BitTorrent Sync

Description: After catching up with the week's security news, Steve and Leo examine everything that's currently known about the recently released "BitTorrent Sync" peer-to-peer file sharing and folder synchronizing application. Everything seen so far looks 100% correct and VERY useful.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-402.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-402-lq.mp3

SHOW TEASE: It's time for Security Now!, ladies and gentlemen, our May Day episode, with Steve Gibson. We're going to talk about something new from BitTorrent that actually sounds amazingly useful: BitTorrent Sync. Some security updates, too. It's all next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 402, recorded May 1, 2013: BitTorrent Sync.

It's time for Security Now!, the show that covers your security and privacy online with this guy right here, the Explainer in Chief, Mr. Steven "Tiberius" Gibson.

Steve Gibson: Oh, tease me, Leo. We have another Star Trek in about two weeks.

Leo: I know. I have to throw that at you. Oh, that's right, this is going to be the...

Steve: I think it's May 17th is the date.

Leo: The solar flare...

Steve: Not that I'm counting or anything.

Leo: The sun flare Star Trek. Reboot. I'm excited. Looks good. Everybody's loving the trailers, anyway.

Steve: Well, and J.J. does such a good job with these. I mean, he's turned them from sort of slow-moving, intellectual, thoughtful sci-fi into real action movies and has consequently really broadened the audience.

Leo: Right, I agree. I can't wait to see what he does with Star Wars. That's going to be amazing.

Steve: What?

Leo: Isn't it J.J.? Yeah, J.J. Abrams is directing the next Star Wars.

Steve: Well, how - what next one?

Leo: What next Star Wars? Oh, you're not up on this, huh? So Lucas...

Steve: Luke is in his retirement? He's a retired Jedi?

Leo: Well, Lucas sold the franchise to Disney. Remember that?

Steve: I remember.

Leo: Okay. So Disney's already signed J.J. Abrams to do - I'm not sure which episode it'll be.

Steve: I do remember that, now, too, yeah.

Leo: Yeah. And all of the originals will make appearances, obviously as old men and women.

Steve: Maybe Jar-Jar could be in a casket. That'd be...

Leo: Princess Leia: "I did a lot of coke in my middle years, but I'm back. Help me, Obi Wan Kenobi. You're my only hope." And it'll be fun. It'll be fun. There's something about Harrison Ford, he just - he has a crazy look in his eyes all the time. Am I not right? I'm not wrong?

Steve: I watched him on "42," which was a great movie, by the way. I did enjoy that. That was last Friday. This Friday, of course, we have "Iron Man 3" coming out, so...

Leo: Yeah, I'm looking forward to that. Looking forward to that.

Steve: Did really well in its European release, just made huge money.

Leo: Yeah, people are raving about it. There's something about Harrison Ford now. I don't know, just looks like he's got crazy eyes. There's just something about him. He's scary-looking. Maybe it's that smile. Anyway, let's get to the - you know, we're not here to discuss Star Wars.

Steve: They're not going to get Harrison Ford in this Star Wars.

Leo: Apparently they are.

Steve: No.

Leo: Everybody. Luke, everybody. They're all signed. That's the scuttlebutt.

Steve: Okay, that would be fun.

Leo: But they won't be the stars. It'll be just like Leonard Nimoy was in the last Star Trek.

Steve: They'll all be in wheelchairs.

Leo: So...

Steve: That, yeah, anyway, never mind.

Leo: Anyway, never mind, no.

Steve: Let's do a podcast.

Leo: Let's do a show. And this is a show about security. What are you going to do today? I think you said - did you say BitTorrent?

Steve: I did. We've never really talked about BitTorrent except as it relates to peer-to-peer filesharing technology through the last, what, seven or eight years of the podcast. But they have released something last week which was - which they began - we began to hear rumors about it a couple months ago, in January. And it generated so much buzz

among our listeners as relayed to me through Twitter that I decided I had to go take a look.

And from everything I've seen, they've done everything right. This is BitTorrent Sync. And the compelling aspect of it is that it uses no third party. It is not a cloud storage solution. It allows individuals or ad hoc groups to create their own drive synchronization networks using the well-proven BitTorrent peer-to-peer technology, the NAT traversal, I mean, all the things that make BitTorrent work, they've moved over into sort of a standalone sync client that is cross-platform, runs on everything. The security looks very good. We'll get into it in detail. But I think this is a strong contender for a great solution for people whose need fits what this does. So there was so much interest, I thought, well, let's talk about it from a security standpoint.

Leo: Yeah, because it could be a free replacement for Dropbox, for one thing; right?

Steve: Yes. Yes.

Leo: But we've got to think about what we're doing here.

Steve: Yup.

Leo: This is certainly a crowded category. Good. And of course we've got the security updates. Boy, I'm just looking. And there's one big one that we really want to talk about. So, Steve, let's get into the tech news, or the security news.

Steve: We have some hacks of the week, as we almost always do. Probably people know, although I didn't mention it last week - I'm not sure if it happened after last week's podcast. But there was a hack of the Associated Press Twitter account where someone tweeted, so essentially on behalf of the Associated Press, that there had been a bomb at 1600 Pennsylvania Avenue. And the U.S. stock market...

Leo: Crashed.

Steve: ...immediately crashed.

Leo: Yeah, yeah.

Steve: It was a spike. It recovered a few minutes later, after it became clear that this was a false report.

Leo: It's programmed trading. I mean, it's so fast now, it's not a human doing it. It just happens, you know.

Steve: Right. So there were bots that were monitoring the Associated Press feed, and they triggered on keywords, "White House" and "bomb," and...

Leo: Sell! Sell!

Steve: ...immediately sold off a lot of stuff.

Leo: Yeah, yeah. Understandably, frankly.

Steve: So the lesson here, though, which is really important, obviously, is that something that began as just sort of a, like, oh, you know, I just had a great hot fudge sundae, has become a major artery for mission-critical communications. I mean, Associated Press is sending out news on their accounts. So the security of those becomes important. And this was a group called the Syrian Electronic Army that took credit for last week's hack.

Well, they're back, and this time they claim to have hacked 11 of the Guardian's accounts. The Guardian is, of course, a very large U.K. newspaper and media outlet. Twitter immediately suspended most of the Guardian's accounts. And then they have been proactive, really for the first time, where they have gone out and contacted major news organizations around the world who have Twitter accounts and use them for these sorts of purposes and to, like, work with them to verify their security and explain to them that this has really become important. And of course we do know that Twitter is adding - is now - there was a job offer out, I think it was in February, looking for a programmer to implement multifactor authentication, and we know it's coming for Twitter. So that's all good news. And we're assuming that it's going to be using the standard OATH technology that we talked about at length last week.

So, yeah, I mean, this is the - I think what happens is we're seeing a common sort of trend again where Twitter was in its infancy years ago. People got accounts, and they weren't important. So they used bad passwords, weak passwords, back when they didn't matter. Then, over the years, Twitter has become a force to be reckoned with, significant content carried, yet the accounts were never updated. So they're still using the weak passwords that you may have been able to justify five years ago. Now you really can't. Now...

Leo: I think you're charitable. I don't - I think you're being kind.

Steve: This is how it happens, Leo.

Leo: I guess.

Steve: This is why somebody still has "monkey" as their password.

Leo: Yeah, but there's no reason, if you're the Guardian, to have weak passwords.

Steve: But my point is that...

Leo: They thought it was meaningless. I understand, yeah.

Steve: ...unless somebody says, unless someone goes through and does an audit and a sweep to, like, force an analysis and change - and everyone's busy. They've got other things to do. So people say, how can this have happened? Well, it's just that once upon a time it wasn't important. We've seen this with personal computers where people used to say, oh, I don't really need any security on my PC because I don't do anything. Nothing's important. And then they begin doing their online banking, and important things creep into an environment that never had the security that it needed in the first place. So this is how people get into trouble.

Leo: Do you think - do they say that they admit it was a password hack? Or was there something more sophisticated?

Steve: The presumption is it was just guessed. Maybe a social engineering thing. They don't really tell people.

Leo: Yeah. I think there's something. I don't - yeah. I bet you, because you just, you should - if you're the Guardian, you use strong passwords, even on dumb accounts, if you're going to be the Guardian. I think they got hacked. They, you know, I got, from a friend, I got a direct message - it had to be because it was somebody I followed - saying, hey, you've got to see this, click this link. I clicked the link, and it went to tvvlitter.com. And if you look up close, you know, you don't look closely, it looks like Twitter. And it had a Twitter login page.

Steve: [Laughing]

Leo: And, now, I wasn't fooled by it, fortunately. But I bet you people are routinely fooled by that. And I think that there's a lot of - I think there's a lot of stuff like that. I really do.

Steve: Yup.

Leo: So I - who knows. We don't know how they got hacked.

Steve: Yeah. Living Social also got hacked. Now, in this case it was a loss of their entire database, 50 million names, email addresses, birthdates and encrypted passwords.

Leo: Unbelievable.

Steve: Yeah. And the good news is that the financial information was stored separately,

so it was not taken. But this was a full breach of their database with 50 million of their subscribers, their users' content stolen. Everyone was advised to change their password, of course.

Leo: Are they forcing them? This is similar to - it's happened now several times to various people, but...

Steve: Yeah. And it did - what I read was they were advised to change, rather than, I mean, and that's the same thought that I had, Leo, was it didn't look like all passwords were immediately rendered obsolete, and they were required to change them. So...

Leo: That's what happened with Dropbox and Twitter. It's what they said was happening with VUDU. But VUDU asks you to log in with your regular password first.

Steve: I know. Oh.

Leo: So I'm not - I'm thinking it's not exactly the same. You can call it a password forced change, but not really.

Steve: Right. Right.

Leo: So we'll see.

Steve: So there was something in the news that a number of our listeners sent that was worth, I thought, sharing. And it was covered by a number of outlets. And I found, like, the best story by Ryan Gallagher in Slate magazine, this was a couple days ago, about the FBI's decision to urge legislators to change the law to allow fining Internet companies who resist the FBI's requests for essentially monitoring people of interest. And I would paraphrase this except Ryan drops a bomb toward the end of this which took my breath away a little bit. So I just thought I would share this - it's not very long - with our listeners.

Ryan wrote, and this is in Slate, "Bad news for telecommunications companies: New details have emerged about the FBI's efforts to upgrade its surveillance powers, and the feds' latest idea is to heavily" - and get a load of the formula here later - "fine firms that don't comply with eavesdropping requests.

"Last month I reported that the bureau said it was having a hard time monitoring services like Gmail, Google Voice, and Dropbox in real-time when attempting to spy on criminals. The FBI's general counsel Andrew Weissmann revealed in a speech that a 'top priority' for the Bureau in 2013 was to reform surveillance laws in order to force email, cloud services, and online chat providers like Skype to provide a wiretap function. The 1994 Communications Assistance for Law Enforcement Act (CALEA) already allows the government to mandate Internet providers and phone companies to install surveillance equipment within their networks. But it doesn't apply to third-party providers like Google or Facebook, which has led the Bureau to claim that its ability to monitor suspected criminal conversations is 'going dark.'" And the FBI of course famously refers to it as the

"going dark problem."

"Now, according to the Washington Post, the feds have prompted a government taskforce to draft a proposal to update CALEA and the 1968 Wiretap Act to put more pressure on companies that do not currently fall under the scope of those powers. This could involve, the Post reports, 'a series of escalating fines, starting at tens of thousands of dollars, on firms that fail to comply with wiretap orders.' If a company fails to comply with an order in a set timeframe, it would 'face an automatic judicial inquiry, which could lead to fines. After 90 days, fines that remain unpaid would double daily.'" So...

Leo: I don't think that's so different, though, from, say, contempt of court fines at the judge's discretion. If you get a court order - which they're saying this is a wiretap order; right?

Steve: Yes.

Leo: If you get a court order and ignore it, you're always subject to contempt of court. And the judge can determine that.

Steve: The problem is the technology, though. One of the things that differs here between what was done before was remember that the Internet service providers were paid to have this equipment installed. And apparently here the idea is that they will - these third parties would be compelled to comply with the requirement, even if they don't have the technological infrastructure in place to do so. So the concern is that it could force them to implement this in a slapdash fashion.

Leo: Interesting. Even before there's a subpoena or a wiretap order.

Steve: Well, they would have to be ready for this.

Leo: Right.

Steve: You know, and we've already - we know, for example, that Google is - we talked about it last week - is actively resisting these sorts of things, saying, you know, no, we're not going to simply accept this.

Leo: But they don't resist court orders. They resist requests. They immediately comply on a court order.

Steve: Correct. Correct. In fact, the EFF has a great page that we'll be covering here in a minute, that shows who does what for us. Anyway, I'm just going to finish this real quickly.

Leo: Sure.

Steve: It says, "The FBI's controversial proposal is reminiscent of what other countries have recently considered. Governments in the United Kingdom, Canada, and Australia have each sought similar surveillance authority. Last year, the British government published a draft Internet snooping law that would have enabled legal action and penalties against companies that did not comply with surveillance requests. But the proposal appears to have been killed off due to political infighting and public opposition. Canada's web spy law was also canceled after an outpouring of criticism, and in Australia the government's surveillance plans have been delayed."

So we're seeing governments having a problem getting these laws through. And, I mean, we're sympathetic to the FBI's need to be able to watch what's going on in the case of bad guys. And we've talked often about the fundamental tension between that and individual civil liberties and our feeling of our right to privacy.

So finishing, this says, "If other countries' experiences are anything to go by, then, the FBI's efforts will certainly not have a smooth passage into law. Aside from privacy and civil liberties concerns, the Bureau will face tough opposition from companies concerned about the potential security risks posed by building in so-called surveillance 'backdoors' for monitoring purposes, which can be exploited by hackers. For that reason alone, the FBI can be sure that not all companies will play ball if it tries to rewrite CALEA in a way that would strong-arm companies into complying with eavesdropping. The CEO of encrypted communications provider Silent Circle told me" - the writer of this column, Ryan - "last year, for instance, that he would 'rather shut Silent Circle down than ever allow a backdoor or be bullied into an "or else" position.'

And then here's the coup de grce: "In the meantime, however, the FBI does have some options on the table if it wants to spy on Skype calls or get transcripts of Gchats in near real-time. The Bureau has a sophisticated spy trojan that can covertly infiltrate a computer to gather all kinds of data, take snaps of a suspect through their webcam, record passwords, and gather logs of conversations, as a judge in Texas disclosed last week in an order denying the tool's use."

Leo: Well, of course. If the bad guys have it, you figure the FBI just, hey, we could use that.

Steve: So reading that, I said, whoa, whoa, whoa, whoa, what? And it turned out that Ryan reported on this on April 25th. He said - and this is "Judge Rejects FBI Attempt to Use Spyware to Infiltrate Unknown" - and get this - "Unknown Suspect's Computer." And so on Monday of week before last, "a judge denied an FBI request to install a spy trojan on a computer in an unknown location" - and apparently, like, not known to the FBI, and I think that one of the problems with the FBI's request in this case was it was just too broad - "in order to track down a suspected fraudster. The order rejecting the request revealed that the FBI wanted to use the surveillance tool to covertly infiltrate the computer and take photographs of its user through his or her webcam. The plan also included recording Internet activity, user location, email contents, chat messaging logs, photographs, documents, and passwords." So now we know what this tool can do.

"As the Wall Street Journal reported, Houston magistrate Judge Stephen Smith said that he would not approve the 'extremely intrusive' tactic because the FBI did not know the

location or identity of the suspect..."

Leo: Yeah, that seems fair.

Steve: ..."and could not guarantee the spy software would not end up targeting innocent people. Smith wrote in a 13-page memorandum" - so anyway, this goes on to explain that now we know what the FBI's spyware trojan looks like, and that they apparently have some means, we don't know what - the sense I got was that they were going to send an email to an address that they had, and that's the reason that they didn't know who or where it was going to go. So essentially they wanted to - they were asking the judge for permission to infect an email account and then get evidence on who was behind the email account, which must have been anonymous. And so this would be, you know, this would de-anonymize them by taking pictures of them through their webcam and also rummaging around in the hard drive.

Leo: Here's the encouraging thing. And thank you, Judge Stephen Smith of Houston.

Steve: Yes.

Leo: The system worked. The judge did the right thing.

Steve: Yes.

Leo: The courts had to get involved. And this is why those secret courts, the FISA courts and lack of court supervision is so scary because...

Steve: Well, yes. And what Google fights is the request for information and stating in the request that, oh, and you can't tell anybody that we asked you to do this.

Leo: Patriot Act. Patriot Act requests.

Steve: Uh-huh.

Leo: And FISA courts and all that. The secret stuff is what - and the fact that this was exposed because a judge - because it's a public proceeding, is fabulous.

Steve: Yes.

Leo: That's what we want.

Steve: Yes.

Leo: So it did work in this case.

Steve: Some transparency.

Leo: Yeah, transparency, yeah.

Steve: So - oh, and of course this relates to the prior story by noting that, if in some cases it is not possible or convenient to intercept the encrypted data in transit, then the FBI potentially has the means to get it at one end or the other of the channel. So if you get the trojan on the individual's machine, then you're recording what their Skype is sending before it encrypts it and sends it out. So there's that alternative also.

Leo: Great.

Steve: So PayPal has become the latest single sign-on provider. They announced a couple days ago that they are now, too, supporting OAuth, which we'll be careful to differentiate from OATH. Remember that OATH is the standard one-time password technology that we talked about extensively last week and how it's looking like everybody is lining up behind that single standard, and that's really good. But they're supporting OAuth, which is the technology we've also discussed many times. That's, for example, where you go to a site, and it says "Sign on with your Facebook account."

We've talked about the concern of OATH hacking - I'm sorry, OAuth hacking - where, for example, if people get used to this, a malicious site could say "Sign on with your Facebook account" and bounce you, just exactly as you were saying, to a URL that, given a cursory observation, looks like it's Twitter.com or Facebook.com, but is subtly different, and prompt people to enter their username and password. So there's some - I'm a little nervous about that aspect of OAuth. I love the idea of minimizing the number of times that you need to create separate accounts across the Internet. And I like the idea of PayPal.

Now, what's interesting is that I watched - there's a video that PayPal has up describing what this is. And they're trying to spin it differently. They're saying, well, this is not a social networking single sign-on. We're a financial payment social sign-on. And so in their example they've got some, like, high-end designer baby clothes website that they use in their example. And so you go to this site, and you have the option of creating an account with them, or log on with PayPal.

And so what's not clear to me is how that's dramatically different from the pay with PayPal which we've had and enjoyed all over the Internet for several years. I mean, I love it when I'm going to - when I'm buying something from a site that I may never come back to again, I really don't want to create an account. I really don't want to give them all of my financial information. I would much prefer bouncing through PayPal and having PayPal transfer the money to them so that it's somewhat anonymous. PayPal will provide - I guess I provide my information. So this is a little bit more like Google because...

Leo: Yeah. It's a single sign-on. I think that makes sense because you trust PayPal for the payments. We'll take one step further and use it for your login, as well.

Steve: Yeah. And I think, though, that because one of the things that's been nice about Google Checkout is that Google Checkout will, for example, also fill in or populate or provide all of your shipping information. So you don't even have to fill that out. Whereas purchasing with PayPal used to be only the money, and you still had to provide that. But I'm sure...

Leo: Yeah, that's right. It was a separate log.

Steve: Yes. But I'm sure from this that PayPal will now provide the full population. So this is a little bit more like a response to Google Checkout, which does more of the work for you than PayPal used to.

Leo: Well, but it's more than that. It's also a response to Facebook and Google's single sign-on.

Steve: Yes, exactly.

Leo: And I think it'd be - the theory being, well, you trust us enough to give us all your financial stuff. So certainly you would trust us to take good care of your login, as well, maybe more so than Facebook certainly. And possibly Google. I bet you Apple, you know, the next one for this is Apple.

Steve: Yup.

Leo: Because Apple with iTunes also has all your information.

Steve: Oh, my goodness, yes. Yep. It makes lots of sense.

Leo: In fact, there's been some talk about the next iPhone having more identity features. You know that Apple bought AuthenTec, the fingerprint reading company, for more than $300 million. And there is some thinking, I wouldn't be surprised, some rumors that there'll be a fingerprint reader in the next iPhone. Now finally Apple's Passport might actually get credit cards. And suddenly be not only a single sign-on, but authentication, biometric authentication, second-factor authentication. That's an interesting, I mean, if anybody could do it kind of in one swell foop...

Steve: Yup. Yup.

Leo: ...it would be Apple.

Steve: Yup.

Leo: But that's all rumor. And you don't deal in rumor.

Steve: We don't. We wait.

Leo: We wait.

Steve: Okay. So that link there, the bit.ly link, you should bring it up before I mention it because we know what happens.

Leo: Done. Go ahead.

Steve: Really nice page. I created a bit.ly link [bit.ly/whyback]. "Why" stands for "who has your," and then "back." So it's bit.ly/whyback, Who Has Your Back. And the EFF put together a really nice summary page showing, from their standpoint, we know that they're really, really strong civil libertarian, user rights, user privacy protection. And we're glad to have them because somebody needs to fight back so that there's some balance of power here...

Leo: You bet.

Steve: ...in the inherent tension that we're going to have. There's some interesting little bits here. For example, MySpace - okay. So there are, for those who can't see it, there are six categories that companies are rated in: requires a warrant for consent; tells users about government...

Leo: For content, not consent. Requires a warrant for content.

Steve: Oh, for content, sorry. Warrant for content. Yeah, thank you. Tells users about government data requests; publishes transparency reports; publishes law enforcement guidelines; fights for users' privacy rights in courts; and, finally, fights for users' privacy rights in Congress. Now, I just did a quick little summary, noting that MySpace and Verizon are zero. No stars. Nada. They don't do anything.

Leo: Nothing. They don't do diddly.

Steve: Nothing good for us.

Leo: There's only one five - six-star, which is interesting.

Steve: Two. There are two.

Leo: Two. Oh, yeah, two.

Steve: AT&T and Apple don't do much, frankly. They both get one star out of those six categories.

Leo: The star they get for is lobbying.

Steve: Yes. Four companies - Dropbox, LinkedIn, Google and SpiderOak - are nearly everything. They get five stars. And then the two companies that are six-star winners are Sonic.net and Twitter.

Leo: Yay. Sonic is located up here in Sonoma County.

Steve: Very impressed.

Leo: My good friend, Dane Jasper, runs it.

Steve: Wow. I mean, that's not - it's not easy to get six stars on this because you've got to be...

Leo: Dane is very committed to this stuff.

Steve: ...fighting, fighting for user rights in court and lobbying Congress.

Leo: No, that's the only Internet service provider on that list with that kind of record. He's a small, independent Internet service provider. But not so small. He's slowly working his way nationwide, and I'm really proud of him.

Steve: Well, I'm impressed. So if our listeners are curious, bit.ly/whyback, who has your back.

Leo: Yahoo!, by the way, one star, only in the fights for user privacy rights in courts. Although that's a good one to have a star in.

Steve: Yeah, if you had to do one, you know. They may not be so - they're not busy

telling everyone what they're doing, but they're rolling up their sleeves when they have to. Meanwhile, our listeners continue playing with proXPN. And somebody assembled an interesting blog post with detailed instructions for configuring iOS's native, well, not native because it's not part of it, but it's an official OpenVPN client for iOS. And so I also created a bit.ly shortcut [bit.ly/iosopenvpn].

Leo: Oh.

Steve: Yeah.

Leo: Because we had to use PPTP before; right?

Steve: And of course the concern is, even though I would argue that it's not a huge problem that, as we talked about, you could upload someone's captured traffic and pay $200 to Moxie Marlinspike, and he would - I think that's who it was.

Leo: Yeah, I think it was, yeah.

Steve: And he would decrypt your traffic for you, given a couple days. I mean, it's way better in open WiFi to have encrypted traffic with a PPTP tunnel using effectively 56-bit encryption, which is not enough bits anymore these days. But still, it's better than zero. But the OpenVPN, the official OpenVPN project does have an iOS client. And this guy did a blog posting where he figured out how to, like, the details of the OpenVPN config file and so forth. So that's bit.ly/iosopenvpn, all lowercase, iosopenvpn. And that'll bounce you over to his blog post, for anybody who is interested. And then you have the benefit of full OpenVPN security, which is state-of-the-art enciphering and crypto.

Leo: That's cool. Very cool.

Steve: I had a couple little blurbs about one-time password multifactor authentication, or as we refer to it here, OTPMFA.

Leo: [Laughing]

Steve: Someone tweeted me, @SGgrc. He said, "I had to stop using Authenticator because over time they kept telling me my codes didn't match, even if I used the original setup key." And so I just, I didn't respond to that person. I don't think he was following me, so I was unable just to send him a DM, and I can't follow everybody because I'm up to about 35,000 followers now.

So I thought I would say on the podcast, in case he's listening, and for anybody else, if you're using time-based one-time passwords, it's important that your clock is correct. Because that's the synchronization mechanism. It's very clever because time, as long as we're all using the same reference, we're all going to have the same time. And individuals are then separately responsible for having their clocks correct. But if you do,

then suddenly everything works. And your little device, plus or minus the difference in your time, will be showing, will be generating the same key that somebody else with the same key as yours would generate given the same time.

So if you're a day, if your, like, calendar is wrong, if you've got the wrong year or the wrong month or the wrong day, then it's not going to work. And you wouldn't want it to work. But that's something I really hadn't explicitly said before that I thought was important to mention. There's the sequential code generators that we've talked about. But what's in vogue now are the time-based authenticators because they don't require a single central server, which the sequential code technology essentially does. They just require everybody is in agreement about what time of day it is.

So that's your responsibility. Otherwise, you can't authenticate. But generally that's pretty easy. Cell phone systems have that built in. GPS has that built in. There's NTP, Network Time Protocol on the Internet that allows your computers to synchronize. I mean, it's hard not to know what time it is. Maybe what day it is is sometimes confusing. But pretty much what time of day.

Leo: Anybody really know what time it is?

Steve: Yeah.

Leo: Anybody really care?

Steve: And then a really interesting comment was somebody asking about reusing their Blizzard Authenticator. And that brings up - that brought up another point that I had never discussed explicitly, was here's a problem. We're talking about - we talked about - there was one that I ran across last week that I shared that I liked, it was an iOS application, because it showed you all of your different OATH, the time-varying, one-time passwords, the six-digit guys, changing at once on sort of a big scrolling screen. And notice, though, that every provider has their own. So you'd have one for Apple. You'd have one for Microsoft login. You'd have one for Blizzard, blah blah blah. One for Google, of course, and so forth.

Well, so he's saying, well, wait a minute. Can't I reuse my Blizzard Authenticator? Well, now we're back into the sort of the equivalent problem of reusing the same password on multiple sites. The problem is that, if you did allow a physical, like a single instance of authentication to be used on multiple sites, that would require that you gave the same secret key to all of those sites. So all of them could simultaneously predict the same code showing in your authenticator, whether it's Blizzard or whatever.

And the problem with that, if course, is that, if one of them got their 50-million-user database stolen, with everyone's authenticator master secret, then suddenly they could attempt to use that to log into impersonate you on any other sites that you were sharing the same secret with. So we know that's a bad idea. The model that this is obsoleting, effectively, is the VeriSign model, where only VeriSign knows the Blizzard Authenticator master key. And then everybody who wants to authenticate to you asks VeriSign, is this the proper token? VeriSign looks it up and says yes. But no one who wants to authenticate gets your master key, only the result of a test for equality.

So but the good news is that's not the model that has ended up being dominant. I think

that's good because I just didn't like that concentration of power, and the fact that it was very expensive to ask VeriSign to do this. I mean, they've got a lot of infrastructure they have to support. It can't be free, and it really isn't. So here, at the cost of every single person you want to authenticate receiving a separate master key or negotiating with you a separate master key, the beauty of that is, even though you've got a scrolling list of authentication codes, none of them are the same. And if any one or two of them did happen to leak, all the rest of yours are still safe. So you really don't want to share and reuse those master codes. You want to - it's easy to make them. And it's kind of fun to have them all on a screen changing at once. So, yeah. I like it.

Leo: Yeah.

Steve: And yesterday was a big day, Leo.

Leo: Yes, I know. Historic, even.

Steve: Historic. And did you look at the page? The site was down most of yesterday, and it's up now.

[info.cern.ch/hypertext/WWW/TheProject.html]

Leo: You mean it's still online? It's not like an archive? Oh, that's neat.

Steve: Yes. They brought it back for the 20th anniversary. So you'll want to click there and poke around a little bit.

Leo: I did. It's open, yeah.

Steve: Okay. It's open. So this is the 20th anniversary was yesterday of the world's first world wide web site, the first web page. And it's sort of, I mean, it is, it's exactly what you'd expect. It's mostly text. I think it's all text, in fact. And it's, this is a hyperlink. If you click the hyperlink, then you'll go to a different page.

[Steve and Leo expressing surprise and delight]

Steve: And anyway, it's...

Leo: Remember that we were using Archie and Gopher when this came out. In fact, it was Gopher that this was kind of the most like, except that you could use a mouse and click a link.

Steve: Yes.

Leo: Instead of picking an item from a menu.

Steve: So 20 years it's been. Wow.

Leo: It's very Gopher-like.

Steve: 20 years.

Leo: It's amazing. It really is.

Steve: So our AskMrWizard.com has asked me to let everyone know he just finished another eight of his animated videos. And this is a very cool, classic Security Now! series, back on Episodes 25, 26, and 27. So this is February 2006, not long after the first web page was created.

Leo: [Laughing] It was close.

Steve: And I shudder to remember what my website looked like. This is Episode 25 was How the Internet Works, Part 1.

Leo: Oh, neat.

Steve: And those were three classic episodes - 25, 26, 27. And Bob is going to fully animate all three of those episodes with a large series of video because he considers that this was the foundation upon which so many of our subsequent podcasts had been based, as we've talked about the technology of the Internet and how it works and what happens when it doesn't and so forth. So it's AskMrWizard.com. There's a link there, or you can go /securitynow to go right to the Security Now! page. And it's Episode 25, How the Internet Works Part 1, and a bunch of videos there.

[askmisterwizard.com/EZINE/SecurityNow/SN025/SN025All.htm]

And I just thought I would update everybody on my own research that I've been working on the last week on certificates, the public key infrastructure certificates. I've been continuing to do research following the release of GRC's fingerprints page [grc.com/fingerprints.htm], which has become very popular. We're getting about 1,500 users a day. It's sort of settled down to that. The idea being, remember, that this allows you to check the fingerprint hash of certificates that you receive from remote websites from your vantage point and compare them to GRC's vantage point, in order to detect anyone who might be intercepting your connections.

One of the things that I realized, I think this was - I may have mentioned it last week, but I wasn't sure about it. I remember this was - I was beginning to rant on Internet Explorer last week, and believe me, they ended up deserving it - is the potential power of extended validation certificates. The EV certificates are those ones that all the browsers

give extra attention to. GRC is an EV site. And in fact, probably by this time next week we will be 100 percent SSL. I'm going to make the switch to HTTPS everywhere, essentially force the display of all of our pages to fully secure, just so that you always know that GRC should be EV. That's important because, for browsers that have not totally screwed up extended validation, and as far as we know only Internet Explorer has completely rendered it useless, all the other...

Leo: Geez.

Steve: I know. It's unbelievable. But I've read the source code of Firefox and Chrome/Chromium. And they both did it right. We have every reason to believe that Opera has done it right because they're really security conscious. And I assume that Safari has, but I have not been able to verify it. The problem is, both in the case of Opera and Safari, I don't have the source code, and I don't have good contacts to, like, the actual guru who could tell me what they're doing. I've reached out to Opera but haven't heard anything from them.

Leo: Can you presume, because they're using WebKit, that whatever is in WebKit, which Chrome uses also, as well as Opera and Safari...

Steve: No.

Leo: No, it's something on top of?

Steve: Yeah. What Chrome does, Chrome has an explicit policy of using the security foundation of whatever platform they're on. So, for example, when you're in Chrome on Windows, and you say "browse certificates" or "show me certification information," you get the Windows certificate UI that is identical to what Internet Explorer shows you on Windows. If you're in Chrome over on Mac, on the Apple Mac platform, OS X, you get exactly the same dialogues that Safari shows you. So what the Chrome guys have done is they've tried to use the existing security and certificate infrastructure. Now, in the case of Windows, Chrome did the right thing. So here's what's going on, is the way extended validation certificates are supposed to function is they're supposed to actually mean something. I mean, they're supposed to...

Leo: Well, that seems fair.

Steve: Wouldn't it be nice? They certainly are - they mean a lot to my pocketbook because I'm paying for them. In response to that, there is much more verification of my and my company and my website identity going in, before in my case DigiCert checked me out and validated me and issued an extended validation certificate. They are always for a shorter length of time. You cannot get them more than two years. The actual policy recommends one year. But it's like, oh, no, please don't make me do this every year. So it is two years. But no longer than that.

The weaker form of certificate is typically called a DV, Domain Validation. And there are even some, like StartSSL is a certificate provider, that'll just give you a free one for a

year, and they just sort of say, okay, put your thumb on your wrist. Do you feel a pulse? Okay, good. We'll send you a certificate. So we would really rather have more verification than that. Extended validation is that. But the question is, how do we keep it from being spoofed?

And what is so cool about both Firefox and Chrome for sure, but absolutely definitely not for Internet Explorer, is that they have in their code, in the code of the browser, is the hash of the EV root at the beginning of the chain of trust that signs the certificate. So in order for the green EV to turn on in Firefox and in Chrome, and that's Chromium whether it's over on Safari or Linux or anywhere because it's built into the code, I've carefully read the source code in Chrome, is there is a serial number, it's called an OID, an Object Identifier, in the EV certificate, for example, that I have, that I received from DigiCert. Firefox looks at this OID, this Object Identifier, and then looks up in its internal database, that is, it explicitly does not use the public key infrastructure, this is outside of PKI, because unfortunately we can't trust PKI enough. It is subject to manipulation.

For example, somebody could put fraudulent certificates in your root store of your computer, and we know that happens. In fact, two AV utilities, BitDefender and Kaspersky, do this. They intercept all SSL HTTPS communications on behalf of their users, and all EV verification disappears because they're in the connection, and the certificate that the browser receives is not the authentic one. So that's one of the consequences of using those features in those AV tools.

But in this case, for example, with Firefox and Chrome, they verify the hash of the root signer of the chain of trust of the certificate, and only if it matches will they turn this on. By comparison, Internet Explorer is broken completely. It is useless. There are pages on Microsoft's site showing companies how they can give their own websites extended validation green coloring, just by clicking a few buttons.

Leo: What?

Steve: Yes. It's unbelievable, Leo. It's like, oh, look. Everyone wants to have EV. So...

Leo: Simulate it.

Steve: It is, it's completely broken.

Leo: That's horrible.

Steve: It's unbelievable. It's like, oh, wouldn't you like to have that on your Intranet site? And one of the great contributors in the GRC newsgroups did in fact run a simulation of creating a fraudulent certificate and got IE to turn green with a certificate that he made himself.

Leo: Self-signed.

Steve: So it is, yes, it is completely useless. Internet Explorer's indication of extended

validation means nothing, unfortunately.

Leo: Wow, wow.

Steve: Yet the good news is Firefox and Chrome did it right. And I would love to hear, if there's anyone who has contact with Opera, I'd love to know what they are doing. And the same goes for Safari. We have to assume nothing until we know for sure. But so the point is, if you're using Firefox or Chrome, and this shows green, then what you are guaranteed of, you don't even need fingerprint matching, in those browsers you are guaranteed there is no middleman. There is no man in the middle. There is no interception of your traffic.

Now, the problem is not everybody is using extended validation yet. The good news is banks tend to be. So you may find that your bank is. GRC is. So if you're using those browsers, and you go to my site, to GRC, and you go to the Fingerprints page, I am now showing which sites use EV. So the point is the problem is, since not everyone's using EV, you don't know if you should be getting an extended validation indication from any random website you visit. But you can go to GRC, to the Fingerprints page, put in the site, bring it up, and I will show you if that site should be extended validation. And then, if you go there, you absolutely want your browser to show you extended validation from that site. And if it's not, then something is in the way. And if you're using Internet Explorer, don't bother with any of that. It is broken.

Leo: Well, that - okay. So Microsoft's been making - and this is Internet Explorer 9 and 10?

Steve: All of them. It's, I think, from 7. From 7 on they brag, there's like all kinds of links showing you - and I'm going to do a full page to explain this and cover it on my site. And I will link to these things where Microsoft is saying, oh, wouldn't it be fun to have extended validation on your own web servers? Here's how you do that. And essentially you can just make some changes to the registry to clear your own certificates as EV, and then IE turns green.

Leo: Wow.

Steve: Which means it means nothing. It's completely broken.

Leo: That's a really strong argument against using IE, I've got to say.

Steve: It really is. I mean, it's awful.

Leo: Yeah [whistling]. Unbelievable.

Steve: And finally, I keep seeing people asking about recovering SSDs with SpinRite. And so I found a great testimonial where Robert Osorio, who's a listener of ours, asked

the question and referred to the podcast. So I just wanted to share this once again because he does a great job of covering it. And he said, "Steve, just to let you know that you can add me to the list of SpinRite users who have found SpinRite useful for reviving SSD drives. I'm an IT consultant and have been using SpinRite for a couple of decades." So, yes, back actually before the first web page existed, we had SpinRite, because it is more than 20 years ago that SpinRite 1.0 was created.

And he said, "I have an older Intel X25-M SSD drive that was the boot drive from my main workstation. I recently upgraded to a much faster SSD and relegated the old Intel drive to a laptop. However, in time, I started getting OS issues that, on a spinning drive, would have indicated bad sectors and would have had me running SpinRite on it immediately. Since this was an SSD, I thought all I could do was update the firmware - which I did, and it did help for a while - or just write off the drive.

"Then I heard you mention on a recent podcast that running SpinRite Level 1 on an SSD could help, so I gave it a shot. It made a dramatic difference, and now this drive is running smoothly. I have now run SpinRite Level 1 on all my SSDs and will continue to do so on a regular basis for preventative maintenance." He said, in parens, ("I religiously run SpinRite Level 4 on all my spinning drives every six months or so, as well.)"

He said, "I did want to get a clarification from you, and I'm sure other listeners would appreciate this, as well. You recently read a testimonial from someone who recovered an unreadable flash drive using SpinRite on Level 2, and you indicated that was a valid procedure. Am I correct in assuming, then, that it's okay to run Level 2 on an SSD or flash drive for preventative maintenance? Or should I use Level 1 for preventative maintenance and Level 2 for data recovery only?

"My concern is avoiding excess writes, which would prematurely wear out the memory cells, thus your admonition against running Level 4 on solid-state media since it performs aggressive writes. Reading from your documentation, it appears that Level 2 is only performing writes if it recovers data from a damaged sector and then has the drive relocate it. As such it seems that Level 2 is not much more aggressive on writes than Level 1 and should be safe to use on a regular basis on SSDs. Thanks again for a great product and a great podcast."

And, yes, Robert, that's exactly right. So the difference between Level 1 and 2 is that 1 is prohibited from ever writing. I just sort of thought, back in the dawn of SpinRite, that that might be useful for some reason, like you could - there were instances back then where, if a drive was really misaligned, then writing could rewrite the data in the misaligned location and cause more trouble. So I thought, well, let's just have the option of an absolutely read-only phase. So 1 absolutely will not perform a write command on the drive. But Level 2 is essentially the same. It does a read pass of the drive. And, if there's a problem, it performs data recovery and will then rewrite the recovered data and only the recovered data back into that location. So it is very gentle on SSDs, and that's what I would recommend actually over Level 1.

The reason he got improvement with Level 1 was that running SpinRite on the drive, as we've discussed before, showed the SSD that it was having problems and allowed the SSD to fix itself and perform hidden relocation of its sectors. And in his case it wasn't necessary to perform any writing to it, it just sort of said, look, wake up. Here's what's going on. It's time to take some action. And the SSD did. But Level 2 is generally what I would recommend. And it does perform recovery and even speeds them up.

Leo: There you go. From the horse's mouth. I mean, no one knows how SpinRite works better than that guy. He wrote it. In assembly language, my friends.

Steve: Yup.

Leo: So BitTorrent we all are very familiar with. I think sometimes people think of it as a pirate service. That's not the case. In fact, when we first started doing podcasts, this show and TWiT, we distributed them via BitTorrent because I couldn't support the bandwidth for the thousands of downloads. Still is probably the No. 1 way Linux distributions are sent out. But it's also used for piracy, and I think BitTorrent is at great pains to kind of shed that image. And they've been adding some interesting features. And this is the newest, BitTorrent Sync.

Steve: Well, yeah, there's a BitTorrent Labs that sort of works on various projects that are related to their technology and leveraging the wealth of experience that they have with peer-to-peer problems. We've talked about some of the challenges associated with hooking up two machines in a direct peering relationship located anywhere on the Internet. You and I are conversing right now over Skype, which is a peer-to-peer network.

Leo: Right, good point.

Steve: And we've established a direct UDP protocol connection between our two computers, despite the fact that we're both behind NAT routers. Now, we did learn that it is better to map a port through and assign a static port so that we avoid the sometimes necessary relay of our traffic which results in far less useful connections between us because of just latency in a real-time stream that we need in order to have the quality that we want. You really want to minimize jitter. And so a direct connection is the way to do that.

But it is a challenge to establish a direct point-to-point connection in the modern network. If in fact the Internet was different, if it turned out to be the way the original designers intended, where every endpoint on the 'Net had its own unique IP, and there were no NAT routers, then this problem would have been much easier to solve. But that's not the way things turned out. And in fact we talked a couple weeks ago about an ISP that's I think going to be moving, by default, a large class of its users behind their own ISP-level NAT router. So the idea of NAT traversal is of increasing importance.

What BitTorrent has done, and the reason it's exciting to me, is they've done this in several places, from a technology standpoint, I think exactly right. Which is what they have made available absolutely free. There's no gotchas. There's no we want to install McAfee Security Scan or the Ask Toolbar or any of that nonsense. This is absolutely free. I should mention it's currently at alpha stage, although upwards of 20,000 people and more are using it and loving it. They've got an active forum. So remember, it is still in its early stages. And even since its first release, it's acquired a bunch of additional features, so features are still being rolled out.

What I like about this is that what you download is absolutely independent of any third party. This does not require a phone home to BitTorrent or to anything. It is a

freestanding peer-to-peer client which leverages all of BitTorrent's proven technology, adding a layer of security, of explicit security that I'll talk about in a minute, which has never really been important for the normal public BitTorrent network because it was all about sharing files with people you didn't know, and it was on you to make sure that what you got was not malware, and it was what you were looking for.

So in their explanation of this, they explain that it is unlimited, secure, and we'll define some of these terms here in a minute. But unlimited secure file syncing. Useful for remote backup. Useful for transferring large folders, I mean unlimited-size folders, of personal media between users and machines, editors, collaborators, whatever. And I'll explain what the security model is and what the granularity model is so that people understand how much flexibility they have. I think this is going to be a win. I think this is going to be very popular.

Leo: Well, and it's free, which I think is...

Steve: And it's fast. The reports have been this thing screams.

Leo: Really.

Steve: And, well, and there's no cloud involved. So unlike Dropbox, it's not going somewhere else and then having to come back down.

Leo: But I also don't have to worry about somebody else having it.

Steve: Precisely. Nobody else ever sees it. So I'm in contact with the guys there. They're aware of this podcast. I've had some interchange just this morning back and forth with the right person. What I don't yet have is the full technical readout at the level of detail we need. I gave them the example of LastPass, where Joe was so forthcoming, like we got the complete technical readout on the technology, so I was able to audit it. We were able to look at it and say, yes, I can bless this. This is absolutely done right. This looks like it's absolutely done right. And I think we're going to get all of the technical details because they seem very willing to share that. I don't yet have that.

Leo: So I'm installing it now. And you can either do the standard setup or say "I have a secret." Secret would be because you set it up on another device. This is really elegant. I think this is - boy.

Steve: Oh, I know, Leo. It's going to be a great solution. So you keep playing with it, and I'm going to keep...

[Talking simultaneously]

Leo: Yeah, I'll have it installed by the time...

Steve: [Laughing]

Leo: Oh, I just showed a secret. I should go back.

Steve: That's all right because...

Leo: It's okay?

Steve: You can spin those any time you want, yes.

Leo: Okay. I'm going to make a new secret.

Steve: So they talk about how it's unlimited and fast file syncing. They compare it to Dropbox, but secure. Dropbox of course is famously, eh, not so much. No third party. No storage limits. No limits of any kind. I should mention I did see a mention of a million files.

Leo: Okay, that's probably enough.

Steve: That's enough. In one folder. I think it kind of begins to bog down and consumes a lot of memory if you give it a folder with a million files. But fundamentally no limit. It is a folder-syncing utility, and they say the bigger the better. It is designed for huge, uncompressed files. So, I mean, it just loves big files. I remember seeing that there's a 4MB blocking factor. So files smaller than 4MB are just treated as one big blob, and they're immediately shot out to all the synced folders when that file appears in a folder. Files way bigger than that are chopped up into 4MB blocks, and they're synced at that level. So if one block of a much larger file changes, only that one 4MB block is resynced.

Now, in addition - so the model is that of a secret. You mentioned the secret before. When you are setting up a folder to share, BitTorrent will randomly, pseudorandomly generate a large text string which is the secret. And there's, you know, I want every detail. And so sometimes they - and they're sort of - they made this easy to understand, and they sort of dumbed it down for people on their website.

Sometimes they talk about a 21-byte key which is then base-32 encoded in order to make it readable by people. So 21 bytes is 168 bits, which is like Triple DES. But don't worry about that because this is all AES encryption. Except they use a 256-bit AES key, they say, somewhere else. So the secret seems to be 168 bits, but somehow we need 256. So maybe, and they talk somewhere about the key being the root, that is, the secret being the root of the key. So there's another 88 bits we need. So maybe - we just don't know. They could be hashing. They could be appending 88 bits to the 168-bit secret-derived key. It's not clear. But the good news is everything I'm seeing says they have implemented good security.

So you create this long pseudorandom secret. You don't have to use theirs, either. You can use your own. But there the advice changes, and they talk about a base-64 string that should be longer than 40 characters. So I need the technology from them in order to

exactly understand what's going on, in order to be able to describe it to our listeners, who care about and understand this kind of stuff.

Leo: You get some interesting things, too, by the way. You get just the regular secret, but a read-only secret, and then a one-time secret.

Steve: Yes. Now, the way you...

Leo: Now, presumably I can just change these at any time.

Steve: So the way those work is a read-only secret is something you would - okay. So first you create a secret for a folder. Now you then install this somewhere else, or you want to share this with a - that folder with a friend. So you arrange to give them the secret. There's no username. There's no password. And they make the point, and I agree, that these pseudorandom secrets are stronger than a username and password. And in fact this is the same technology I ended up settling on for CryptoLink when I was doing my brainstorming for what I was going to do there. CryptoLink was going to essentially embed a pseudorandom key and then also generate one for the user. I mean, that is the right way to handle these kinds of encrypted connections.

So the idea would be you give a buddy the same secret. Now, yes, it then is incumbent on us about moving that secret to the other person in a secure fashion. So you could use something like AxCrypt, or you could chop it up in pieces. You could fax it to them, blah blah blah. They talk about base-32, which I assume they chose because that's probably case insensitive. They don't say that anywhere, but base-32 means that we would be using 5 bits out of a character. So probably A-Z, independent of case, gives us 26 possibilities, and then 0-7, for example, gives us 8. So 26 and 8 - what? Wait. 26...

Leo: Math is hard.

Steve: 26 and 8. Somehow you need 32. And so that means that the secrets are probably case insensitive. So you could fax it to somebody. You could dictate it over the phone. You could encrypt it using AxCrypt and then email it to them, somehow get that to them. Then what happens is the clients in the network are able to find each other based on the shared secret. So this is a shared secret technology. Because you've got two BitTorrent sync clients, no matter where they are in the world, if they have the same secret, they are able to find each other. And the traffic that they exchange goes directly, point-to-point, between them and is encrypted using a key derived from that secret.

So that was another one of the key phrases I saw that leads me to think these guys did it right. They're not actually using the secret directly. They're deriving the actual interchange key from the secret. We don't know how. It would be nice to know how. So that's the normal, full-access, read-write secret. It's also possible, as you saw, Leo, to give somebody a - they call it, a little confusingly, one-way synchronization. That's, I mean, what it really means is it's a read-only secret, meaning that you give it to somebody, and they use it to synchronize a folder that they have. But it's read-only, which is to say one-way sync. They're able to - their folder will contain everything your folder has, but it doesn't go in the other direction. They are unable to change the contents of yours. So that's very useful.

Then they have something they call a one-time secret, which is also a little odd. The idea is you're able to generate it on demand, and that one-time secret can be either a full read-write secret or a read-only secret. And that you give to someone with whom you don't want to share the master secret. And the limitation on it is that it must be used within 24 hours. Now, they don't explicitly say it except one place. And again, it'll be good to have this all clarified. But what it sounds like is the secret, it's not that the secret - the key dies after 24 hours. But when it's used, it allows the holder within 24 hours to obtain some sort of a master secret copy from the master secret holder. And that does not expire.

So there also doesn't seem to be sort of like a secret management system. I got the sense that if, for example, if you ever wanted to rekey, you would need to change the master secret on all of the folders that you wanted to keep synchronized. That is, so there's no way to, like, revoke somebody's rights. Which I think is a reasonable tradeoff. This system could get very complicated pretty quickly. You can synchronize as many folders as you want. You are able to exclude folders and files using wildcard pattern matching in a file that you put in the folder, saying, you know, and so the client will see that and not synchronize files that you specifically tell it you don't want synchronized, either explicitly by filename or by wildcard matching.

There is an Advanced tab that gives you the ability to enable or disable Universal Plug & Play management of your router. So as is the case for the normal full-strength regular BitTorrent client, the BitTorrent Sync client is able to communicate with a UPnP-enabled router and ask it to open a port through to it to help with NAT traversal. But you can disable that if you like and set up static port mapping. You can assign BitTorrent a port to use and set it up that way. It is cross-platform.

Oh, I should also mention that you can establish explicit bandwidth throttling. The normal case is just to use all the network bandwidth. People in the forums have talked about setting it up on their LAN and experimenting with it. And they drop monster files in a folder, and it just, bang, appears in the other machine almost immediately. Small files are instantaneous. And in one case a 3.5GB file took a minute or two to transfer.

So clients on the same LAN will discover each other using a LAN broadcast, again, all tied to the secret. The nice thing is the secret is the master key for this global BitTorrent sync network. And that's the way everything is glued together. So you do have bandwidth throttling, separate for uploading and downloading, if, for example, you just want your data not to - or if you're, like, synchronizing huge files, but you still want access to your network while this is going on, or you're trying to do online gaming or something. We've talked about how network saturation with buffer bloat can ruin real-time gaming performance. So you're able to put in a throttle. There's no schedule-based throttling. But those are the kind of things we may get in the future. And I have seen requests for that in the forums. And no quality of service management yet, but that's the kind of thing these guys are clearly able to do.

It's multiplatform. It requires on XP you have to have SP3, and only 32-bit of XP. It will not run on the 64-bit version of XP. It will run on both 32- and 64-bits of Vista and Windows 7. Runs on Mac OS X from Snow Leopard on. Runs on Linux with kernel 2.6.16 and later, for the ARM, the PowerPC, the 386 and the 64-bit platform. So they've got that covered.

Leo: That's good news because it means it probably could be ported to mobile

devices quickly. It currently doesn't...

Steve: It's coming. It's coming.

Leo: Yeah, it doesn't have any mobile support right now. And that's...

Steve: No, it doesn't.

Leo: ...an issue.

Steve: Yes, although it's on its way. It has absolutely committed, they are working on iOS, and there will be iOS and Android clients. So those are coming. And it was supposed to be this week, but I didn't see it, because I found a posting from the BitTorrent guys late last week, a lot of people were wanting it on FreeNAS. It is compatible with Linux-based NASes, and they have said they have it running on FreeBSD and PC-BSD, which brings it out on - which means that it will be running on FreeNAS.

Leo: Having it on a network-attached storage device is very intriguing.

Steve: Oh, Leo, it's perfect. So now, so the idea would be you stick it up on FreeNAS...

Leo: You can have NASes synchronize with each other.

Steve: Yes, yes.

Leo: Offsite and onsite.

Steve: Yes.

Leo: This is going to really put a lot of companies out of business, I think.

Steve: I think it's wonderful. You set it up with network storage. You map a static port through your router. Now everyone is able to find it. The peer-to-peer discovery, basically they're leveraging everything that they've learned from their years of using BitTorrent. I mentioned that a local LAN is able to use subnet broadcasts in order for clients to find each other. When two peers discover each other, they then immediately exchange their knowledge of all other peers. So the network tends to be strong and robust and sort of self-healing.

If you have a known, like, static port, static IP and port number, you can use the UI to set that. So, for example, if you - and I didn't see any support for Dynamic DNS. That

would be a nice thing to add because many NAT routers will support DynDNS. And then you could use DynDNS and a port number so that any roaming clients always could find your NAS at home for syncing. But so you're able to use it that way. They do have a distributed hash table system within the BitTorrent client network. And you can also use a BitTorrent tracker in order to help find each other and to act as a...

Leo: Oh, how interesting. You can use a tracker?

Steve: Yes.

Leo: Hah. Wow.

Steve: And they actually recommend that over dynamic hash tables, just because it's faster, and it helps with NAT traversal.

Leo: This is going to be - this could be a wonderful piracy tool.

Steve: And as the absolute last resort, they will relay traffic.

Leo: Really.

Steve: If there's no way for two clients to connect directly, and that's really what you want for speed, but if it just - if it can't happen, then they will relay your traffic, your encrypted traffic for you. They have no idea. For them it just looks like pseudorandom data because everything is protected by apparently a per-session negotiated key based on the shared secret. So I can see people synthesizing really ad hoc complex networks of arbitrary folders shared among their devices, their offices and satellite offices, home and work, and among their friends, simply by generating these large pseudorandom keys, sharing those, and suddenly your folder contents is shared. I mean, there have been times I've wanted to send Jenny something big, and there just isn't a good, easy way to do that. Well, now there is. And it's free, and looks robust, and it looks like they've done security in the right way.

Leo: Traffic is encrypted.

Steve: Yup.

Leo: So it doesn't matter where you're using this. You don't need a VPN tunnel or anything like that.

Steve: Nope. So, for example, you're at Starbucks, and you've got this running on your laptop, and eventually on your mobile device. And you simply - you're looking at a folder which is kept synchronized to your folder at home. And it's peer-to-peer over the

Internet. The finding each other has all been worked out and solved. And essentially you're creating little independent networks that are not reliant on any central phone-home headquarters.

Leo: Wow. Is it open source?

Steve: Not open source.

Leo: Interesting.

Steve: And I'm okay with that because - as long as it's open protocol. I mean, for example, you know, Joe at LastPass just was completely open about what he's doing and how it works. And it was because of that that I was able to analyze everything he did and say, okay, I'm using this. They nailed it. I don't doubt that these guys have, too. But all the words they're using, all of the phrases and so forth, look like they've done it right. But it would be nice to have all the I's dotted and the T's crossed, to know, like, what's the length of this, how are the per-session keys derived from the shared secrets.

Leo: Is there an API? I mean, could you write an open client?

Steve: Don't think there is yet.

Leo: There's not an API.

Steve: And I don't - and again, remember, this is still very early. This is only a week old. It just came out...

Leo: It's from their labs, yeah.

Steve: Yeah.

Leo: And it's already quite mature, in a way.

Steve: I know, it looks great.

Leo: I've just installed it. What a great way to share stuff fast and easily with yourself or others.

Steve: Yup.

Leo: Very cool. So you get it, it's kind of a long URL, but if you go to labs.bittorrent.com, I think you'll find it. I just googled "BitTorrent Sync."

Steve: That'll - it'll get you right there.

Leo: Got me right there, yeah. Very good. Very good.

Steve: I think we have a hit on our hands, Leo.

Leo: Yeah. Yeah, and if I were Dropbox, MegaUpload, some of those people, I'd be a little worried. I don't think that the consumer products have to worry too much because this is a little geeky.

Steve: Once upon a time the cloud providers offered the allure of terabytes of storage. Now everybody has terabytes of storage. I mean, what we need is to get this stuff synchronized, and this provides that.

Leo: Right. Very neat.

Steve: Yeah.

Leo: Steve Gibson's at GRC.com. That's where he hangs out. That's where you'll find BitTorrent - BitTorrent. No. SpinRite. There's no BitTorrent there. SpinRite, the world's finest hard drive recovery and maintenance utility. All the freebies he gives away, his passwords, Perfect Paper Passwords, padding and all of the stuff, all the security information. Of course ShieldsUP!, which you really should use right now for the UPnP tester.

Steve: Yeah, I think we're on the high side of 5,000, we were getting close last time I looked, different people who had Universal Plug & Play exposed on the public WAN side of the network.

Leo: Vulnerable systems, yeah.

Steve: Yeah.

Leo: He also puts 16Kb audio versions there for people who really don't have a lot of bandwidth. It's not the greatest sounding, but it's there, the content's there. And the smallest version's the English-language transcript that he offers, as well. That's at GRC.com. We have, of course, larger files, the higher quality audio and video, available at TWiT.tv, which is down right now, I know. It'll be up momentarily. We're

just doing a little maintenance. TWiT.tv/sn for Security Now!. You can also subscribe. In fact, that's a good idea if you want to get every episode. All the podcast aggregators support Security Now! and all of our TWiT shows. We do this show every Wednesday, 11:00 a.m. Pacific, 2:00 p.m. Eastern time, 1800 UTC on TWiT.tv, so you can tune in and enjoy live, if you want.

Steve: We'll have a Q&A next week.

Leo: GRC.com/feedback for the questions.

Steve: Yup. Send me questions. Maybe play with BitTorrent sync and, if you've got some questions, let me know. I will - I don't know how soon the guys there will get a protocol white paper assembled. But I've got a dialogue open with them. And they seem accessible to that. So I think we're probably going to get that.

Leo: Cool. Thank you, Steve. What a great subject.

Steve: Thanks, Leo.

Leo: Thank you for joining us. We'll see you next time on Security Now!.