as it becomes more popular. And it doesn't require any sort of third-party. The interaction is just between the user and any supporting website that wants to offer this authentication alternative, and any website that wished to simply could. There's no large startup cost, no critical mass needed. There's really no way for anyone to make any money with it. It needs to be free.

"I wasn't trying to come up with anything like this. I wasn't even thinking about the topic. I've been 100% saturated by this present work on SpinRite. But, as you all know, I've also been living in that realm, thanks to the podcast, which keeps my attention focused weekly. And you all know also what a passion I have for the problem with Perfect Passwords, Perfect Paper Passwords, Off The Grid, and Password Haystacks. It's THE problem.

"So we'll get this present work on the ATA bus mastering DMA working, solidified, and finished. Then I need to take a BRIEF" - and I put that in caps - "hiatus from the v6.1 work to create a web page describing and explaining my proposed solution. Implementation is not something that I need or can do. It's way bigger than me and would need an RFC-style standards body to ratify a single standard. So I won't be away from here for long, only long enough to create a page carefully describing the idea. Then I'll give it a podcast to launch it into the world and send it on its way. Then we'll immediately plow into adding AHCI controller technology as a next step in this work."

So there were a couple responses from people. One from a very security- and crypto-savvy guy who's also a very great contributor to the newsgroup said: "Interesting. Does it solve these problems, as well?" And he said: "One, can be backed up and restored easily." Yes. "Two, provides a key to the authenticator that can be used for encryption on the user's behalf." And actually, as a matter of fact, it does that. He said: "Three, can be completely protected by something you know, i.e., a master password." And the answer is yes, trivially. And he said: "Those are my three must-haves for any authentication system." And I said, yup, it's got them all. And then somebody else posted: "Okay, I can't stand it any longer." This was, like, two days later. He said...

Leo: You're such a tease.

Steve: "Have you given it a name?" and I said: "It has a name, a pretty good one. But, if I share it, it's likely to set off a firestorm of speculation, which I would prefer to avoid for the time being. I shared what happened last Thursday and what has been on my mind because it was the right thing to do when I found my own focus and concentration, though not my time, increasingly distracted by the idea because ultimately it will have some - likely modest - impact on this SpinRite work.

"Since then, pieces have been coming together, and the range of applications is

expanding. The patent landscape appears to be completely clear, with everything required either in the public domain or explicitly released from any usage encumbrance. The world should look at and consider it sooner rather than later, so I don't want to wait long. And, since there are interfaces among the pieces that need to be standardized to create a single universal interoperable solution, it would be better if my initial proposal was well thought out and fully specified so that interoperable endpoints could be immediately created from the initial disclosure.

"Because it's always possible to miss something, and because it's inherently impossible for me to adequately attack anything I have created, it needs to be reviewed by crypto-savvy third parties who can approach it from an adversarial perspective." And then I said: "If you think about it, that's exactly what this SpinTesting work really is that we do here, though it's other people's hardware that's taking an adversarial role."

So I will - SpinRite work is really going well, and I'm days away from - I'm probably later today releasing the next iteration of testing. We are now able to transfer, all in assembler, at the hardware level, 32MB contiguous blocks up into extended memory. So all of that - remember I talked about the real protected mode, of how with 16-bit code you can change, as a consequence of that weird fluke in the original implementation of the Intel system, you can actually get 32-bit addressing by sort of breaking the way segmentation is handled. And that's all working.

So we have a 32MB transfer buffer, and we are now using Ultra DMA at the highest speed the drive can go to transfer 32MB at a time, so SpinRite will be screaming along. We're doing that right now on the older style ATA specification. And I want to - originally I was thinking I would do 6.1, and then I would hold off to do the AHCI controller. But so many people have that now, and I'm right in the middle of all this. So it's like, oh, let's just get it done while I'm in the middle of it. So I do want to take a break and create a web page to explain this Eureka! Aha! event. And when I tell everyone, they're just going to go, well, that's obvious. I mean, it's so simple. And it's just like, yeah, it is, like a zipper, or like Velcro. But somehow no one has done it. And it just - it solves every problem. It's really cool. So...

Leo: Good, I can't wait.

Steve: The problem is I think you're going to be gone.

Leo: Yeah, I am. I'll miss it. So I'll have to hear about it from a distance.

Steve: Oh, well, we'll see how the timing goes. But, yeah.

Leo: Yeah, I can't wait.

Steve: Yeah. So that's where SpinRite is. And we're coming along really well. And it's looking like it's going to scream.

Leo: Okay. Okay. And you don't want to say the name out loud of this thing.

Steve: Nope.

Leo: Because people will do what they did, in fact, in the chatroom, when you did mention it before the show, speculate as to what it is and how it might work.

Steve: One other person in the world knows about it, because we had a three-hour conversation by phone yesterday, Mark Thompson.

Leo: I figured you'd tell Mark.

Steve: I needed someone, I needed to bounce it off of somebody really smart. And Mark is, similarly, he's, like, holy crap. And I said, I know. And he loves it. I mean, it's like, so correct.

Leo: Good.

Steve: And but there was a point I was going to make. That's the reason I mentioned Mark.

Leo: Well, that's the only person who knows what...

Steve: You were asking, you were...

Leo: Well, just why you didn't want to say the name, because you didn't want to stimulate people to say what it might be.

Steve: I don't remember where I was going with that thought, unfortunately. But anyway, so one person knows. And, oh, I know what it was. Mark felt that in order for this to succeed, I had to do everything. I had to have the mobile app...

Leo: Before you announce it, yeah.

Steve: And I had to have web-side stuff and everything. I don't think so. I mean, I understand Mark's position. But this is so compelling, it will just - it will immediately kill one-time passwords. They're dead. They are, I mean, the reason - it's just, it's so much better than that. And so I want to lay out the concept. But it's crazy for me to - I just don't have the time. And it doesn't make sense for me to, like, try to do the whole thing. It's not necessary. The concept will - it's so much better than anything we have ever seen that it will just happen. It will acquire its own traction.

Leo: Good. I look forward to finding out more. We don't have an ad. You can go

right into Bitmessage, our topic of the day.

Steve: Okay. So, Bitmessage. As I said at the top of the show, I'm not bothering to dissect the protocol because there are too many problems with it. In time, when it gets to 1.0 - it's currently at 0.3.5 - then maybe these things will be figured out. Or maybe it will have sort of been the first shot. I think I was reading that Bitcoin - no, no, it wasn't. It was the guy who did Litecoin. This was not his first alternative to Bitcoin. He did a first one, and there were a lot of problems with it. And then he figured out how to do it right, and he learned from that and did Litecoin. So similarly, this is sort of - this was just proposed in the wake of the surveillance that was assumed to be going on. And this was even - this was back in 2012, late. I think it was, like, maybe November of 2012 that the whitepaper was produced proposing it. It's a little short six-page document just sort of laying out the concept. And there's now code.

So one of the things it's getting is a little bit of credibility that it really doesn't deserve from Bitcoin because the reason I got so excited about Bitcoin when we did the podcast was that it was done so right. So it's called "Bitmessage" sort of unfairly because the only thing it really has in common with Bitcoin is, first of all, the "Bit" prefix. The fact that there is the concept of a partial hash collision, which was one of the things that was so cool about Bitcoin, the notion of a proof of work, the way the Bitcoin system slows down the generation of coins to keep them at a constant rate in the face of increasing amount of total work being done to create coinage by increasing proof of work. Bitmessage uses that to thwart spamming because flooding of the network is an inherent problem.

And so the guy recognized that he needed to limit the way, to limit the ability to inject messages into the network. And so there's a proof of work as part of this. So that's something it borrowed from the Bitcoin concept. And then the other thing is that the way Bitcoin works, as everyone knows, is that essentially everybody gets the block chain. And so that is to say that you are - it's a peer-to-peer network where everything receives the current state of Bitcoin. And Bitmessage has that same "everyone receives everything" model, although it isn't a chain, and the information is not kept forever.

So those are the only - that's all there is, really, to tie it to bitness relative to Bitcoin. So I wanted to differentiate it from Bitcoin. It's not like this is some messaging system written on top of Bitcoin or in any way related to it. And so we shouldn't give it any props for being a relative of Bitcoin. It isn't. It just has that name.

Leo: Can you just tell me what it does?

Steve: Yes.

Leo: [Laughing] I mean, I can kind of infer from the name, but I'm curious.

Steve: Yes, yes. So it is a peer-to-peer messaging system where everybody who wants to use it runs the Bitmessage client. Which I think is a Python app. I think it's written in Python. And so it's multiplatform: Windows, Linux, and Mac. So they run this client, which connects into this peer-to-peer network. And all users retain the most recent two days' worth of messages. So when somebody new connects up, they connect into the peer-to-peer network, so they're adding their node to this, and they receive from the

peers the most recent two days' worth of Bitmessages, which the entire network maintains.

So basically this is a bunch of peers that are passing messages around among each other. Anyone who wants to send somebody else in the network a message can. So everybody gets an asymmetric key pair, a public key and a private key. The hash of your public key is obviously a much smaller token. And the Bitmessage starts with a BM for Bitmessage, hyphen, and then it's a 36-character, we're used to seeing these pseudorandom strings, just looks like gobbledy-gook. It's weird. And I think it's like base58, which I've seen some people say, huh? Why 58 and not 64? It's like, okay, 58 for some reason. Maybe it's to make - maybe they eliminated some of the visually confusing characters in case someone was going to type it in. You could type in an address. Hopefully you don't have to very often.

So everyone is known in the network only anonymously. So there's no username. There's no password. In that way it's sort of also like Bitcoin. You are just this string, this token, which is a hash of your public key. So you could put that on your website. You could email it to a friend. You could do whatever you wanted to with it. And there are four types of objects in the system. There's a request for the public key. So given the hash, you could ask the network for this user's full public key, which you would use to encrypt a message that only they can read. So that's the way a person-to-person message is sent. Then another object is the public key. There's also a person-to-person message or a broadcast because what you could also do is a person could put onto the network something that they want multiple people to read. And so rather than encrypting with a recipient's single public key, so that only they're able to receive it, you would instead encrypt with your private key so that anybody who wants to read what you have broadcast is able to do so.

So anyway, so basically it's a weird concept. All these people are in this completely, this densely interconnected peer-to-peer network. And anyone who sends a message, that message is received by everyone. And the only way you know if a message has been put in, sort of like dropped into the network, where it propagates across the entire network, is if your private key can decrypt it. So messages are coming in, and your client checks each one that comes in to see if it's able to decrypt it. And if it is, then it must have been encrypted with your public key and thus bound for you. And so, when you think about it, this defeats to some degree, but there are many - this has been around long enough, and enough people have looked at it, that there are people beginning to find little chinks in the armor here.

The concept was that, since everybody received all messages, there was no way to tell when a message was meant for someone. Which is clever. I mean, it defeats traffic analysis just by virtue of sending everybody everything. It's like, uh, okay. So that's essentially the concept. There are - normally, messages are acknowledged. And so some critics of this have said, wait a minute. If messages are coming in, and they're being selectively decoded, and they're acknowledged when decoded, then a node, a user, could be seen sending an acknowledgment out, which would mean that that's acknowledging something they received. So you can see where you could begin to chip away at this, at the opacity that the entire system was designed to have.

Also it feels very immature from a cryptographic standpoint. Normally what you do, and we've talked about this often, is if you wanted to encrypt a message, you would use a pseudorandom number generator to generate a nonce, a one-time symmetric key. You would use any of the well-established, proven, cipher-block chaining approaches to encrypt your message under that symmetric key. Then you would only use your - either the private or the public key, the asymmetric key, to encrypt that symmetric key. And

you attach it, and off it goes.

As I understand it - and again, I didn't dig really deep, I'm just looking at comments from people who have been critical of it - the system uses the asymmetric key for the bulk encryption. Now, part of their motivation was probably to slow down the creation of longer messages because there is this intention in the system for the bigger the message is, the harder it is to create it. So they may have deliberate- I mean, it's hard to imagine that somebody who created this wouldn't have understood how basic PGP or SSL, I mean, all encryption uses the concept of a pseudorandom key that you use for symmetric encryption, and you use the much more slow asymmetric encryption only for the symmetric key.

Anyway, as I understand it, this system doesn't do this. Also, there is no interblock connection. And I've seen some people talking about that you could reuse already encrypted blocks and reorder them. And allowing blocks to be reordered is never a good idea. And you can also append additional information to the end of blocks. Anyway, it's like the implementation really feels weak. So again, it's receiving well-deserved criticism. And I think maybe at some point it'll get off the ground. It's not clear. It is possible to tell people you are very concerned about privacy, so you will not respond with an acknowledgment. Otherwise, the problem is, if somebody were sending a message to you, remember that messages only stay, in all, sort of are hosted by this network for two days. So what if the recipient was offline for two days, and then the message that was bound for them they would never see?

So the idea is that you would need to check in every two days in order to get an update of all the messages that you're behind on so you can see if any of them are for you. Oh, and when you receive it, you're supposed to acknowledge it to the sender so the sender can remove it from their sender's queue and essentially know that the recipient got it. Otherwise the proper behavior, if the sender really wants to verify receipt, is an exponential back-off, where they'll wait two days, then they'll wait four days, then they'll wait eight days, then they'll wait 16 days, resending, each time doubling the length of time. Then the logic on the receiver's side is you need to listen, if you've been away for some number of days, like 16 days, you need to listen for at least that length of time in order to create a window during which the sender will resend during the period of time you're looking.

So anyway, it's, as you can kind of hear, it's like, uh, okay. It's kind of clunky. People are really concerned about scaling. The original author of the document addresses this. And he comes up with a way of, like, forking the network through a series of binary decisions to create substreams, and then a sort of a complex way of managing parents and siblings of streams. And anyway, it's just - it's interesting, but it just doesn't feel like it's there yet. And I actually found myself being encouraged by looking at some of the very good criticisms of it. And as I mentioned at the top of the show, people who are saying, eh, well, I'm working on something that I think is going to solve these problems, so I want to let everyone know that. And but in the meantime here's what I think about Bitmessage.

So it's like, yeah, if you need - I think if you're curious about things, grab it, load it up, play with it. People who have played with it say it's very cool. They get this token which is their identity. They send it to a friend. And then they get their friend's token. And then in the client you're able to, say, generate a person-to-person message or a broadcast. So you choose which type of message you want to put out onto the network, and then you just type a message and send it. So there's no support currently for files. There's no support for formatted text and all those things. Those things could come later, in this or in some other form. So it's interesting. But I don't think it's really there to be taken seriously.

Leo: Yeah, I guess somebody said in the chatroom that Dvorak's using this on No Agenda. So there. If you're a fan, you might want to mention this.

Steve: Yeah, there was also, somebody did an experiment to de-anonymize Bitmessage users. Because you can see the tokens of people on the network, this person collected thousands of current Bitmessage tokens, then requested their public keys so he could send them something, and then sent each person a customized message with a URL to a server. And it made it look like it was an official Bitmessage announcement addressed to them. And so it's like 15,000 people clicked the link and immediately lost their anonymity because, of course, not only did he have their IP, but the URL was customized so he knew which Bitmessage token was associated with which user coming in, and therefore what their browser query headers were and the IP where they were.

Leo: Wow.

Steve: So that's, now, arguably that's an out-of-band attack. It's clever. And so it's the standard, well, don't click on links, please. But it's an example of how the system could be abused. So anyway, it's interesting. It's like, eh, okay, we need something better.

Leo: And Cryptocat still a good choice, do you think?

Steve: For, yes, I would say from point to point. See, Cryptocat is real-time. It requires you both be online in order to...

Leo: A traditional IM system, yeah.

Steve: Yes, like a traditional IM system. Whereas this is more of an asynchronous...

Leo: Store it forward, yeah, yeah.

Steve: Yes, store-it-forward asynchronous messaging.

Leo: Google Hangout says that, too.

Steve: Yeah. So for real-time interaction, the Cryptocat implementation of OTP is great.

Leo: And you can still use PGP and encrypt any arbitrary bit of text.

Steve: Well, and remember that, when we talked about OTP, the protocol, of which Cryptocat is just one of many implementations, many existing, like Trillian, many existing IM clients already have OTP support. And it's bulletproof technology, though it is real-

time because you need to negotiate on the fly. And so, I mean, there really are existing good solutions for this. And this is just interesting. And I liked - I think the real attraction was this concept of no one can tell when you're the recipient because everybody gets everything. And it's like, okay, well, there's good and there's bad to that. And scaling is one of the problems because you can see the problem with spamming. Imagine if something injects spam into this, and now everybody, every single node has to store every single message for two days. As this system gets bigger, it just - scalability is a real problem.

Leo: Steve Gibson scales amazingly. Yet still we must wait for his amazing revelation.

Steve: Yeah, I'm sorry, I don't mean to tease, but I just - it hit me, and I just - I want to get it down, put it on...

Leo: You don't scale. You don't scale either, come to think of it.

Steve: No, I don't.

Leo: One thing at a time.

Steve: That's the problem. Several people have said we wish we had two Steves.

Leo: He's single-threaded. Even if his brain is multitasking, his abilities are single-threaded. And that's as it should be. You can go to GRC.com and see all the things he's cranked out one at a time.

Steve: One more, one more biggie very soon.

Leo: Good. I'm excited.

Steve: Yeah.

Leo: That's where you get SpinRite, the world's finest hard drive maintenance and recovery utility. You must have it, if you have a hard drive. That's at GRC.com. Steve's also got lots of other stuff, including his feedback form. This would be an opportunity for you to ask questions of Steve, GRC.com/feedback. We'll be answering them next week, god willing. God and the security environment willing.

Steve: If nothing horrible happens.

Leo: If somebody doesn't leave his TrueCrypt password unencrypted. What else? Oh, 16Kb audio is there. Transcriptions by Elaine Farris. It's a great resource. You can also follow Steve on the Twitter: @SGgrc. And if you want full audio or video versions of this show, we have those, too, at our website, TWiT.tv/sn. And of course you can always subscribe to any version just by going to iTunes or wherever you subscribe to podcasts.

Steve: And Leo, I have agreed to come up and be with you for New Year's.

Leo: Yay. So that's something going to be exciting. We're going to look for people who are in every time zone because we're going to do the 24 hours of New Year's. I'm the only one who's going to stay up 24 hours. You don't have to.

Steve: I don't plan to.

Leo: But we're going to get a lot of people up here. And I'm really thrilled you're going to do that. And we're just going to have a party. And every hour, and actually in some cases less than an hour because there are 26 time zones, we will do a countdown to New Year, starting at 4:00 a.m. New Year's Eve. That's what I calculated to be the beginning of this program. Four in the morning, New Year's Eve.

Steve: Okay, now, and so the podcast is normally, let's see, where are we on the 2014...

Leo: We'll do you at the normal time.

Steve: New Year's Day...

Leo: Oh, yeah.

Steve: New Year's Day is Wednesday, January 1st.

Leo: Right. I'll be done by 4:00 a.m. New Year's Day. But I'll stick around if you want to - you know, we should just do the show the day before.

Steve: Yeah, let's do it the day before.

Leo: We'll do it New Year's Eve.

Steve: When you're still, like, able to put sentences together.

Leo: Early. In the first 12 hours of the show. If we do it at the normal time a day early, that'll be fine.

Steve: Perfect, perfect.

Leo: But we'll have to do it after the 11:00 a.m. countdown because it's New Year's, that's the beauty of this, it's New Year's Eve somewhere for 24 hours.

Steve: Leo, we can do it in four 15-minute segments.

Leo: That's what we'll have to do.

Steve: I'm happy - I'll work with you.

Leo: Oh, I'm so glad you're coming. We're starting to line up people. And it looks like it's going to be a great party. I'm hoping we get some music and stuff. Anybody who's listening who wants to be up here at the Brick House for that, please do. New Year's Eve, we start at 4:00 a.m. New Year's Eve morning. And then we go through 4:00 a.m. New Year's Day because we have to get Hawaii. And then we're done. Should be fun. I haven't done 24 hours in a long time.

Steve: Yeah, not since you were...

Leo: Not since the iPhone.

Steve: ...young, Leo.

Leo: Well, no, no, not that long ago.

Steve: Oh!

Leo: 2008. When the iPhone came out in 2008 we did a 24-hours of the iPhone.

Steve: Wow.

Leo: I'm hoping I'm going to get a nurse to check my blood pressure. I want to get a masseuse to give me backrubs. I want to get a barber to come and give me a haircut and a shave. We're going to - it's going to be an endurance.

Steve: Oh, my god. You know what I'm going to do?

Leo: What?

Steve: I'm going to bring my coffee for absolutely...

Leo: I will need you to do that for me, yes.

Steve: Yes. You have got to taste this coffee which everyone I expose it to says, "This is coffee?"

Leo: You said you were going to send me a kit. But we can hold off till New Year's Day. That's fine.

Steve: We're holding off because I don't trust you to, like...

Leo: No, no, no. And we have to titrate - we have to titrate this every hour.

Steve: I have to oversee the production, the grinding and the production. You have all the equipment. I'm going to bring the raw materials. And we'll hand you the cup of coffee and see what you think.

Leo: Yes. And I want 50 ml, 50 cc every hour. I will slowly titrate this.

Steve: It's just fabulous coffee, Leo.

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

Steve: Bye.

Copyright (c) 2012 by Steve Gibson and Leo Laporte. SOME RIGHTS RESERVED

This work is licensed for the good of the Internet Community under the Creative Commons License v2.5. See the following Web page for details: http://creativecommons.org/licenses/by-nc-sa/2.5/