SECURITY NOW!

Transcript of Episode #734

The Joy of Sync

Description: With this week's "The Joy of Sync" podcast, we focus upon the latest state-of-the-art secure solutions for cross-device, cross-location device synchronization. But before we delve into that abyss, we'll update on Mozilla's recently announced plans to gradually and carefully bring DNS-over-HTTPS to all Firefox users in the U.S. It turns out it's not quite the slam dunk that we might imagine. We'll also check in with the EFF to see what they think, and remind our listeners about the 100% free VPN offering coming from our friends at Cloudflare.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-734.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-734-lq.mp3

SHOW TEASE: Hey, everybody. It's time for Security Now! in a special episode. Folks, this is one you're going to want to download and keep. You're going to want to get the show notes, too. Steve Gibson - we'll talk a little bit about some new stuff in Firefox and so forth. But the bulk of this show is about Steve's research on file syncing, something more secure, more private than Dropbox or Google Drive or OneCloud/Drive or any of those. Steve says he's found the answer. "The Joy of Sync" is next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 734, recorded September 14th for airing October 1st, 2019: The Joy of Sync.

It's time for Security Now!, the show where we cover your security and privacy and, today, data synchronization with this guy here, Steve Gibson of GRC.com. Hi, Steve.

Steve Gibson: Leo, great to be with you once again for, well, in this case we're actually - we're sneaking in an early record. So we do not have lots of security news because while we're recording this I have no idea what has been going on recently. However, I do promise that the next podcast, 735, is going to be a big mega catch-up, and we will catch everybody up on anything that happened while Lorrie and I were traveling around Europe, meeting with OWASP groups and telling them about SQRL.

Leo: Nice.

Steve: Which is actually where I will be on October 1st, which is nominally the date of this podcast. And of course you and I are going to be, two days later, on Thursday, in Boston for the LogMeIn/LastPass identity over the Internet panel.

Leo: Yeah, this is going to be fun. I'm really looking forward to that panel. And I think it's probably too late to get tickets to that now. But just in case, if you want to see if there's any tickets left, twit.to/unlocked is the short URL. It is this Thursday. You need to be in Boston Thursday, 3:45 p.m., at the Intercontinental Hotel. Steve and I and Bill Cheswick and Gerry Beuchelt, who is the CISO for LogMeIn will be talking about the problem with passwords. And I suspect we'll get a little SQRL in there.

Steve: I think maybe we will. And we will definitely - apparently there will be sort of a cocktail hour.

Leo: Yeah, there's a reception afterwards. So we're going to be onstage for about an hour and a half. We'll take questions for half an hour. So it'll be an hour of us talking, half an hour of questions. And then we'll be around for an hour or so after that to talk individually with people. And that's going to be, I think, the highlight of it. I can't wait.

Steve: Very fun. I really enjoyed that when I did the OWASP Orange County group. I mean, it was basically a roomful of Security Now! listeners, so we had a lot of fun.

Leo: I think that's what it's going to be this time.

Steve: We also had people bringing their SpinRite CDs for autographs.

Leo: Oh.

Steve: So I autographed a lot of SpinRite CDs.

Leo: Bring your SpinRite CDs, kids.

Steve: Lorrie said, "People still use CDs?" I said, well, it's not easy to autograph a thumb drive. You know, there's just not really enough room there to...

Leo: They are an excellent substrate for a Sharpie. They really do [crosstalk]. So this week we wanted to do something that was a little more evergreen because we are recording it a couple of weeks ahead of time. But this is a project you've been working on for some time.

Steve: For several months now. So this week's podcast we titled "The Joy of Sync." And I talked about this a couple months ago, where I was struggling with operating in two different physical locations. I've maintained where I'm podcasting from. Everybody knows that hasn't changed in 15 years. It's the way it always has been. But I'm now spending my evenings in a different location with Lorrie. And we normally both - she's busy in the evening, and so I get another burst of opportunity to work. So I was copying stuff onto a thumb drive and using sneakernet, even in this day and age, or sticking things in a Dropbox folder and then bringing them down. But basically I didn't have automation for keeping things dynamically synchronized. And so I decided finally, okay, this is crazy. Let's use some technology here, Gibson, and figure out what we should do.

And so you and I talked about this briefly, about ownCloud, and you pointed me at Nextcloud, and I had sort of settled on Dropbox. But I was uncomfortable with the fact that it wasn't encrypted storage. I mean, it's encrypted at Dropbox, and it's encrypted in transit. But if you read the fine print in the terms of service, it says that they're able to look at your files to verify that it doesn't violate their terms of service. And the other cloud providers do the same thing. So we're going to get to all that, and also to the solution that I have found. I have something, actually two different tools and a trick for using one of them, that I am really happy with. It's completely solved my problem. It's TNO, which of course in fact it was for cloud storage back in the Jungle Disk days that we coined that term, that acronym, TNO.

Leo: Oh, really. I didn't realize that. Okay.

Steve: That's what it was. It was Trust No One, the idea being to encrypt it on your machine before it leaves. And of course that creates some challenges because one of the things that many of our listeners have said they want the ability to do is to selectively share specific files with other people. But if everything that's stored in the cloud is encrypted, then how are you able to offer, like selectively let people have specific things? And then of course there are people who don't have an interest in cloud stuff, but they still want to do interdevice synchronization. Anyway, I have two tools that completely offer solutions to all of that. Anyway, so we're going to talk about that.

But I do want to, because this is still Security Now!, take a look at Mozilla's recently announced plans to gradually and carefully bring DNS-over-HTTPS to all Firefox users in the U.S. It turns out that it's not quite the slam dunk that we might imagine. And I also want to check in with the EFF to see what they think about this, and then also remind our listeners about the 100% free VPN offering coming from our friends at Cloudflare. So do a little bit of Security Now! business, and then we're going to plow into how you achieve The Joy of Sync.

Leo: The Joy of Sync. And of course we have a Picture of the Week because you wouldn't go a week without a picture.

Steve: No, we can't.

Leo: All right, Steve. Let's, well, actually, should we do the Picture of the Week before we get syncing?

Steve: I think we should. And this is a little gruesome, actually, when you look at the second half of the picture. But it's been in my archive of photos for a while, so I thought it would be sort of fun. This is the CAPTCHA, the "I'm not a robot." And the upper half of the picture says, "Are you a robot?" And it shows a human finger touching the "I'm not a robot" checkbox to assert that fact. And then the gruesome part is the second half of the picture.

Leo: Oh.

Steve: Where we encounter that disturbingly lifelike Boston Dynamics robot holding, unfortunately, a severed human hand, and in its other hand the phone showing that the checkbox of "I'm not a robot" has been checked.

Leo: Of course, it isn't a fingerprint, so this doesn't really scan.

Steve: Doesn't quite capture the [crosstalk].

Leo: Sometime I would love to spend some time on this "I'm not a robot" reCAPTCHA because I understand how it works; but sometimes you click it, and it goes, okay, I believe you, and sometimes you have to look at road signs for hours.

Steve: Yes.

Leo: And it's an interesting thing that Google's doing with this. And so at some point it would be kind of fun to talk about that.

Steve: Yes. It uses basically a deep reputation network in order to see if it knows your browser, if it knows your IP. I mean, Google's watching us. And so...

Leo: It knows who you are.

Steve: Yeah, it's even like what have you been doing recently before you claim not to be a robot, or does this suddenly just come out of nowhere by surprise.

Leo: Which shows ironically that a robot could click it, as long as it had your phone that you'd been using up to the point where it ripped your head off.

Steve: Yes, exactly. Exactly. So I did want to talk a little bit - we've been talking recently about DNS-over-HTTPS. A month or two ago we talked about the setting that was not yet the default for Firefox, which Mozilla is beginning to experiment with. And I remember that the default provider - the idea, of course, is that, rather than sending DNS queries out over UDP to your regularly scheduled DNS provider, which is more often than not your ISP, people's home routers will use DHCP to obtain, not only an IP address, its public IP for itself, but also the IPs, typically, of your local ISP, the idea being that DNS servers that are near you are going to be the fastest to respond.

DNS servers are caching, so if anybody in your area had looked at any of the same sites that you go to, and if you look at the distribution, lots of people are going to Google, lots are going to Facebook.com and so forth, then it's very often the case that, if your computer doesn't already know the IP address for www.facebook.com, that very short distance away your ISP's DNS server will.

So anyway, of course the problem with that is that UDP and DNS are unencrypted by default, and that represents a privacy concern for people. So what Firefox has been experimenting with is bringing up an HTTPS tunnel so that all of your web browser's DNS queries, instead of going to that local ISP, will go to a DNS server at the other end of the tunnel, which at this point by default is Cloudflare because they offer, as it's called, a DoH, DNS-over-HTTPS service.

Leo: Unless you just do your own. But I don't do my own. I use Cloudflare. I mean, you're basically sending all that traffic to Cloudflare, and I think they're doing it for a reason. They want to know all that stuff; right?

Steve: They really do seem to be good guys. I mean...

Leo: I don't think they're selling it, but I think they're probably using it to make their service better, that kind of thing.

Steve: They may be. I doubt that they're monetizing.

Leo: No.

Steve: I would be very surprised if they were. So what was interesting, and because there is more technical depth to this, is I picked up on a posting, a blog posting by Selena Deckelmann, who's Mozilla's senior director of Firefox Engineering. She had some very interesting details about their experiments that I knew our listeners would find fascinating. Her blog posting was - it was from September 6th - "What's Next in Making Encrypted DNS-over-HTTPS the Default." So she said: "In 2017, Mozilla began working on the DNS-over-HTTPS protocol, and since June of 2018 we've been running experiments in Firefox to ensure the performance and user experience are great." She said: "We've also been surprised and excited by the more than 70,000 users who have already chosen on their own to explicitly enable DoH in Firefox Release edition."

And, you know, we probably contributed in some small measure to that because we talked a few months ago about where to go, what to flip, and there is a UI panel now in Firefox. You don't have to dig into that about:config thing. It's just right there. And when you turn it on, your DNS queries for your browser, not for your system as a whole, because this is just browser-centric, they're all going to be encrypted. And I think there is a client for pfSense and other routers which would allow then your entire network to have its DNS handled through DoH.

Anyway, so she said: "We are close to releasing DoH in the U.S., and we have a few updates to share." She wrote: "After many experiments, we've demonstrated that we have a reliable service whose performance is good, that we can detect and mitigate key deployment problems" - and that's interesting because there are some - "and that most of our users will benefit from the greater protections of encrypted DNS traffic. We feel confident that enabling DoH by default is the right next step. When DoH is enabled, users will be notified and given the opportunity to opt out."

So of course this is huge; right? They're talking about flipping the default to DoH, meaning that, as we know, my favorite term is, one of them, "the tyranny of the default," meaning that that's, you know, the default setting is what almost everybody ends up using. So she said: "This post includes results of our latest experiment, configuration recommendations for systems administrators and parental controls providers, and our plans for enabling DoH for some users in the U.S. Our latest DoH experiment was designed to help us determine how we could deploy DoH, honor enterprise configuration, and respect user choice about parental controls."

And of course what that means is that filtering DNS is one of the ways you can keep your family away from bad websites, where "bad" can be selectively or deliberately or to some degree controlled, you know, like what class of sites do you not want people to get to. And on the enterprise side, there are collisions with, for example, enterprise DNS might define non-public domains, which are of use to the enterprise. But if you tunnel DNS outside, then public DNS servers won't know about the enterprise private DNS.

So she said: "We had a few key learnings from the experiment," as she put it. And there were two she cited. First: "We found that OpenDNS's parental controls and Google's safe-search feature were rarely configured by Firefox users in the U.S. In total, 4.3% of users in the study used OpenDNS's parental controls or safe-search. Surprisingly, there was little overlap between users of safe-search and OpenDNS parental controls. As a result, we're reaching out to parental controls operators to find out more about why this might be happening."

Then she also said: "We found 9.2% of users triggered one of our split-horizon heuristics." So a split horizon means that you intelligently fall back to a different DNS server based on some heuristics, based on some rules of thumb. So they were recognizing that there would be a problem if everybody was tunneled without exception.

So she said: "The heuristics were triggered in two situations: when websites were accessed whose domains had non-public suffixes" - not prefixes, but suffixes; right? So like some different, some non-public TLD, Top-Level Domain, like maybe dot IBM or something, for example. And she said: "And when domain lookups returned both public and private IP addresses," so the so-called RFC 1918 IP ranges, like 192.168.

So she said: "There was also little overlap between users of our split-horizon heuristics, with only 1% of clients triggering both." She said: "Now that we have these results, we want to tell you about the approach we have settled on to address managed networks and parental controls. At the high level, our plan is to respect user choice for opt-in parental controls and disable DoH if we detect them." So that is to say, if you are using a, for example, OpenDNS where you're taking advantage of their parental controls, they will not tunnel. Firefox will not tunnel in that instance.

Leo: Ah. That's smart. That's good.

Steve: Yes, yes. And she said also: "Respect enterprise configuration and disable DoH unless explicitly enabled by enterprise configuration. and fall back to operating system defaults for DNS when split-horizon configuration or other DNS issues cause lookup failures." So, for example, if they're tunneling, and you do ask for johnsmith.mycomputer.ibm, well, that's going to fail because IBM is not a TLD. So they will detect that and go, whoops. This is a configuration where something special has been done for this user's DNS. So then they will back off and disable DNS-over-HTTPS.

So essentially what that means is that what they found was that it was a very small percentage of Firefox users who would need that, and that by enabling it by default - so they will intelligently back off if they encounter that. But by and large, the huge majority of users won't ever see a problem and will get the benefit of having their browser's DNS queries tunneled to Cloudflare or wherever.

And of course, as you noted, Leo, one of the concerns is that, very much in the same way that a VPN has the effect of concentrating all of your traffic out to a single point, of course, as you noted, whoever it is you choose as your DoH provider is seeing all of your DNS lookups. So it has that same sort of concentration effect. On the other hand, many people are more concerned about what their own local ISP is doing with, you know, having advantage of their DNS lookups. Like, for example, it's considered controversial that you enter, you mistype a domain name, and you get your, like in my case, your local Cox Cable intercept page saying, oh, well, here's some things to consider. I mean, they take it as a marketing opportunity.

Anyway, so her post continues: "We're planning to deploy DoH in 'fallback' mode; that is, if domain name lookups using DoH fail, or if our heuristics are triggered, Firefox will fall back and use the default operating system DNS. This means that for the minority of users whose DNS lookups might fail because of split-horizon configuration, Firefox will attempt to find the correct address through the operating system DNS. In addition," she said, "Firefox already detects that parental controls are enabled in the operating system; and, if they are in effect, Firefox will disable DoH.

"Similarly, Firefox will detect whether enterprise policies have been set on the device and will disable DoH in those circumstances. If an enterprise policy explicitly enables DoH, which we think would be awesome," she writes, "we will also respect that. If you're a system administrator interested in how to configure enterprise policies, please find documentation here." And she provided a link. And I have a link to her posting in the show notes, for anyone who wants to follow up. She said: "If you find any bugs, please report them here," and another link.

So this was interesting, too. She said, under "Options for Providers of Parental Controls," she said: "We're also working with providers of parental controls, including ISPs, to add" - and I thought this was very clever - "a canary domain to their block lists. This helps us in situations where the parental controls operate on the network rather than an individual computer. If Firefox determines that our canary domain is blocked, this will indicate that opt-in parental controls are in effect on the network, and Firefox will disable DoH automatically. If you are a provider of parental controls, details are available here." And she provided a link.

"Please reach out to us for more information at doh-canary-domain@mozilla.com," so by email. "We're also interested in connecting with commercial block list providers, in the U.S. and internationally. This canary domain is intended for use in cases where users have opted in to parental controls. We plan to revisit the use of this heuristic over time, and we'll be paying close attention to how the canary domain is adopted. If we find that it is being abused to disable DoH in situations where users have not explicitly opted in, we will revisit our approach."

And then she concluded, saying, under "Plans for Enabling DoH Protections by Default," she said: "We plan to gradually roll out DoH in the U.S. starting in late September. Our plan is to start slowly enabling DoH for a small percentage of users while monitoring for any issues before enabling it for a larger audience. If this goes well, we will let you know when we're ready for 100% deployment. For the moment, we encourage enterprise administrators and parental control providers to check out our config documentation and get in touch with any questions."

So anyway, I think this is great. As we noted, and as we know, most users, even Firefox users, just sort of say, okay, I know that Firefox is doing the right thing. It's not monetizing what I do. It's protecting my privacy more so than more popular browsers. So I'm just going to leave it set the way it is. You know, there are things one can do to increase that, and eventually all your DNS lookups will get tunneled. And, you know, of course you can turn that off if you'd rather have a more dispersed DNS querying, although I can't imagine trusting anyone more than Cloudflare. I mean, they're good guys.

Leo: Yeah.

Steve: And I was curious about the EFF weighing in. They recently posted: "Encrypted DNS Could Help Close the Biggest Privacy Gap on the Internet." And they said: "Why Are Some Groups Fighting Against It?" And I thought this was interesting. They wrote: "Thanks to the success of projects like Let's Encrypt" - which of course we know the EFF was big behind and has succeeded at their goal of providing free encryption for sites that weren't otherwise encrypted, with the recognition that of course it's domain-level encryption, not organization validation, and not extended validation, just domain-level encryption. But it means that traffic is no longer to a great degree traveling over the 'Net unencrypted.

And the EFF wrote also, thanks to the success of "recent UX changes in the browsers, most page loads are now encrypted with TLS." They wrote: "But DNS, the system that looks up a site's IP address when you type the site's name into your browser, remains unprotected by encryption. Because of this, anyone along the path from your network to your DNS resolver, where domain names are converted to IP addresses, can collect information about which sites you visit. This means that certain eavesdroppers can still profile your online activity by making a list of sites you visit, or a list of who visits a particular site. Malicious DNS resolvers or on-path routers can also tamper," they note, "with your DNS request, blocking you from accessing sites or even routing you to fake versions of the sites you requested.

"A team of engineers is working to fix these problems with DNS-over-HTTPS, a draft technology under development through the IETF that has been championed by Mozilla. DNS-over-HTTPS prevents on-path eavesdropping, spoofing, and blocking by encrypting your DNS requests with TLS. Alongside technologies like TLS 1.3 and encrypted SNI" - that's Server Name Indication, which is one of the last things that needed to get fixed, which TLS 1.3 does, because even on a secure connection, and even if DNS were encrypted, the browser now indicates what site it wants to connect to so that a multihosted server is able to select the proper security certificate for the connection. So that was one remaining privacy leakage which is resolved in TLS 1.3.

So they say: "DoH has the potential to provide tremendous privacy protections. But many Internet service providers and participants in the standardization process have expressed strong concerns about the development of the protocol." And of course we covered this, and they remind us: "The U.K. Internet Service Providers Association even went so far as to call Mozilla an 'Internet Villain'..."

Leo: Can you believe that?

Steve: "...for its role in developing DoH." Of course they later backtracked on that, saying, whoops, we didn't realize that was going to be such a mistake. So the EFF says: "ISPs are concerned that DoH will complicate the use of captive portals, which are used to intercept connections briefly to force users to log into a network, and will make it more difficult to block content at the resolver level. DNS-over-HTTPS may undermine plans in the U.K. to block access to online pornography." They said: "(The block, introduced as part of the Digital Economy Act of 2017, was planned to be implemented through DNS.)"

They say: "Members of civil society have also expressed concerns over plans for browsers to automatically use specific DNS resolvers, overriding the resolver configured by the operating system, which today is most often the one suggested by the ISP. This would contribute to the centralization of Internet infrastructure, as thousands of DNS resolvers used for web requests would be replaced by a small handful." True. "That centralization would increase the power of the DNS resolver operators chosen by the browser vendors." Also true.

Leo: That's a good point, yeah.

Steve: Yeah, "which would make it possible for those resolver operators to censor or monitor browser users' online activity. This capability prompted Mozilla to push for strong policies that forbid this kind of censorship and monitoring." Well, and think about it. I mean, if Mozilla chooses you, as they chose Cloudflare, then in return Mozilla can say, hey, Cloudflare, we'd like to promote you, and implicitly your service, but you've got to promise in return no funny business. And of course Mozilla has a lot of clout because who they aim all of their browsers at is entirely their choice, with the exception of users who manually override. But as we know, that's going to be vanishingly small percentage.

So the EFF says: "The merits of trusting different entities for this purpose are complicated, and different users might have reasons to make different choices. But to avoid having this technology deployment produce such a powerful centralizing effect, EFF is calling for widespread deployment of DNS-over-HTTPS support by Internet service providers themselves." In other words, let's bring a bunch of services, you know, like OpenDNS and Google and others, so that users will have more choice. And you can imagine an easy-to-use list where you choose, or maybe even rotate them. Doesn't always have to be the same one; right? So there's no reason you couldn't bring up a couple tunnels and just hand out queries in a round-robin fashion to a spread, so you sort of have a spread spectrum DoH in that case.

Leo: Ooh, that's clever. What a good idea, yeah.

Steve: So they say: "This will allow the security and privacy benefits of the technology to be realized while giving users the option to continue to use the huge variety of ISP-provided resolvers that they typically use now." And they said: "Several privacy-friendly ISPs have already answered the call." They said: "We spoke with Marek Isalski, Chief Technology Officer at U.K.-based ISP Faelix [F-A-E-L-I-X] to discuss their plans around encrypted DNS. Faelix has implemented support for DNS-over-HTTPS on their pdns.faelix.net resolver. They weren't motivated by concerns about government surveillance, Marek says, but by 'the monetization of our personal data.'" Which is to say, they want to work against it.

They wrote: "To Marek, supporting privacy-protecting technologies is a moral imperative." He was quoted by the EFF: "I feel it is our calling as a privacy- and tech-literate people to help others understand the rights that GDPR has brought to Europeans," he said, "and to give people the tools they can use to take control of their privacy." So the EFF concludes: "EFF is very excited about the privacy protections that DoH will bring, especially since many Internet standards and infrastructure developers have pointed to unencrypted DNS queries as an excuse to delay turning on encryption elsewhere in the Internet.

"But as with any fundamental shift in the infrastructure of the Internet, DoH must be deployed in a way that respects the rights of the users. Browsers must be transparent about who will gain access to DNS request data and give users an opportunity to choose their own resolver. ISPs and other operators of public resolvers should implement support for encrypted DNS to help preserve a decentralized ecosystem in which users have more choices of whom they rely on for various services. They should also commit to data protections like the ones Mozilla has outlined in their Trusted Recursive Resolver policy. With these steps, DNS-over-HTTPS has the potential to close one of the largest privacy gaps on the Internet."

So that's all for the best. Now, of course, this is all happy news. But we do need to remember that, even though our communications are encrypted, and now our DNS lookups would also be encrypted and hidden, the IP address to which we're communicating remains in the clear and fully visible. So none of this does anything about that. If you're on the wire, you're still seeing connections to IPs after they've been resolved by DNS. And it's true you can't see into them. But there is that technically metadata of who you are connecting to that continues to stand out.

So I did want to remind our listeners, we talked about it at the time, but Cloudflare also has a project called WARP which, at this point, they've sort of followed their 1.1.1.1 DNS offering with a WARP client for mobile. At this point it's iOS and Android only. There was a signup countdown where you could - I don't think it's widely deployed yet. I haven't checked back to see what the status is. But it was a 100% free for mobile VPN, which then would basically solve all of this and solve the IP monitoring problem, with the caveat that, yes, all of your traffic is then being centralized.

I mean, this was really the point that the EFF was making, was the power of the Internet is its decentralization. So that's one of the downsides of any time you're creating a tunnel, any tunnel, whether it's a DNS tunnel or a VPN tunnel, it's going to be inherently providing some centralization.

Leo: I don't know if you saw that Firefox is now offering a private network. It's in beta right now.

Steve: No.

Leo: Yeah. I'm using it right now. If you have a Firefox account, you install an extension and turn it on. And I was just looking, and it says I'm coming from 8.45.41.210, which is a Level 3 domain, but I don't know anything else about it. It's not mine, obviously.

Steve: Yeah.

Leo: So it's working as a VPN. Of course, it only works in the browser. But it seems to be pretty transparent and painless, and it doesn't seem to be slowing things down much. So that's kind of cool, too.

Steve: Very nice.

Leo: I think we're making real progress. And a lot of credit to Firefox, I have to say, and Cloudflare, for pushing this stuff forward.

Steve: Yeah. Yeah. Well, and I think we're rapidly approaching a point now with the way the Internet's CDN and multihoming structures have evolved, I mean, where if you use 1.1.1.1, you're actually going to a local resolver because all...

Leo: Right. Oh, that's interesting. You're not going to Cloudflare.

Steve: Yeah, yeah, yeah. Well, you're going to their IP, but the way the resolution happens is you're being - it's all fancy top-level Internet stuff. But they've got resolvers spread all over the world that answer that IP local to you through the whole...

Leo: Oh, that's why it's fast. That's great.

Steve: Yes, exactly.

Leo: Have you run it through your DNS Benchmark? Have you checked its speed?

Steve: Yeah, it does pretty well. It holds its own.

Leo: Can we take a little break, Steve, before we get to Closing the Loop?

Steve: Yup, perfect.

Leo: Back to you, Steve. We are back.

Steve: So I have two tweets, which sort of also double as our closing-the-loop feedback that takes us into our topic.

Leo: Oh, good.

Steve: Ashton, he said: "Hi, Steve. On the latest SN you mentioned Boxcryptor Classic for transparent encryption of synchronized files." He says: "I was using it for years, ever since you mentioned it on a previous SN episode, with great success; but the fact that they discontinued development on Classic never sat well with me. A couple of years ago I found and moved to Cryptomator, which is open source donationware that does the same thing. It's cross-platform and has Android/iOS apps, though those aren't free. Thought you might want to check it out." And that's Cryptomator, C-R-Y-P-T-O-M-A-T-O-R, dot org.

And so responding to Ashton, I agreed completely. I said: "First of all, Boxcryptor can still be used for free for TNO [Trust No One] cloud storage with the limitations that it will only sync two devices and only with a single cloud provider. My big complaint with Boxcryptor is that they have switched to a software rental plan where it's no longer possible to purchase, for any price, a version that is not artificially constrained. Anything beyond two devices and a single cloud provider requires a subscription. And that's where I say no."

So Cryptomator is really the logical inheritor of Boxcryptor. It runs on everything, it's open source, and encrypts all file metadata so that the cloud sees nothing. It doesn't itself deal with synchronization, but relies on the cloud provider's folder and file sync.

Essentially you mount an encrypted volume which creates a virtual drive in your drive hierarchy. And then files that are there, you know, you basically work with those.

So while Cryptomator is not one of the two tools or services that I have chosen, it may be perfect if it fits some users' needs. So, for example, if you have an existing cloud provider that you want to continue to use, they don't offer Trust No One encryption, but they do do file syncing, then Cryptomator, I mean, it's a very nice piece of work. I watched the early release and introduction videos by the authors, European authors. It looks, I mean, they did a very nice job, and it is cross-platform, supports everything. So maybe something to consider.

Paul Comfort tweeted from @PCComf. He said: "Hi, Steve. I listened to your recent sync episode and felt your pain." He said: "I've done a lot of sync testing, and I have used or tested most major platforms, both with my own storage and using cloud storage. I wanted to make you aware of a TNO solution I've found that meet my various needs best, and I think might also do the same for you: Sync.com." He says: "I recommend the Business Solo plan, which is similar in pricing to Dropbox." And actually it's better than Dropbox.

He says: "Any TNO solution is going to have inherent limitations to what they can do simply because they can't see your data. It's going to make them less feature-full than Dropbox or OneDrive, for example, but Sync.com makes the most of what it can do and provides enough features for me. They still allow web access through an in-browser local decryption, similar to what LastPass does."

He says: "I currently sync over 100,000 files with them across several systems, both macOS and Windows." He says: "During my testing, that number of files gave both Syncthing and SpiderOak trouble. I make use of their archive function/location for things I never want synced. I make use of their selective sync for systems that don't need my entire file structure. The one Dropbox/iCloud/OneDrive-like feature I wish it had would be on-demand sync so that my smaller systems with smaller drives can still display the entire tree, but only download items when I need them. But selective sync makes up for that."

He says: "I just need to be smarter about what I sync. I also have a system connected to a Drobo that synchronizes everything to the Drobo so I can take a full personal backup of all the online data in case something happens to them." He says: "I'm not affiliated with them at all except as a customer, even though this message probably sounds like an advertisement." Well, believe me, compared to what I'm about to do with this podcast. He says: "I'm a satisfied user, and I think it would meet your requirements better than any other solution you mentioned."

Leo: I've seen a lot of that from Sync.com. You know, the price is 10 bucks a month for three terabytes.

Steve: Yup, three. Three terabytes.

Leo: So I pay $99 a year, that's compared to $120 a year for two terabytes on Google Drive. So that, I think, is a very good price, if you have that much storage.

Steve: I tweeted back to Paul, and I let him know that, yes, indeed, Sync.com, for many of the reasons he noted, was one of the two tools I had settled upon after looking at everything.

Leo: Oh, good. My negative on it, no Linux. Now, don't they say they're going to do Linux?

Steve: Yes, yes. That is the - you hit it. That is the one downside that will affect some of our listeners. They do, however, have it on their road map.

Leo: Yeah.

Steve: So they are intending to do Linux. And I ended up with a kind of an interesting workaround using a Raspberry Pi that I will explain.

Leo: Oh, clever.

Steve: And I also should note that I have it running on my Drobos, and that there's a Synology client for it, also.

Leo: Oh, that's good. I use Synology. I mean, I basically do this with a Synology.

Steve: Right.

Leo: So I have my own cloud.

Steve: So I looked at ownCloud, which is what I was talking about when you and I first breached this topic a couple months ago. I looked at Nextcloud. I saw what you were talking about, about the...

Leo: The schism.

Steve: Basically, yes, the originator of ownCloud was annoyed that they were beginning to also offer some paid-only, like, super cloud sorts of things. And he said, yeah, I really don't want to do that. So he forked the source. And, I mean, looking at them side by side, there's no difference between them. I mean, they're, like, they are identical in every meaningful way. So I don't know what the ownCloud people have been doing, but they sure haven't changed the UI at all that I was able to see. Of course, as I mentioned, I looked at Cryptomator. There's something called an encrypted file system, Cryfs.org. There's also Syncovery. There's Duplicati. There's Duplicit - I don't want to say "duplicity."

Leo: Duplicacy. Duplicacy.

Steve: Duplicacy, thank you, duplicacy. There's Tresorit. There's pCloud. There's something, AJC Sync. We talked, in between my first mention and now, about that clever idea of using VeraCrypt, that is, mounting a VeraCrypt expandable volume, for example, in Dropbox because Dropbox is clever about only sending changed blocks. And so it would make sense to mount a VeraCrypt volume because we know VeraCrypt's crypto has been looked at and scrutinized. and it's been audited and all that.

And so the idea would be, when you're working in a file system and changing files, you know, a file system is only changing clusters. And so if the blocks were like clusters, it would make sense. I mean, you could see how that would be efficient to do in a mode where only the things that are being changed in a big volume would be resynced. It would never work in an environment where changing a file, changing part of a file meant the whole file was resent because then, if you touched anything in the VeraCrypt volume, it would send your entire volume back up to the cloud provider. So that would not work.

But I ended up being very bullish on Sync.com. As you noted, Leo, you get a lot more storage for the money. Dropbox gives you, for our listeners who want to play with this, Dropbox gives you 2GB for free. Sync gives you 5GB for free. So even just paying nothing, setting up a free plan, you get 5GB. And in fact there is an affiliate link which you can use to add an additional 20 if you've got friends. I tweeted my affiliate link when I was discovering them.

I found out that it is capped at an additional 20GB. I thought, you know, with my Twitter followers - oh, and people who use the link also get a free GB for using the link. So they start out with six rather than five for free. I ended up with 25 because I got the five plus the 20GB from people following my affiliate link. And I told everybody, I said, well, this is an affiliate link. We each get a free gigabyte. So let's do that. And then that's how I found that it was capped.

But they have, if someone wants to pay less, they have a $4 per month plan they call the Personal Plan, which is 500GB, so half a terabyte, so that would be, what, $48 a year. But really the bargain, I think, is what they call - and this is what - was it Steve who tweeted? No, Paul, who tweeted about the so-called Solo Business. They have 2TB for $8 a month, whereas Dropbox charges $10 a month for 2TB. But for that same $10 a month from Sync you can get 3TB. And that's, 3TB, that's a lot of...

Leo: Even two for most people is probably more than you need.

Steve: Yes. Well, yes, because...

Leo: One is probably more than you need because you have to upload it.

Steve: Yes. And what I'm using it for is cross-system synchronization of just subsets of my data. Okay. So let me talk a little bit more about what Sync is and to introduce it to our listeners. So basically you can think of it as Dropbox without all the other bells and whistles. So it's just cloud storage. But they did solve the TNO problem perfectly. So there are clients for Windows and Mac, notably not yet for Linux. So that still needs to get resolved. And you and I, Leo, were also talking about the fact that, where Dropbox creates its own Dropbox folder, Sync does the same thing. And so that chafed a little bit because, for example, I wanted my ASM tree, I have an ASM, for assembly language, on the root of my C drive. And I wanted that to be synchronized, rather than stuff in the Sync folder.

I found a beautiful solution that involves junction points on NTF volumes, also known as hard links, the idea being that you move the folder from your normal folder hierarchy under the Sync directory. And so like in my case I have moved my ASM folder. I just dragged it under the Sync directory, and that was instant. And then Sync, of course, went about doing its synchronization to the cloud.

And remember, I mean, these are my crown jewels. This is all of the - this is like the SpinRite source and everything is now synced with Sync in the cloud. Then Windows has a command, "mklink," and you use the /J for creating a junction point. And basically you create a junction point at the \ASM, pointing it to the ASM folder under the Sync directory, and it just reappears. For me, my ASM folder is back under C: where I like it, where all of my tools expect it to be, where all of my batch files and scripts and everything expects it to be. Under a command window I can go to C:\ASM, so in no way is it different, as if it had never been moved.

Leo: Is it like a hard link in Linux, or like a symbolic link?

Steve: Yes.

Leo: It's like a hard link.

Steve: Yes. The symbolic link is not what you want because those are kind of flaky, and it's sort of a pointer. It is like Linux's hard link, and it is in fact a hard, well, a hard link in Windows cannot be a directory. It's only a file. So this is like a directory-level hard link.

Leo: Yeah, you can do symbolic links to folders in Linux.

Steve: Folders, yes, yeah. And so this solves that problem for me in Windows without any duplication. So it's not like I have two ASMs now. It's just that I moved my ASM folder under the Sync directory and then pointed to it from its original location. And it works, I mean, absolutely the same. I've been using it now for a month or so, and I've never seen it not do the right thing.

Leo: Interesting.

Steve: So let's see. I'm just going to scroll through here. Protect your ideas. I've looked at this stuff. I've looked at their technology. It's bulletproof. So, for example, they get it that their competitive advantage is not only price, but privacy.

So they note that Google Drive, the Google terms of service gives their automated systems permission to access the data stored on their servers for the purpose of monetization through advertising. The Dropbox terms of service gives Dropbox employees and trusted "third parties," whoever they may be, permission to access, view, and share the files stored on their servers at any time. The Box terms of service gives Box permission to view the files stored on their servers to ensure users are in compliance with the Box terms of service. And Microsoft OneDrive terms of service gives Microsoft employees permission to view the files stored on their servers to ensure users are in compliance with Microsoft terms of service.

So obviously I'm not putting my crown jewels with any of those guys without providing encryption. And again, Cryptomator would be what you would use if you needed to use one or more of those services. For me, Sync has it nailed. So under features, it's 100%

private cloud. "End-to-end encryption," they wrote, "protects your confidential data in the cloud from unauthorized access at all times. We can't read your files; and no one else can, either. Sync doesn't collect, sell, or share your personal data or app usage information to advertisers or third parties, and we do not claim ownership of your data."

They said, under global data privacy compliance: "Sync is safe to use, no matter where your business operates, with USA, EU, U.K., GDPR, and Canadian compliance built in." Oh, they are a Canadian company, by the way. And they said: "Including Canadian data residency. Backs up your files in real-time and makes it easy to recover deleted files and previous versions of any file, any time. Never lose a file again." That's another one of the things that is in my criteria. We've talked a lot about the danger of ransomware. So the way you resolve that is you use a system that does file versioning. And this does that.

And I feel almost guilty because I'm a person who's hitting Ctrl-S all the time, like to save, to take snapshots and save things I'm working on. And when I look in my sync history, it just like scrolls off forever. It's like, oh, my goodness, I hope they're doing, I mean, maybe they're doing something smart about eliminating redundancy. But the problem is we're doing client-side encryption, so they're not able to see, they're not able to decompose this the way, for example, Dropbox is in order to do more intelligent synchronization. But I do have versioning on all of these files. They said: "Data is replicated across multiple SSAE 16 Type II certified datacenter locations with SAS RAID storage, automatic failover and a 99.9% or better uptime SLA," you know, Service Level Agreement.

"For account security, two-factor authentication, granular user permissions, remote wipe, custom passwords, expiration dates, notification and more ensure you're always in control." And Leo, the thing that I saw that immediately made me think, okay, I'm with the right guys now, is when I was setting up my account and created username and password and sent that in, I got a dialog prompt that said "Use email for password recovery," and it was not checked by default.

Leo: Good.

Steve: And I thought, whoa. When have you ever seen that? I mean, so there, like, by default you're not using email for password recovery. Because of course, as we know, that's the weakest link in password handling. So in fact, in my SQRL presentation I joke with people that I've run across people who don't even bother to write passwords down. They say, "No, I just gave up on that a couple years ago. I just click 'I forgot my password.'"

Leo: Every single time?

Steve: Every single time.

Leo: And then of course it gets emailed. But it's a reset, usually, which is - that's not totally insecure, right, a reset link? Of course anybody could get that.

Steve: Yeah, it's totally, yeah.

Leo: Yeah, because then they've got it, yeah.

Steve: Exactly. So, I mean, it just makes you cringe. But, you know, okay. I mean, so here's, you know, that's what we've done. The point I make during my SQRL presentation is we have so - "we," IT, Internet, techie world have so abused the common man, you know, the typical user, that they're just, like, okay.

Leo: I give up. I give up.

Steve: I give up. I'm just going to say, no, it's true, I don't remember because I never wrote it down. I just click the "I forgot," and I get it. And look, oh, I'm logged in.

So they say, under their technical overview, it's zero-knowledge, end-to-end encryption. File and file metadata is encrypted client-side and remains encrypted in transit and at rest. The web panel, file sharing, and share collaboration features are also zero-knowledge. Private encryption keys are only accessible by the user, never by Sync. Passwords are never transmitted or stored and are only ever known by the user.

So we get some juicy crypto details: "A randomly generated 2048-bit RSA private encryption key serves as the basis for all encryption at Sync. During account creation, that unique private key is generated on the user's machine and encrypted with 256-bit AES GCM, which coincidentally is what I use to encrypt SQRL's identities in the SQRL system, so same bulletproof encryption. And that is locked with a user's password, which also is what I do. This takes place client-side within the web browser or app. PBKDF2 key stretching with a high iteration count is used to help make weak passwords more cryptographically secure.

"Encrypted private keys are stored on Sync's servers and downloaded and decrypted locally by the desktop app, web panel, or mobile apps after successful authentication. At no time does Sync have access to a user's private key. A username and password is required to authenticate and log into the Sync desktop app, the Sync web panel, and Sync's mobile apps." Oh, and there's another cool thing that happens, Leo, when you have the Sync app running on your desktop. It has a minimal UI, so it will launch your web browser. It then authenticates with your desktop app in order to get itself authenticated. So you don't have to relog into the web browser. It automatically does a single sign-on behind the scenes with the desktop app.

They said: "During authentication, a bcrypt hash of the user inputted password is generated locally" - bcrypt being a hard-to-accelerate solution - "locally on the computer or device, using a unique salt that is stored on the server. Bcrypt," they remind us, "is a one-way hashing mechanism, meaning the hash cannot be unhashed or deciphered. The benefit of bcrypt is that it is slow by design, which prevents brute force or rainbow table attacks. At no time" - and remember, every single user gets a unique salt which the server holds so that you're able to operate across platforms. "The server authenticates against the hash, but at no time is the hash itself stored. Successful authentication allows the app to download the user's encrypted private key, which is then decrypted locally with the user's actual password." In other words, they have done everything right from a crypto standpoint.

They say: "Sync's web panel runs on the client, within the web browser, as a self-contained AngularJS app. The web panel utilizes Stanford JavaScript Crypto Library" - which, by the way, is the right one - "a JavaScript implementation of ISAAC, which is a fast cryptographic random number generator" - so they're not trusting the one built into JavaScript - "and HTML5 local storage. This means only modern web browsers are supported. In other words, IE10 is the minimum requirement." Which of course is no problem.

"The web panel authenticates the user, downloads the encrypted encryption keys, decrypts the keys locally within the web browser, then downloads and decrypts file and file metadata as required." And so short version is your web browser is a very nice UI, presents a nice UI that gives you access to everything that is there as if it wasn't encrypted, but it is. "Passwords are never transmitted or stored during this process. The web panel is open source. Source code is easily viewable from any web browser for those technically inclined to see how it works.

"File data is always encrypted, in transit and at rest. Sync utilizes a unique 256-bit AES GCM data key on each file, locked with the user's 2048-bit RSA key. File metadata is encrypted separately with the user's password, PBKDF2 key stretched, to generate a key for the 256-bit AES GCM. This all happens locally on the user's computer or device before the files are transferred to Sync's servers. This ensures that the encrypted file data stored on Sync's servers is impossible to access, even in the event that the servers themselves are compromised."

And they've solved the problem of users. One of the requests I got from our listeners is, yeah, I'd like to store things encrypted in the cloud, but I also want to be able to share specific files with people. Like give somebody a link that they're able to use. They explain: "Secure links are locked with a unique password appended to the link after the hashtag." That's very clever because what's after the hashtag, the pound sign, is never transmitted. "The data appended to a URL after the hashtag is called a 'fragment identifier,' which is never transmitted or stored, meaning the password is never known to Sync. The password itself is key stretched using PBKDF2 with a high iteration count, to generate the encryption key used to unlock the link."

So that's a link that an individual can generate and share with a friend, and then tell them what the password is. When they use the link, they'll be prompted for the password. Put that in, and then that file will be, after the information provided is verified, be downloaded and decrypted locally for that user to gain access to. So again, they've solved the link sharing, the specific file sharing problem, and maintained TNO encryption.

Then they also provide a shared folders facility. Shared folders utilize key wrapping, which is multikeying, basically, to allow files to be shared between different Sync users privately, without the need to reencrypt the actual file data when users are added and removed from the share. Sync never has access to file data, even when being shared. When a new share folder is created, the unique encryption key on each file within the share is encrypted with a unique 512-bit share key that is created specifically for the share. The share key is then encrypted with the RSA 2048-bit public key of each user and locked with the user's RSA private key, which is encrypted with the user's password.

So in other words, they do, again, exactly the right thing. They meta encrypt the share under a 512-bit key, which is then multiply encrypted under all of the people who are sharing that folder, multiply encrypted under all of their public keys, which means that any one of them are able to decrypt that 512-bit key with their private key and so, again, create basically a network of truly encrypted shared folders where you get to use Sync to, like, create the cloud-based infrastructure for sharing folders among multiple Sync users.

And the whole process is transparent. Basically, you end up with a folder on your system that you are securely sharing among other Sync users. I mean, they've just nailed this. Single sign-on, and here they mention what I was saying before. The Sync desktop app allows the user to log into the web panel automatically. The web panel is loaded locally using a unique URL that contains a memcache lookup value. And here they're getting into the details of how the web panel works. I'll skip all that.

And they note that Sync uses SSL and TLS for all data transfers, but does not rely on SSL or TLS for any meaningful security. They say SSL on its own cannot be trusted. SSL is applied as an extra layer on top of the 2048-bit RSA and 256-bit AES encryption used to encrypt each file. This ensures that in the event that SSL is compromised, for example, a man-in-the-middle attack or an SSL software vulnerability, and of course we're talking about certificate authorities being compromised, or middleboxes that are performing on-the-fly decryption of SSL, none of that affects the security of this system. It remains encrypted inside that tunnel and thus impossible to access.

So anyway, I mentioned how I had solved the problem of things needing to be in the Sync folder by using the junction point or directory junction, actually, /J, in the mklink command for Windows. So I am, actually, I am super happy with it. For me, it's the right cost, the right technology. These guys nailed it. Somehow I like the idea of the things I absolutely really care about never losing access to. And also, of course, I have access now on the road. So all of these things are available while I'm traveling just by tying into a publicly available cloud provider without sacrificing any security. So that is one of the two things I have found. And Leo, let's take our last break, and then I will tell everybody about the second really cool tool.

Leo: Thing 2. Boy, if this only had a Linux solution.

Steve: Yeah.

Leo: I wonder how close they are to solving that because that really would - see, right now I'm doing it all to a Synology. But with 3TB, I mean, because I use a Synology basically as a backup tool, but then I also have a Documents folder which is the stuff I want on every machine. It's about 20GB. It's not that big. I use a git for my source code, which I don't - I'm always amazed that you don't use version control of any kind in your source code. But to each his own. And I use an encrypted git, which I like a lot. But I could also, you know, I could do this, too. And I think probably I could get even like my music and photos back up. I think those are each about a terabyte. I can get it all into 3TB. And it would have the advantage of it would be offsite. My Synology's in the house.

Steve: That is exactly it, is that just having it somewhere else. I mean, I have servers at Level 3, so I could run...

Leo: Well, that's the thing. You could do this.

Steve: Yeah. Although I'm just - I'm so security conscious that having a server, having my server at Level 3 that has a public port, I just, you know, it just makes me uncomfortable that there isn't going to be some buffer overrun or buffer overflow. And so here it's somewhere else.

Leo: It is public. I mean, that it's accessible by the Internet.

Steve: Yeah, well, so, yes, but it's not my server that would be exposed, it's their server that would be exposed.

Leo: And your data on there is encrypted.

Steve: And my data in there cannot be accessed, exactly.

Leo: I think you may have convinced me that this would be a - and as you said, there's a Synology app, so I could easily use my Synology, too. That's interesting. I just wish there was a Linux. All right, Steve. Thank you. Back to the show. Steve Gibson, Leo Laporte. The Joy of Sync continues.

Steve: Now, I just gave everybody my absolute first choice of a service that did everything right. There is another incredibly cool tool which, Leo, you will be glad to know does support Linux, and has iOS and Android apps, and is also just, I mean, it is addictive seductive, it is done so well. A number of people recommended it. I looked at it. And, I mean, just in terms of their philosophy and what they've done, it's so cool. And that's called...

Leo: Now you've got me excited.

Steve: It's called Syncthing.

Leo: Oh, yeah, okay.

Steve: Yup.

Leo: Yup.

Steve: Yes. So Syncthing is a peer-to-peer solution. And it is what I have running on both Drobos and also on my workstations. What I'm now thinking I'm going to do is I'm going to use Sync.com as my glue between my locations, and Syncthing as my LAN-level sync, although it is not constrained to that. So, okay. So what is it? It is an open source, open protocol - you can think of it, if you wanted to think of it like something like BitTorrent, which is now Resilio...

Leo: Resilio, yeah.

Steve: Yes.

Leo: And not free.

Steve: And not free, and not open.

Leo: And not open. I love that the Syncthing is, which is really great.

Steve: Yes. And open protocol, open source, and their heart is in the right place. So the project's formally stated goals are Syncthing should be, one, safe from data loss. They said: "Protecting the user's data is paramount. We take every reasonable precaution to avoid corrupting the user's files. This is the overriding goal, without which synchronizing files becomes pointless. This means that we do not make unsafe tradeoffs for the sake of performance or, in some cases, even usability." And I should note that it's written in Go, so it outperforms other solutions that are written in PHP, like ownCloud, or what's the other one? OwnCloud or...

Leo: Nextcloud.

Steve: Nextcloud, yes. Secure against attackers, they say. "Again, protecting the user's data is paramount. Regardless of our other goals, we must never allow the user's data to be susceptible to eavesdropping or modification by unauthorized parties. This should be understood in context. It is not necessarily reasonable to expect Syncthing to be resistant against well-equipped state-level attackers. We will, however, do our best. Note also that this is different from anonymity, which is not currently a goal." And I'll explain what that's about.

Ease of use: "Syncthing should be approachable, understandable and inclusive. Complex concepts and math form the basis of Syncthing's functionality. This should nonetheless be abstracted or hidden to a degree where Syncthing is usable by the general public." Certainly by our listeners. There's no question about that.

Leo: I admit, I have installed Syncthing and am having a little trouble figuring out how to get it syncing.

Steve: I did, too.

Leo: Okay, good.

Steve: I didn't initially understand the model.

Leo: Yeah, yeah.

Steve: I've nailed it now.

Leo: And maybe because I was thinking of it as a BitTorrent sync client, kind of.

Steve: Yes.

Leo: And that made it hard to understand, yeah.

Steve: Yes, because one of the things that's noted is that BitTorrent is sort of more automagic than this is, whereas - but here you get control. And now that I understand what it's doing, I really like that control. So our listeners should know there's a little bit of a learning curve. But just take it easy, and you'll get there.

So then they said it should also be automatic. "User interaction should be required only when absolutely necessary. Specifically, this means that changes to files are picked up without prompting, conflicts are resolved without prompting, and connections are maintained without prompting. We only prompt the user when it is required to fulfill one of the overriding secure, safe or easy-to-use goals."

Also, universally available. "Syncthing," they say, "should run on every common computer. We are mindful that the latest technology is not always available to any given individual. Computers include desktops, laptops, servers, virtual machines, general purpose computers such as Raspberry Pis and, where possible, tablets and phones. NAS appliances, toasters, cars, firearms, thermostats, and so on may include computing capabilities, but it is not our goal for Syncthing to run smoothly on these devices." So maybe not your toaster or your shotgun, but things where it's reasonable for it to run.

And they said: "For individuals, Syncthing is primarily about empowering the individual user with safe, secure, and easy-to-use file synchronization. We acknowledge that it's also useful in an enterprise setting and include functionality to support that. If this is in conflict with the requirements of the individual, those will, however, take priority."

And then they said: "Everything else. There are many things we care about that don't make it onto the list. It's fine to optimize for these values, as well, as long as they're not in conflict with the stated goals above. For example, performance is a thing we care about. We just don't care more about it than safety, security, et cetera. Maintainability of the code base and providing entertainment value for the maintainers are also things that matter. It is understood that there are aspects of Syncthing that are suboptimal or even in opposition with the goals above. However, we continually strive to align Syncthing more and more with those goals."

So I am 100% sold on this thing, this Syncthing, now that I understand it. My original setup, I was really interested to know whether a Syncthing instance at one of my locations would be able to connect to my Syncthing at the other because in my place with Lorrie, I'm not behind one NAT, I'm behind two NATs because I ended up putting a pfSense router in front of my ASUS WiFi router, so they're both NATing. So there's double NAT there. And of course here I'm in the Fortress of Solitude. I mean, I am really locked down. They instantly found each other, and I had no relay required.

So to explain a little bit about this architecture, it is peer to peer. There are a network of global discovery servers and relay servers out on the public Internet so that an instance of Syncthing is able to use the discovery servers to get the IP and port that another instance of Syncthing has a public appearance on. And so they solve, using STUN and TURN that we talked about a long time ago, NAT piercing. My Syncthing clients, each running behind all of that NATing, are able to connect directly to each other across the public Internet with no relays. So nobody sees my traffic except anybody in a direct connect. And actually since both locations are on Cox.net, it doesn't even leave Cox Cable's infrastructure.

I set up a five-node Syncthing peer-to-peer network. Both Drobos had Syncthing running. They found each other. So they created a synchronized folder hierarchy there. Then I was running on - my workstation at the other end found the local Drobo. Syncthing is able to use the public discovery servers if it needs to find a Syncthing instance elsewhere on the public Internet. But on the LAN it uses broadcast or multicast because it supports IPv6. So it'll use a LAN broadcast or an IPv6 multicast in order to find other instances of Syncthing on your own LAN. So again, what I will end up doing is using Syncthing on my LAN, on each of the LANs. But because I like Sync.com so much, I will use it to perform my interlocation straddle, and also because Sync.com gives me roaming access to everything that I am syncing in my locations with full TNO encryption.

Syncthing does not encrypt, the argument being, I mean, its connections are encrypting, but it's not itself a storage provider. It is a synchronizing interconnector. It works on a directory basis. So it works in a headless fashion. There is a, for Windows users, what I would recommend is something called SyncTrayzor, as in the Windows tray. So it's called SyncTrayzor. If you just run SyncTrayzor under Windows, it includes Sync with it. The version of Sync is out of date, but the first thing Syncthing does is check for any updates, and immediately updates itself. The version, for example, that I found that I was able to install on my Drobo was older.

Again, first thing it did was perform and in-place upgrade to the latest and greatest. So using a browser UI, you decide which folders on your local machine you want to sync and with which other instances of Syncthing you want to sync them. There are two, like, weird-looking crypto tokens. One is really long, and that's the Syncthing instance identifier. When you install Syncthing, it synthesizes on the fly a TLS certificate for itself. So each Syncthing instance has a private and public key in the form of a formal TLS certificate. That identifier is a hash, an SHA-256 hash of the certificate, which is Base32 encoded with some check characters added to create that identifier.

So the beauty of that is each instance has a unique identifier which is the hash of its certificate. And this is where you were talking about how it's not quite so automatic, is it is necessary to provide that thing to the other side in order for instances of Syncthing to find each other. On the LAN, that's not necessary. What I find - it's a little jarring, actually - is that, like I'll bring up an instance of Syncthing, and it'll say, hey, who do you want to share this with? How about this guy? It was like, oh, yeah, that's the one I wanted to share it with.

Leo: Oh, that's nice.

Steve: So on the LAN it's...

Leo: That's convenient.

Steve: Yeah. On the LAN it's able to do discovery, and you're not having to, like, it's a really long thing. I mean, it's not unwieldy. You could read it over the phone. But I just stuck it on a thumb drive when I went to my other location and put it in. And once, for example, once remotely located instances have connected, then they all share all of the Syncthings that they know about with each other. So, I mean, it really does form a Syncthing hub or cloud or network. Syncthing does updates on a block level. They use 128K blocks by default.

Remember that the tweet that I read before about someone who had found that Syncthing he was having trouble with, they have addressed that since then by now supporting much larger instances of things that are being synced. So initially when you set up, there's like a startup process because it hashes all the files that you're wanting to sync. And then it's able to track changes. And I have found that, I mean, it immediately spots a change that I've made. I think there's a deliberate 10-second delay. But after 10 seconds, it finds it, and it clones it.

And because I was curious in my early experimenting phases, and because this isn't about cloud storage, it is point-to-point device synchronization, I wanted to see whether

I could create a cloud presence of my Syncthing synchronization. And so I got a $35 Raspberry Pi, and I used "rclone" and "mount" to mount my Dropbox account. This is before I was playing with Sync.com. So I mounted my Dropbox account on the Raspberry Pi running Linux and then ran Syncthing. Because Syncthing is able to sync a mounted drive, I used Syncthing to sync with my own LAN directory syncs, and it worked perfectly.

It isn't something I would recommend on a Raspberry Pi because in order - and this is an rclone problem. In order for rclone to work reliably, it has to make a full copy of the file that it is going to be syncing to remotely, or mounted to. And so that's going to, you know, Raspberry Pi runs typically on a little SD card. And those things do not have the resilience to handle me hitting Ctrl-S every 30 seconds while I'm saving copies of things. It would chew up the little SD card and cause it to die. But what you could do, if you really wanted to do this, would be just to get, you know, any older PC and just set it up with Linux and then run Syncthing on it, and then also mount a Dropbox or even a Sync.com drive and have Syncthing sync to that.

Anyway, I mean, they've nailed the way this thing works. I mentioned that there were two different crypto things. The other thing is that folders are identified by a unique string to be known to the system, that is, network-wide. But then you decide which local folder you want that to be synced to. So I think that our users, our listeners of this podcast, once they get the hang of this, will love Syncthing and the power it gives them for what it is. It does take, I mean, there's a learning curve. This is not for non-technical users. But for technical users, because the philosophy is right, it is very fast block-level synchronization between devices.

It runs on anything. There are iOS and - I think there are iOS and Android clients. I have to - oh, there is Syncthing for FreeBSD. So I was happy to see that in the FreshPorts collection. I don't know, in my notes here, whether - I don't remember now where it is with mobile apps. But anything that runs - no, no. You don't have access to a browser. I think there are mobile clients, but I'm not sure. Anyway, it does solve the Linux syncing problem and Mac and Windows. And I'm annoyed that I didn't have in my notes here whether it does - let's see. Syncthing, oh, not iOS, but yes to Android. That's what it was. There was something called "fsync" that was an older version of something on iOS, no longer supported.

So we do not have - Syncthing does not currently have an iOS client, but it does have an app in the Google Play Store and also F-Droid. You're able to grab it there. There's a Mac, Windows, Linux cross-platform. It's got GUI wrappers using Syncthing-GTK. And there is a Syncthing for macOS. So FreeBSD, NetBSD, Dragonfly, Solaris, OpenBSD, Windows. The source code is there. It is terrific for interdevice sync. And it is, if I had not discovered Sync.com, Syncthing is what I would be using. Oh, I forgot to mention. I feel a little queasy about using public discovery servers. You are able to run your own discovery server. So if you have any sort of a hosted presence on the 'Net, you can run your own discovery server and point your Syncthing instances at it in order to allow them to find each other if you're roaming around.

You are also able, if for example endpoints have DynDNS or relatively static IPs, you're also able to just simply point these things at each other across the public Internet. But as I said, having found Sync.com, for the price, and remember, 5GB just to play with for free, that's what I'm using for my public presence, Trust No One synchronization. And then Syncthing is what I'm using - oh, and also it handles deep versioning. So, for example, I don't do versioning on my workstation instances of Syncthing, but I have versioning turned on on the Drobos. So that's perfect. That's what you want is all the things I'm doing are automatically being synchronized to the Drobo.

So it's got deep RAID backup and very sophisticated versioning. It's like for the first hour it keeps a copy every 30 seconds. Then for the next day it keeps one every hour. Then for the next week it keeps one every day, and blah blah. So it's like, you know, a whole hierarchical staged file synchronization. So, I mean, which is exactly what you want, so that you're able to go as far back in time as you want to, and all of that is user configurable and tweakable.

So anyway, they nailed it for peer-to-peer synchronization across all of our desktop platforms and Android mobile. And Sync.com, the only thing it's missing, because it does have apps for iOS and Android, the only thing it's missing - and of course you have a web browser interface, as well. The only thing it's missing is Linux desktop support. So that's my Joy of Sync.

Leo: You did look at pCloud, which...

Steve: I did look at pCloud.

Leo: Which has Linux support. If that was a deal breaker, would you consider pCloud? Or are there issues with pCloud that you've seen?

Steve: I don't remember now.

Leo: You looked at a lot of them.

Steve: I looked at so many things, I'm at risk of getting them confused.

Leo: One thing I do like about - it's only 2TB, but they offer a lifetime subscription for 350 bucks, and then you can pay for crypto. You can have arbitrarily synced crypto folders, which is nice. So you can encrypt certain folders, but have other folders unencrypted for things like - because they have audio playback, stuff like that, which obviously encrypted folders you couldn't do. And then the other thing I like, you could sync arbitrary folders, so you don't have to do the junction link thing. You can say this folder is synced, this folder is synced, this folder is synced.

Steve: Ah, that is nice, yup.

Leo: Yeah. And they do have a Linux, as well as iOS, Android, Mac, and Windows clients.

Steve: And pCloud was recommended by some people. So I think, you know, I may have been a little put off by the lifetime, only because...

Leo: Well, you don't have to do that.

Steve: Right, right, right.

Leo: It's the lifetime of pCloud, not of you, obviously.

Steve: Correct.

Leo: So you don't have to do that. In fact, the pricing's very comparable. It's, I think, 100 bucks a year for two terabytes. So it's a little bit less storage. Honestly, though, for most people, two terabytes is infinite, as is probably one terabyte. Microsoft famously called one terabyte "infinite" on its OneDrive offer. It's really nice that there are these good Trust No One choices out there.

Steve: Yes. I was going to say that the world has evolved and matured. I mean, we are, if anything, we are just buried in riches through this whole category.

Leo: Yeah, yeah. It's very exciting. And I think that I will look at all of these. I think, I'll tell you what, I'll give a try to pCloud only because of the Linux client, which really for me is kind of important because I use Linux as much as anything else.

Steve: That does make sense. And when I had tweeted about Sync weeks ago, I did get some listeners say, yeah, looks great except for Linux.

Leo: Yeah, yeah. Really cool. I think this has been so important. By the way, get the show notes. They're available at GRC.com, especially if you're going to set up Syncthing, because Steve does a great job of walking you through that process, better than the documentation, I think. So if you're curious about how to do it, I was a little stymied by it. I've installed it everywhere, but I still haven't hooked it up. But I will be starting to walk through your show notes to get that information. Show notes...

Steve: So I really...

Leo: Go ahead.

Steve: Go ahead.

Leo: I was just going to say, show notes are available, along with 16 and 64Kb audio versions and a transcript, a full transcript at Steve's site, GRC.com. While you're there - he didn't even mention SpinRite. I will. Get yourself a copy of the best file and hard drive recovery and maintenance utility out there. There's only one. It's called SpinRite, and you get it at GRC.com. And since it's Steve's bread and butter, it's how we keep him afloat. So we want to do that.

SQRL information there, as well. And if you're going to see one of Steve's presentations in Europe or our presentation in Boston in a couple of days, you might want to read that first, get an idea of SQRL. Lots of other great stuff, as always, at GRC.com.

We have audio and video of every show. And this is one, this is a keeper. This is an evergreen show. And this will be available at TWiT.tv/sn. That's the Security Now! folder. All the shows are there, audio for every show, video for all the shows for which we recorded video, which is several hundred of them, including this one. You can watch us do the shows.

Normally, we're a little off today, but normally we do these Tuesday, 1:30 Pacific, 4:30 Eastern, 20:30 UTC. And if you are around at that time, the stream is live. You can ask your Amazon Echo to listen to TWiT Live, or you can do that with your Google Assistant, or you can just go to TWiT.tv/live and get an audio or video stream. Join us in the chatroom if you're doing that, irc.twit.tv. And by all means, subscribe in your favorite podcast application so you can start building your superlative collection of Security Nows.

Steve, thanks so much. Great job on this. I think this is going to be a favorite for years to come. See you next time.

Steve: I love sharing good solutions with our listeners. And these are really - I really encourage people to look at Syncthing for what it does, and Sync.com. Or pCloud, if you need Linux. That makes a lot of sense.

Leo: I'll report back. Everything I've seen looks pretty good for pCloud as an alternative.

Steve: Cool.

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

Steve: Thanks, buddy.