RPKI

Description: This week we update on the Apple/Google contact tracing technology. We take a close look at the past week's frenzy over two newly disclosed vulnerabilities in iOS's mail application. We consider the choice of VPN provider relative to expanding global surveillance agreements. We look at some recently spotted dangers of public repositories. We share a bit of miscellany, a SpinRite update, and some useful feedback from a listener regarding Oracle's VirtualBox VM system. We wrap up the week with a look into RPKI (Resource Public Key Infrastructure) for finally bringing some security to BGP, the Internet's critical Border Gateway Protocol.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-764.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-764-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. He's going to talk a little more about the Google/Apple proposal for contact tracing and why it's better than that a lot of countries are proposing. We'll also talk about that email exploit, the zero-click, zero-day on Apple iOS devices. Is it really that deadly? Well, maybe not so fast. Steve will explain. And a look at RPKI in a better way to protect us from BGP router mistakes. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 764, recorded Tuesday, April 28th, 2020: RPKI.

It's time for Security Now!, the show where we cover your safety, security, privacy, technology in general with this guy right here, Steve Gibson of the Gibson Research Corporation. Hello, Steve.

Steve Gibson: And other things that we find interesting, even if they're sometimes a little non sequitur or off-topic.

Leo: Honestly, I think that's the appeal of the show. It's really - it's about Steve.

Steve: Well, actually we do this far less than any other podcast that you produce, Leo. But pretty much we're on target.

Leo: You stick to, yeah, most of my stuff is pretty meandering, yeah.

Steve: We do have a miscellany section where random stuff wanders in. And sometimes it takes over the whole podcast. But not very often. In fact, not today. This is, as I promised last week, RPKI, which is the move toward doing something really important for the Internet, which is working to begin securing BGP, the Border Gateway Protocol. We've spoken many times, and I have some fun examples of mistakes that were maybe inadvertent because it's even more scary if they were deliberate. But it's interesting how fragile the routing management of the Internet turns out to be.

So this is Security Now! 764 for the last podcast of April. We're going to start by updating on the Apple/Google contact tracing technology. They've tweaked it since two weeks ago, when it was the topic of the podcast. But I thought makes sense then to update about the tweaks and how I feel about them two weeks later. And this week we're supposed to get the beta. We're also going to take a close look at the past week's frenzy over two newly disclosed vulnerabilities in iOS's mail application, as you on the previous podcast suspected we would do. And I do think it was probably overblown. We'll consider the choice of a VPN provider relative to expanding global surveillance agreements. And the good news is, one of this network's VPN sponsors, if ExpressVPN is still a sponsor...

Leo: Is still a sponsor. Yes, it is.

Steve: Still a sponsor on the page.

Leo: Yes, it is.

Steve: I did my homework. And they're in the clear...

Leo: Oh, good.

Steve: ...relative to the, what is it now, it's 15 Eyes or something. The Eyes keep multiplying.

Leo: A lot of Eyes.

Steve: Also we're going to look at some recently spotted dangers of public code repositories, taking a specific example. We've got a bit of miscellany, a quick SpinRite update, and a useful bit of feedback from a listener regarding my rave last week of Oracle's VirtualBox VM system, just sort of a caution. And we're going to wrap up by taking a look at what RPKI is. And Leo, we do have a Picture of the Week that is one for the ages. So I think another great podcast for our friends. Ah, yes. So this is a woman who doesn't quite understand the concept of a face mask. She's taken a Norton Antivirus CD and rubber-banded it to her face, hoping that it will...

Leo: But it's so easy to breathe through the little hole in the middle.

Steve: It is much more convenient, and you don't get all steamed up and fogged up and everything that we're putting up with, with our face masks. So anyway, I don't think it will be as effective as she hopes. For one thing, that hole is way too large, you know, it's

got to be down in the microns in order to block the virus. So anyway, someone tweeted that to me, and I thought, well, that's perfect for a little bit of levity.

So the joint Apple/Google virus contact tracing that was our main focus two weeks ago has been gaining traction, I'm glad to say. There are some organizations, some state actors, some countries that have indicated that they're going to adopt it, as opposed to their own ad hoc - I just think, thank goodness, because that's one of the big benefits of there being an established platform, which was...

Leo: I agree. I was talking about this earlier. Australia's trying something that's terrible. France is pissed at Apple because they said you won't give us the data. It's clear that Apple and Google have got it together compared to these nationalities.

Steve: Yeah. And we've seen what happens when government tries to do technology. Famously, in this country, despite all of the technology and lots of money, what was the health system that tried to switch on and was a complete disaster, Obama's...

Leo: Obamacare, yeah, yeah, yeah, the ACA.

Steve: Obamacare, oh, my lord.

Leo: But, you know, it was a good ending to that story. A lot of times people talk about that and, oh, the billion dollar health. But what happened was a bunch of coders from Silicon Valley got together, and the United States Digital Service was born, and they made a much better site. And as a result you've got some really competent people - Matt Cutts is their administrator - doing stuff like for the Veterans Administration and other websites. And these government websites, they really do need to be updated. As you know, that's the one reason TLS 1.2 won't die.

Steve: Well, exactly. And this sort of speaks to the point, which is there are some things you don't want to have amateurs do. It's just like, okay, step away from the computer. We don't want you to try to home roll your own contact tracing system that feeds up to Big Brother. So the idea that, quickly, as I said, this week we're supposed to have this in beta from Apple and Google.

Leo: Today, I think, supposedly.

Steve: Oh, good. That's heading, you know, heading these independent efforts off at the pass, which is where they need to go to die. It's like, okay, just stay away. So I'm already on record saying that, based on my reading of the technology, which I shared with our listeners two weeks ago, I believe that the system is clearly and cleverly designed to achieve its stated goal of allowing someone who's chosen to use this system to obtain a notification if they were probably in proximity, both in space and time, to another user of the system during a time when that other user may have been contagious with this novel coronavirus. And most importantly, this system enables this minimal service while providing maximal protection of the privacy of all participants. Which is why I looked at it, and it passed muster.

Okay. So as we know, they're on schedule to be providing an initial beta. As you said, today. Last Friday they disclosed a number of changes they were making, they said to further enhance the system's existing privacy protections and accuracy. It's like, oh, okay. Well, so we'll look at those. What are there? There's eight of them. So first, the term "contact tracing" has been changed to "exposure notification," which Apple and Google feel better describes the functionality of the API. That's what they're actually offering is exposure notification, which I think is accurate. They said the system may also not stand alone. While it is intended to notify a person of potential exposure, it may be used to augment broader contact tracing efforts that public health authorities are planning.

So again, that sort of reminds us that this is an API. That's actually something, and I'll expand on this a little bit in a minute, that makes me uncomfortable because what you build on top of it could still be doing evil, even if it's on top of this API. And so the fact that they're not offering turnkey apps that I would also tend to trust is a bit of a problem.

Anyway, the second point, they said, keys will now be randomly generated, rather than derived from a temporary tracing key, making it more difficult for someone to guess how the keys are derived, and use that information to try and track people. Okay, that is utter nonsense. But okay. Maybe someone somewhere else raised their hand and said, "I don't know what HMAC means. Could you just make them random? Because I don't know what HMAC means." And so someone at Google and Apple probably just looked at each other and said, yeah, okay, fine. I mean, it doesn't really matter. It means that rather than storing one's secret from which you are able to re-derive the keys from the past, again, there's nothing wrong with that. That was perfect. But "I don't know what HMAC means." Okay, fine.

So third point. Bluetooth metadata will be encrypted, making it more difficult for someone to try and use that information to identify a person. I think that sounds like a great enhancement. Bravo. So, good. Encrypt your Bluetooth metadata.

Fourth point, exposure time will be recorded in five-minute intervals, with a maximum exposure reported capped at 30 minutes. Okay. Not a biggie. Slightly more granular. So a bit of a privacy enhancement, you know, you're not sure was it seven, was it eight, no, we're either going to tell you it was five or 10, and we're not going to tell you if they were there longer than 30. So okay. If there wasn't a cap, you could imagine how you could make an inference if it was like a long exposure. Then it's like, oh, I know who that had to be. So, fine.

Fifth point. The API will include information about the power level of the Bluetooth signal in the data that is exchanged between phones. This can be used in conjunction with RSSI, that's the Received Signal Strength Indication, to more accurately estimate the distance between two phones when contact was made. And I think that's very clever. That's, you know, so the point is the beacon that is sent will also include its transmitted power level so that the receiver can incorporate that along with the received power that its own receiver gets in order to make a better judgment. That's brilliant. Whoever added that, bravo.

Sixth point. Apple and Google will allow developers to specify signal strength and duration thresholds for exposure events. In other words, maybe that was originally just going to - they were going to do that and decide how long and how strong and what the bar was that you had to clear in order to decide that the phones were close enough to relate to viral exposure. Now that'll be a parameter in the API. Okay, fine.

Seventh point, the API will now allow for determining the number of days since the last exposure event to better determine what actions the user should take next. Okay, that's a nice enhancement. I guess that means that that information will be available in the

API. Previously everybody would be checking daily to check to see whether they received any keys that indicated that the people had been near them. Now there'll be an explicit, when was this exposure event? That'll be added to the data flow. So, cool.

And lastly, the API's encryption algorithm is switched from HMAC to AES. They said: "Many devices have built-in hardware for accelerating AES encryption, so this change should help performance and efficiency on phones." That's also total nonsense. I mean, the difference in power, in electrical power, for the minuscule amount of data involved in doing AES with hardware acceleration versus HMAC, pales in comparison to having your screen turned on. I mean, that's complete nonsense. But okay, fine. Again, we want this to be adopted. So by all means, again, "I don't know what HMAC means." Fine. AES, everybody knows that's, ooh, encryption. Encryption good. So it's in there.

So we're in a weird position here. Everyone is talking about the critical need for contact tracing. New York's Governor Andrew Cuomo is planning to hire and train up a major human force to pursue old-school pencil-and-paper contact tracing. Maybe it'll be augmented with this. Maybe this is where you start. Then you follow up with people. I mean, like if you are told you are near somebody through the technology, then you call a hotline, and then someone comes out and interviews you. Who knows how this is going to go? It's going to be interesting to see how this whole thing fares.

But we do have technology now to augment traditional pre-technology contact tracing. Most people have smartphones that are rarely out of their reach. And Bluetooth is well suited to allowing those phones to sense one another's proximity in time and distance. So at the minimum there's an intriguing possibility for automating at least some of this contact tracing challenge. And as I said before we began recording, Leo, I kind of think the whole issue doesn't make much sense to me. What we're learning is that this virus is already everywhere. And the only way, as I've said before, the only way that the numbers we're seeing make sense is that it is pernicious. I mean, New York is still under a complete lockdown, yet they're guessing that they've managed to push the R-naught in full lockdown to 0.8 in one region of New York and 0.9 in another. So in lockdown, just barely below one, which is where it needs to be.

So I guess to me this feels a little bit like the magnetometers that we all now have to walk through at the airport because of 9/11. But, you know, fine. That's another discussion. People appear to think this will be useful, and I salute Apple and Google for so quickly offering a technological foundation, and offering a truly good solution. Which is what I'm sure it is. But as the saying goes, "It's complicated." And as many of those who are, well, and many of those who are opining on this complex technical issue and presumably being read by those looking for some guidance, unfortunately, they're getting important facts very wrong.

For example, I have seen since this announcement, Wired wrote: "Even if the keys that the app uploads to a server cannot identify someone, they could, for instance, be linked with the IP addresses of the phones that upload them. That would let whoever runs that server - most likely a government healthcare agency - identify the phones of people who report as positive, and thus their locations and identities." Except no. We know that's nonsense because cellular phone IPs are extremely ephemeral.

And in fact that's exactly the word Microsoft Research used in a paper exploring the feasibility of geolocating cellular phones by IP. They wrote: "In this study we show that the reasons for geolocation inaccuracy are twofold. First, cell phone IPs are ephemeral, changing rapidly across HTTP requests. As a result, each queried service observes a different IP address for the same device, even though the queries are executed in quick succession within a span of five to 10 minutes." So again, here's Wired scaring people incorrectly.

And the Brookings Institution, in their piece titled - and they really didn't like this at all - titled "Contact-tracing apps are not a solution to the COVID-19 crisis." They wrote: "Some of the contact-tracing frameworks have been designed with security and privacy in mind, to some degree." Well, how refreshing. They said: "The Apple/Google proposal, for example, stores the information about what contacts the device has made on each users' device, rather than reporting that information to a central server, as is the case with some of the other approaches." Okay, well, that's not true.

Then they go on. "This decentralized architecture isn't completely free of privacy and security concerns, however, and actually opens apps based on these APIs to new and different classes of privacy and security vulnerabilities. For example, because contact-tracing apps constantly broadcast health status in connection with a unique, if rotating, identifier, it is possible to correlate infected people with their pictures using a stationary camera connected to a Bluetooth device in a public place." And that of course is absolute utter nonsense.

As we know, the smartphones are not broadcasting a beacon declaring that its owner has been infected with the coronavirus, so stay clear. But unfortunately, because the way the system does work is complicated, and few people outside this podcast apparently have any idea how it actually does work, the public's natural and healthy skepticism is being primed to say "No, thanks." And I know, Leo, you agree that having that happen would be unfortunate. So we understand that what Apple and Google have created is a solid platform which strongly encourages and enables the creation of secure and private contact tracing apps.

Could individual apps built on this API work hard to subvert the system by adding the user's identity to the uploaded reports? Yes. Of course. And this is why I so strongly wish that Apple and Google were going further and were themselves providing the turnkey apps that run atop those APIs. Would I be surprised if the Chinese government's apps running on top of this API did not also embed their citizens' identity into the uploaded content? Of course not. None of us would. But that's not the API's fault, and it's not even theoretically possible to prevent that from being done by any untrustworthy app.

So yeah, the stakes are high, and everyone is freaked out right now. And at the very least, what we've got is a very good start. Virus researchers and epidemiologists are telling us that this will probably not be the last such pandemic we encounter in our lifetimes. Hopefully we'll all be around. Well, yeah. Hopefully it's a ways off, and we'll also still be around for it. But having this extremely well-designed contact tracing platform in place today, for whatever good it may do now, is probably equipping us to better handle the next one.

We have heard that both Apple and Google intend to submerge this into the OS, which says, great, it'll be there next time. We know that Apple has a strong interest in leveraging their iDevices to improve personal health. It would make a lot of sense for a future iOS, iPad OS, and Watch OS to natively incorporate Apple's own trusted and trustworthy contact tracing app. It would sit there unused, but ready for next time. So maybe.

So, you know, they're moving forward. It'll be fun for them to release something. And I'm glad that they did it quickly so that app developers can jump on it, and let's cross our fingers that they follow the clear intent that the API has of honoring people's privacy. Maybe there'll be some open source apps that will get adopted, in which case they could be audited, and we'll be able to see what's inside and what they're doing. That would be good.

Last week two ways of crashing the native mail app in iOS were disclosed. What more of it was there? Unfortunately, thanks to the inflammatory nature of the disclosure by, you

know, I guess it's not a firm we've ever spoken of on this podcast. They've never been on our radar before. It's the company called ZecOps in San Francisco. Their disclosure made a lot of claims. And the tech press had a probably somewhat unjustified field day over this one.

One of the reports started out with the headline: "Zero-Day Warning: It's Possible to Hack iPhones Just by Sending Emails." Then they said: "Watch out, Apple users! The default mailing app preinstalled on millions of iPhones and iPads [in other words, all of them] has been found vulnerable to two critical flaws that attackers are exploiting in the wild [we're not sure about that] at least from the last two years [and maybe longer] to spy on high-profile victims. The flaws could eventually let remote hackers secretly take complete control over Apple devices [there's no evidence of that] just by sending an email to any targeted individual with his email account logged into the vulnerable app." Okay. Meaning you get email.

"According to cybersecurity researchers at ZecOps [that's Z-E-C-O-P-S] the bugs in question are remote code execution flaws [no, but that's okay] that reside in the MIME library of Apple's mail app, first due to an out-of-bounds write bug, and second a heap overflow issue." So that was the way one report began. The trouble here is that, while there are, as we know, many legitimate security concerns, this almost certainly, at least at this level of hype, was not one of them. And this sort of headline-grabbing, crying wolf, damages the security industry's credibility somewhat when it turns out not to have been an issue. I mean, Spectre on Intel has been a little bit like that.

And we also had "iPhone Zero-Day: Don't panic! Here's what you need to know." And "A critical iPhone and iPad bug that lurked for eight years may be under active attack. Malicious emails require little or no interaction. Exploits active since at least 2018." And I have more, but I'll skip them because they're all similarly breathless.

So the full disclosure by this company, ZecOps - which, by the way, in very poor form, has preceded Apple's release of a patch for iOS. Still not available today. It's expected soon. It is in the beta. But it's going to be in 13.4.5, which I checked just a short time ago. I'm at .1, so we're vulnerable. And there's been a complete disclosure of this, as I said, in very poor form. They claimed, without details or proof, that they had found evidence of the bugs being used in the wild against a list of high-profile targets that included - and here they get obscure. Maybe I guess these are clients of theirs based on what I heard you talking about, Leo, on MacBreak. That sounds like it's the case.

They wrote: "Individuals of a Fortune 500 organization in North America; an executive from a carrier in Japan; a VIP from Germany; MSSPs, that's Managed Security Service Providers, from Saudi Arabia and Israel; a journalist in Europe; and, they suspected, an executive from a Swiss enterprise." So that's the way their disclosure went. And in fairness, we know that it's true that forensic backtracking and analysis often leave holes that need to be filled. And it's conceivable that these flaws, which are real, were being used in targeted attacks. But the researchers never managed to actually do that. They managed to use very large emails with MIME attachment embeddings to crash iOS with a heap overflow and an out-of-bounds write.

Both are bad, and either could theoretically lead to a more powerful attack by a sufficiently skilled attacker. And this may be the way this was used, by some nation state-grade attacker. This may have been the way in. And then after that, other exploits were used. But there's no evidence of that. It was never shown. And their own disclosure FAQ, they asked and answered the question they asked themselves: "Why are you disclosing these bugs before a patch is available?"

And their answer was, they said: "It's important to understand the following." Three points: "These bugs alone cannot cause harm to iOS users since the attackers would

require an additional infoleak bug" - that's for ASLR avoidance - "and a kernel bug afterwards for full control over the targeted device. Two: Both bugs were already disclosed during the publicly available beta update." And it's like, what? Wait a minute. We know how closed-mouth Apple is. That's not the case. They may have said "fix some bugs in Mail," but they're saying that all the time.

They continue: "The attackers are already aware that the golden opportunity with MobileMail/mail daemon is almost over, and they will likely use the time until a patch is available to attack as many devices as possible." Well, you've heard me espouse that theory often. So maybe. But there's no evidence of that happening.

"Three: With very limited data we were able to see that at least six organizations were impacted by this vulnerability" - okay, and we should back off and say their phones crashed, but we'll get there in a second - "and the potential abuse of this vulnerability is enormous." Potential, yeah, okay. "We were confident that a patch must be provided for such issues with public triggers ASAP."

So then they said: "It is our obligation to the public, our customers, partners, and iOS users globally to disclose these issues so people who are interested can protect themselves by applying the beta patch, or stop using Mail and temporarily switch to alternatives that are not vulnerable to these bugs. We hope that, with making this information public, it will help to promote a faster patch." In other words, they're saying they have some reason to believe Apple's not moving on this as quickly as possible, yet there's no reason to believe that that's the case. Apple immediately put this thing in beta; and I'm sure, as soon as they know it's safe, we'll all be having to update our iDevices. So I think that these guys were a little more desperate than they should have been to get headlines. Headlines they got. So they'll have to live with the consequences.

Last Friday Apple said that based on the details shared by ZecOps in its report, it could not reach the conclusion that the bug was exploited in the wild. Apple wrote: "Apple takes all reports of security threats seriously. We have thoroughly investigated the researcher's report and, based on the information provided, have concluded these issues do not pose an immediate risk to our users. The researcher identified three issues in Mail; but alone they are insufficient to bypass iPhone and iPad security protections, and we have found no evidence they were used against customers. These potential issues will be addressed in a software update soon. We value our collaboration with security researchers to help keep our users safe and will be crediting the researcher for their assistance."

So the ZecOps research sparked many other dissenting opinions similar to mine, from several interested iOS security researchers who also questioned ZecOps' somewhat self-serving conclusion that the bugs had been successfully exploited in the real world. There were many dubious statements and claims in their disclosure. My favorite, for example, is they said: "Few of the suspicious events even included strings commonly used by hackers, for example, 41414141414141414141." They said that they were saying that hackers used hex 41. Okay, except that hex 41 is uppercase "A," which is what you get when you Base64-encode a region of nulls or zeroes.

Leo: Oh, how interesting. I didn't know that.

Steve: Yeah. And the MIME encoding used by email uses Base64 encoding for binary data. So long runs of 41s is not frequently used by hackers. It frequently appears in MIME encodings.

Leo: Right, because it's a lot of zeroes.

Steve: Yes.

Leo: Yeah. Interesting. Because I've seen those 414141s before, yeah.

Steve: Yeah. That's capital "A," and those are zeroes encoded into ASCII so that you can send binary over an ASCII transport like email. So it was like that. And so...

Leo: I learned something today. That's really neat. That's good.

Steve: So the ZecOps research based its assumption on the evidence of in-the-wild exploitation on crash logs that were found on devices they were inspecting where the crash logs were interpreted as failed attempts to trigger the bug. ZecOps said that the failed exploitation event left an empty email and a crash log on the device. But then, during a subsequent successful exploitation, ZecOps said the attacker would delete the empty emails in order to hide the attacks from the user. Okay. But then why leave the crash logs behind as evidence of the previous failed attempts, if you have the ability to delete things from the phone?

And it's pretty much all like that. They really did seem to be reaching for headlines. And they were never able to produce a working proof of concept. All they showed was that some bugs which have existed for many years in Apple's MIME encoding interpreter could cause iOS to crash. And they found instances of iOS having crashed in the past. So yeah, it's good that the forthcoming release of iOS v13.4.5 will have this fixed. For that, we can definitely thank them.

Leo: I think that's probably Beta 5 of 13.4.2.

Steve: Oh, okay. That makes more sense. That makes more sense.

Leo: We're currently at 1, point 1.

Steve: We're at 1, yes. And I was wondering why...

Leo: Yes, yes, be a big jump. Yeah, I think it's Beta 5 of 13 point whatever point 2.

Steve: So whatever is past...

Leo: 13.4.2.

Steve: Yeah, whatever's past 13.4.1.

Leo: The next one.

Steve: That's the one you want. And until then, I don't think I would worry very much, unless you're a Saudi sheikh or something maybe. But no, probably not.

Leo: This has been a week that we've been quoting your famous - we're going to have to call it "Gibson's Law" or something - "Interpreters are hard."

Steve: Yes, yes. Boy, does that keep paying off.

Leo: Mm-hmm.

Steve: Okay. So TechRadar had this, and I thought sort of they raised an interesting point, something perhaps for VPN users to consider. We've often covered aspects, various aspects of intelligence sharing which involves the so-called "Five Eyes" alliance. It was originally, it turns out, just Two Eyes, established between the U.S. and the U.K. back in the '40s, back in the 1940s. It was an agreement to share intelligence gathered by each other's national intelligence services. And we know that they're like, it's impossible for the U.S. to spy on its own citizens, but the U.K. doesn't have that obligation.

Leo: How handy.

Steve: Yeah. So we'll look at yours if you'll look at ours, and then maybe we'll compare notes. That Two Eyes later expanded to include Australia, New Zealand, and Canada. The intelligence sharing agreement was originally military in nature, designed to give participating nations an advantage in the Cold War. But as we've often noted, today it also encompasses information relating to Internet activity and probably other interesting things. And according to some of Edward Snowden's famous leaked documents, the group later grew to include Denmark, Norway, France, Italy, Belgium, Germany, Spain, Sweden, and the Netherlands, creating what is now the 14 Eyes pact. Whoa.

Leo: I didn't realize it had gotten so big.

Steve: Yeah. And it's now also known as SIGINT Seniors Europe. So they decided that 14 Eyes sounds a little dumb. Let's give it a more fancy-sounding name. So now, although somewhat less formal and official, the members of the 14 Eyes syndicate participate in similar intelligence collaboration activities, which falls outside the legal jurisdiction of any single nation state. This may be significant for our listeners because it means that a VPN endpoint which is being used to deliberately relocate to another country may be conferring less privacy than its user intended.

In their discussion of this issue relating to VPNs, TechRadar noted that the potential privacy issues are amplified by the widespread use of free VPNs, which are more likely to keep activity logs than their paid counterparts, despite claims surrounding zero-log or logless policies. They noted that we know that the information collected and logged could include websites visited, collection timestamps, bandwidth usage, server location, and

even the client's original IP address, all of which is shareable among members of this intelligence pact.

In their coverage of this, TechRadar suggested that to avoid the potential privacy issues connected to the growing number of Eyes, users are advised to opt for a paid VPN with an audited no-logging policy, based if possible also in a country that does not fall under the growing number of Eyes alliance. They noted that two popular services, one which immediately raised my antenna, ExpressVPN and NordVPN, are headquartered in the British Virgin Islands and Panama, respectively, and so avoid any association with the problematic and privacy compromising alliance. And as our listeners know, ExpressVPN conveniently is a long-running sponsor of TWiT and has our recommendation. So anyway, I just ran across that, and I thought, yeah, that would be something that our listeners would probably want to keep in mind.

Leo: Is Switzerland one of the Five, or 15, or whatever Eyes?

Steve: I didn't see them. I think the only S's we have are Spain and Sweden.

Leo: Okay. Because a lot of stuff is hosted in Switzerland, ProtonMail and stuff like that.

Steve: Yes, yes, yes.

Leo: Germany, though, is in the 15 Eyes, I'm sure.

Steve: Oh, jawohl.

Leo: Jawohl. And there are a lot of secure servers. Tutanota I think is my encrypted email server, and they're in Germany. So, hmm, that's interesting.

Steve: Yeah. And what's that messaging app that I liked a lot?

Leo: Oh, yeah, Threema.

Steve: Threema. They're Swiss, also.

Leo: They're Swiss, yeah.

Steve: Yeah.

Leo: Interesting. So, yeah. And, you know, my favorite sync system, Sync.com...

Steve: Sync.com.

Leo: Yeah, where's that?

Steve: They're Canadian. On the other hand, they are completely TNO.

Leo: Well, let's see. If it's end-to-end encrypted, who cares; right? Because they can't give anything to the authorities anyway.

Steve: Right, exactly. Unfortunately, a VPN is not. It's client-to-server encrypted, and then your unencrypted stuff all comes out of a very concentrated, well-known, single point of exit.

Leo: Right.

Steve: Which, you know, very much like the Tor nodes. You just have to know that there's a lot of flies buzzing around those Tor nodes. So, yeah.

Leo: Yup, yup, yup, yup, yup.

Steve: Okay. So no surprise to anyone that there are attacking hackers everywhere. The so-called "typosquatting" attacks have been in the news lately. I've not mentioned them before. I decided, okay, it's worth just mentioning it so that it's on the record. The programming language Ruby describes itself as a "dynamic open source programming language with a focus on simplicity and productivity." It claims to have an elegant syntax that is natural to read and easy to write. What we know is it has become quite popular over the years.

Leo: I love it.

Steve: And has built up a strong following.

Leo: It's a beautiful language, yeah.

Steve: And it's got a really cool history. I won't go into it now. But, I mean, it was deliberately designed to be pretty, I mean, to be elegant. And it pulls from many different interesting automatic - it's in the automatic language class with garbage collection and so forth. As with any popular programming language today, what will arise, and has, is a large and growing public repository of prepackaged add-on modules that can be freely downloaded and incorporated. In this case...

Leo: The RubyGems, they're awesome. Love them.

Steve: Exactly. RubyGems. So a RubyGems package is obtained, downloaded, and installed from the public repository by simply entering the command "gem install

{package name}." The problem is typos in the package's name. It turns out that if instead of entering, for example, "gem install atlas_client" to obtain the correct package, the unwitting Ruby coder entered "gem install atlas-client," the package that's downloaded and installed is the original intended atlas_client plus malware.

The security firm Reversing Labs found that the malicious Ruby gem, that one in particular, had added an apparent image file named aaa.png. And when the package, the RubyGems package atlas_client was run under Windows, the file would be renamed to a.exe and run. Such malware could, of course, do anything that it chose to, having been invited into your system inadvertently. It could encrypt your drive. So this is definitely something you don't want to get.

But in this case the a.exe malware monitors the system's Windows clipboard for text that looks like a cryptocurrency address which typically appears shortly before a cryptocurrency user performs an online transaction. We've talked about these clipboard monitors before. The attacker's own cryptocurrency address replaces the user's when the clipboard contents is pasted into the "Send the money here" field on a cryptocurrency transaction page, thus causing the bad guys to receive the money into their wallet rather than its intended recipient. And of course the malware also adds an entry to the Windows registry for persistence so that it will be loaded every time Windows restarts.

Reversing Labs found more than 725 malicious typosquatted instances of this particular malware within the RubyGems repository. So they, you know, atlas_client was just an example of one. They went through, and they slightly modified. They just looked at, okay, how could somebody mistype this? And they grabbed it, downloaded the original, made the modification, and put it back up, hoping that somebody would mistake a hyphen for an underscore. And the records demonstrated that many people got themselves infected this way. And of course this is just one instance. Who knows what else might be lurking there and elsewhere.

So unfortunately, sometimes you get more than you pay for. These things are free, and I just wanted to remind our listeners to exercise caution. It is, I mean, when I've done Perl stuff, I've grabbed things. I've grabbed them from the official Perl repository. I've been as careful as I can be. But all of this free open source stuff is not wrapped in licenses and code signing and protection. And it's not been scrutinized highly. The problems, when found, are cleaned up quickly. But in the meantime there's some exposure. So anyway, I just sort of wanted to plant a little bit of a "proceed with caution" note.

Leo: I hadn't thought about that. But I do all kinds of package installs, and you do it in Linux, too. You do use Apt or Pacman install. I never thought about a typo and what that could be. That's interesting, yeah.

Steve: Yeah, typosquatting.

Leo: Yeah. I've seen it before with websites.

Steve: Yeah, of course, domain names like crazy.

Leo: Domain names, yeah, interesting.

Steve: So this is just a random little bit of COVID-19 note. I got a tweet from someone, and our listeners will remember that back at the beginning of all this novel coronavirus news I reminded everybody about the possible importance of Vitamin D. I received a tweet from a listener pointing me to an interesting article on the Irish Health website. It referred to a Longitudinal Study on Aging conducted by Trinity College in Dublin. Which is the one landmark I told our listeners - and you'll remember, Leo, that I couldn't figure out - this is going to sound so dumb - how to open the train door.

Leo: You were in Ireland last year to talk about SQRL.

Steve: Yeah. And Lorrie really wanted to spend some time...

Leo: And you're struggling with the door.

Steve: ...at Trinity College. And there was, I don't know, you had to push something and then flip a handle or something. And I was like, trying to do it. Here, you know, Mr. Big Techie. And meanwhile I'm watching the Trinity College go past.

Leo: Bye-bye. Oh, how frustrating.

Steve: So we didn't get to go.

Leo: That's horrible.

Steve: But anyway, it turns out that it's Trinity College that did this longitudinal study. I've got a link to the full PDF of their report and to their press release about the study. The original IrishHealth.com piece sums it up quickly. They said: "Vitamin D may be an important factor in determining the severity of COVID-19 infections, new research from the Irish Longitudinal Study on Aging, which has the acronym TILDA, at Trinity College Dublin has found.

"According to Professor Rose Anne Kenny, principal investigator of TILDA, Vitamin D benefits bone health, muscle health, and the immune system." Then, quoting her, "in addition to a potentially critical role in suppression of the severe pro-inflammatory response which characterizes severe COVID-19 complications." And it says: "As a result of their findings, the researchers are recommending that all nursing home residents in Ireland take Vitamin D."

So anyway, I just wanted to say again it turns out when I checked Twitter, when I was on Twitter to post the show notes link that Leo uses to download the show notes every week, somebody else had tweeted another study showing a correlation between longitude and COVID-19. And as we know, longitude factors in because most people's only supply of Vitamin D is UVB radiation striking their skin. So the further you are away from the Equator, the less strong the sun is, the less Vitamin D you have the opportunity to synthesize endogenously. And glass blocks UVB completely. So now we're all in our homes, not venturing out as much maybe as we normally do, which makes it a little more important.

So anyway, I wanted to say again, just a little reminder, that what's interesting is that it appears that it has an immune effect, immune strengthening to help you not get sick. But on the other side, it appears to dampen that cytokine storm which is one of the things that makes people go critical if they have a severe attack of COVID-19. So anyway, just another little blip.

And Blake Helms tweeted, and I appreciated this, Blake, thank you, although I already thanked him via DM. He said: "Hi, Steve. During the latest episode you mentioned VirtualBox." That of course was last week when I was raving about how impressed I was. He said: "One thing that should be noted is that included with the installer is the VirtualBox Extension Pack. It provides things such as support for USB 2.0 and 3.0 and VM encryption. It is closed source. And while free for personal use, it is not free for commercial use.

"What's more, Oracle uses a highly inclusive definition of commercial use. If a machine is used for any type of commercial work, even if VirtualBox is not part of that work, it's considered a commercial use and thus requires you to settle with Oracle. Because it's a default option, many users don't realize that they are agreeing to the license fee. Later, Oracle shows up, does an audit, and sends you a bill. A local company just settled," he says, "with Oracle for $600K."

Leo: Geez.

Steve: Yikes.

Leo: Oh, Oracle.

Steve: "Because they had employees who installed it thinking it was free. It's banned from the company I work for, along with most other Oracle software, for that reason."

So Blake, thank you. I wanted to share that with our listeners. The good news is I'm using it for DOS. And DOS has never heard of USB anything. And in my approach to minimize everything I install, I looked back, and I thought, no, there's nothing there for me. But anyway, for our listeners, for what it's worth, I'm sure most of our listeners are probably just using their systems and VMs to screw around with and personal, non-commercial. But again, Blake, thank you for the heads-up.

Work is proceeding nicely on SpinRite. I want to make the testing phase as easy as possible for those who are interested in participating. Since SpinTest, like SpinRite, boots and runs on DOS, I've prepared a new version of SpinRite's Windows app which will be able to prepare boot media, installing a bootable system into a diskette or a USB thumb drive, or create an ISO file for burning to an optical disc. So that now exists. I'm currently working to get the first release of SpinTest ready for packaging in its boot-prep installer. So I hope that for next week's podcast I'll have some sense for how compatible this first AHCI driver code that I have written and which is working is across all of our testers' motherboards.

It turns out that there were some people having a problem with the himem.sys driver that FreeDOS uses on some systems. And I had already, back in 2013, experimented with writing my own. So I've decided that's the approach I'll take for the sake of compatibility. And so I'm just now in the process. I created a little ram.exe utility that everyone is playing with right now. It's 1,224 bytes long, and it enumerates all of the RAM available from zero to the 1GB boundary in any machine that it's run on under DOS.

They're all having fun with that. I'm getting a lot of good feedback. I'll incorporate that into SpinTest. So anyway, it's all going really well. And it just feels, Leo, it feels so good to be working on SpinRite.

Leo: I bet, yeah.

Steve: Lorrie and I normally take a walk every day. And I said to her last week, I said, I just - it was hard to describe the lightness that I felt that I was finally doing what I'm actually supposed to be doing.

Leo: Nice.

Steve: Rather than stealing time from it. So anyway...

Leo: That's great.

Steve: Making very good progress.

Leo: Good.

Steve: Okay. RPKI, Resource Public Key Infrastructure. As we know, big iron public Internet routers move the Internet's packet traffic around the Internet. Inside each of these routers is a massive routing table. And oh, my god, the number of entries has gone exponential, from I think I saw a chart from 1991 to 2015. I was annoyed that it didn't have a last five years in it because it was, I mean, it looked like the virus taking off. We've all seen those exponential curves recently. I mean, in a sense it is a virus of IP network subdivisions. The idea is - I'll be using the term "network prefix." And that's how you take, in the case of an IPv4 32-bit IP address, as we know, the most significant nbits of - oh, and there went a sale of SpinRite. Someone's going to be able to upgrade for free soon and be able to play with what I have.

The most significant nbits of the IP address are considered to be the network that all of the machines consuming the least significant side of that IP address are on. So a routing table doesn't have an entry for every IP address, thank goodness. What it has is, it has a technology in classic Internet routing that we've spent podcasts on in the past, back in the early days of the podcast, which tries to match the most number of bits of an incoming packet's destination IP to find the entry in the table that refers to that network. And with that entry in the table is the interface, the outbound interface onto which that packet should be put in order to send it on its way toward wherever that network is on the Internet. And that's all there is. It's really elegant and super cool.

The problem is managing those tables. So that's where Border Gateway Protocol (BGP) comes in. All of the routers maintain persistent BGP protocol TCP connections with each other, that is, with each of the routers that they are peering with, with the routers that are on the other side of the interface and the wire going somewhere. They use BGP to share news of any updates to their routing tables so that changes to the Internet can propagate across the Internet, and routers can maintain proper tables.

However, as we know, since routing changes are shared, and they propagate across the Internet, if bogus routes are either accidentally or deliberately introduced, the Internet will break. And the idea of breaking the Internet is something of a meme. Oh, my god, I broke the Internet. Turns out if you google that, you get hits. But messing up BGP really is one way to break the Internet for real. In their explanation of BGP, Cloudflare cites a couple of perfect examples of true past BGP routing errors.

Under "How BGP Can Break the Internet" they said: "In 2004 a Turkish Internet service provider (ISP) called TTNet accidentally advertised bad BGP routes to its neighbors. These routes claimed that TTNet itself was the best destination for all traffic on the Internet. As these routes spread further and further to more autonomous systems, a massive disruption occurred, creating a one-day crisis where many people across the world were unable to access some or all of the Internet."

And I'll pause here for a bit of nomenclature. And I do cover this in a second. But this autonomous system, that's the designation with a number of somebody who owns a block of IP space. So for example, all of our ISPs typically are - they're autonomous systems, and they have some AS number. Way long ago, when I was talking to Mark Thompson, he was trying to talk me into applying for one, that is, getting an AS number, which would then have allocated me 256 IPs. I would have been a little Class C net. The point is I would have owned those IPs, and they would have been transportable from one bandwidth provider to another.

So, for example, when I left Verio and came to - first I was at XO for a while, and now I'm at Level 3. I could have taken those with me. And what would have happened was that, when I set myself up at a new location, I would have said, "I am autonomous system number something, and this is my block of IPs." They would have put that into their router and advertised that this little network is now at this location. So that's why there's kind of this weird jargon. You advertise a network. You advertise - actually what they're doing is called "advertising a prefix." So they're saying this network prefix, 72.124.something, whatever it would be, dot and then dot star, this network prefix is now here. And so as that propagates out through the Internet, routing tables get updated that change where any traffic matching that prefix will go. It used to go to Verio. Now it comes to Level 3.

Now, none of that happened. I just didn't need 256 IPs. I used to have 64, then I pared down, I have 16 now, and I'm happy. That's plenty. And as IPs have become increasingly scarce, those who do have big blocks are being challenged about how they're using them. There are now IP justification forms that you need to fill out in order to explain how you're using all of those and why you need them. So it's just as well because it would have been sad to part with my own Class C network. But anyway.

They also said: "Similarly, in 2008, a Pakistani ISP attempted to use a BGP route to block" - now, this is deliberately - "attempted to use a BGP route to block Pakistani users from visiting YouTube. The ISP then accidentally advertised these routes with its neighboring autonomous systems, and the route quickly spread across the Internet's BGP network. This route sent users trying to access YouTube to a dead end, which resulted in YouTube being inaccessible for several hours."

So the idea there was Pakistan was trying to do internal BGP to essentially null route the network which was actually owned by YouTube, sending it to some dead IP. When that escaped, they null-routed YouTube, not just for Pakistan, but for the Internet. Whoops. So that needed to get fixed.

And then they said: "There are examples of a practice called" - I'm still quoting from Cloudflare. "There are examples of a practice called 'BGP hijacking,' and it isn't always accidental. In April of 2018, attackers deliberately created bad BGP routes to redirect

traffic that was meant for Amazon's DNS Service. The attackers were able to steal over $100,000 worth of cryptocurrency by redirecting this traffic to themselves."

And they finished: "Incidents like these can happen because the route-sharing function of BGP relies on trust, and autonomous systems implicitly trust the routes that are shared with them. While there have been a number of ambitious proposals intended to make BGP more secure, these are hard to implement because they would require every autonomous system to simultaneously update their behavior. Since this would require the coordination of hundreds of thousands of organizations and potentially result in a temporary takedown of the entire Internet, it seems unlikely that any of these major proposals will be put in place anytime soon." Well, the company known as BBN Technologies - Leo, you'll remember BBN.

Leo: They invented the Internet. Bolt, Beranek and Newman, yeah.

Steve: Exactly. They were originally Bolt, Beranek and Newman, one of the earliest and key participants in the creation of the Internet. All major players on the Internet who obtain their own permanent allocation of IP addresses, as I mentioned, like all of the early Internet originators had autonomous system numbers. BBN's number was one.

Leo: Nice number.

Steve: They were AS1.

Leo: Oh, I like it.

Steve: Who do you think came up with Autonomous Systems? Bet it was the guys that gave themselves the first one. And a BBN employee by the name of Ray Tomlinson is credited with the invention of Internet email. He's the guy who chose the "@" sign as the separator between an account and the mail domain name. So my point is we old-timers all know Bolt, Beranek and Newman, which has now changed its name just to BBN Technologies.

Eight years ago, in February of 2012, RFC 6480 was published by two guys at BBN Technologies. That RFC is titled "An Infrastructure to Support Secure Internet Routing." The RFC's abstract reads: "This document describes an architecture for an infrastructure to support improved security of Internet routing. The foundation of this architecture is a Resource Public Key Infrastructure (RPKI) that represents the allocation hierarchy of IP address space and Autonomous System numbers; and a distributed repository system for storing and disseminating the data objects that comprise the RPKI, as well as other signed objects necessary for improved routing security. As an initial application of this architecture, the document describes how a legitimate holder of IP address space can explicitly and verifiably authorize one or more ASes (Autonomous Systems) to originate routes to that address space. Such verifiable authorizations could be used, for example, to more securely construct BGP route filters."

So the necessary flexibility of BGP allows any route to be originated and announced by any random network, independent of its rights to announce, that is to say, to advertise that route, meaning traffic matching a block of IPs should be sent to it, as opposed to anywhere else. That's the situation we're in today. So we need an out-of-band method to help BGP manage which network can announce which route. The Resource Public Key

Infrastructure is a cryptographic method of signing records that associate a BGP route announcement with the correct originating AS number. As its name suggests, RPKI uses a certificate system similar to secure web browsing that we're all familiar with. But the model breaks down rather quickly. For a web connection to be secure, only two parties, the client and the server, need to play. But to fully secure Internet routing, we sort of have to have an all-or-nothing situation. We need broad, widespread, and thorough adoption of RPKI.

Now, it turns out that's not completely true anymore. If the Internet had stayed hugely disaggregated, that would have been true. But there are an increasing number of major players, like Cloudflare, like Amazon, like Microsoft, like Google, where these major players are carrying a huge - well, and Level 3, like the Tier 1 providers. They are carrying a huge amount of the Internet's traffic. So if only they make sure they don't accept bogus routes, that solves a bunch of the problem.

So we have Internet Routing Registries (IRRs) that are the entities that assign these Autonomous System numbers and the blocks of IP space. There are five of these regional registries with familiar names. There's FRINIC, APNIC, ARIN, LACNIC, and RIPE. And their respective territories are shown on the map that I put in the show notes above. You can see like the entire globe is covered, and it's one, two, three, four, five different colors. These registries cover the world.

All five already provide a means for the registrants, that is, their registrants, to take IP and ASN (Autonomous System Numbers) pairs and get a Route Origin Authorization (ROA) record signed. So just as a website obtains a certificate, signed by a certificate authority, attesting to the certificate holder's ownership of one or more domains, the ROA, this Route Origin Authorization, as its name sounds, is signed by one of the five registries operating as a TA, a Trust Anchor, and attesting to the fact that that Autonomous System's ownership of one or more blocks of IP space is valid.

In a world where no IPs would be routed to an Autonomous System without signed authorization for it to receive that incoming traffic, it's this ROA that will allow an autonomous system to authenticate the routes that it is advertising to the world over BGP. So it doesn't go through BGP. We're leaving BGP alone. This is an out-of-band authentication architecture, an infrastructure that will be allowing a means for authenticating the ownership of IP ranges. Strong support for RPKI as a consequence is what's needed for the future.

But IP-owning organizations are almost certainly going to need a push for its adoption. Recall that we recently spoke about MANRS. That was Mutually Agreed Norms for Routing Security. No surprise. And not surprisingly, this RPKI is one of the things the MANRS group is working toward. And remember that Cloudflare recently joined or was invited to join, and a few others, like that class of Internet traffic carrier, to join the MANRS effort.

Cloudflare said: "The Internet Society has pushed an initiative called MANRS (Mutually Agreed Norms for Routing Security) in order to convince the network operator community to implement routing security. It focuses on filtering, anti-spoofing, coordination, and global validation. The Internet Society is doing a good job in educating networks on the importance of better routing security. While they do educate networks about various aspects of running a healthy BGP environment, it's not an effort that creates any of the required new technologies. MANRS simply promotes best practices, which is a good start, and something Cloudflare can collaborate on. All that said, we think it's simply too polite an effort, as it doesn't have enough teeth to quickly change how networks behave."

So to put a bit more political - and this is not them anymore, this is me. To put a bit more political pressure on recalcitrant Internet Service Providers, Cloudflare has created

a BGP security shaming website called "Is BGP Safe Yet?" And yes, https://isbgpsafeyet.com. Go there with your Internet connection and click the "Test Your ISP" button, and in a few seconds you'll find out whether your ISP is safely ignoring invalid prefixes. The first time I went there, I got a no.

Leo: Oh, no.

Steve: Oh, and I'm still getting a no. That's interesting. My location here says that Cox Communications at AS22773 - you can see we've come a long way from AS1. AS22773 does not implement BGP safely. It should be using RPKI to protect the Internet from BGP hijacks. And then there's a link to tweet this news.

Leo: Shame them, yeah. Look at this. I would expect SonicNet would, but apparently...

Steve: Well, again, this is new, Leo. And it is not without some effort. So I'm not holding anybody to, at this point - well, in fact, on that page is a list of known yeas and nays and some uncertain results. And there are a lot of people, I think Verizon is still not qualified.

Leo: I'm not surprised.

Steve: For doing it. So again, we're still in the Wild West of the Internet. This is one of the problems. The good news is irresponsible people are generally not running a router. And if you misbehave with your router at the AS level, at the Autonomous System level, you can get blacklisted. So you will suddenly have no traffic going to you, and none of your peers - your peers will just disconnect, and nothing you have to say will be useful. So, I mean, it is a privilege to have your traffic accepted by your peers. And with that comes great responsibility.

So believe me, the people who are putting routes into BGP-equipped routers are having triple double-checking of making sure that the asterisk is in the right place and the slash is where it should be. And it's worth noting RPKI is not a bulletproof solution to securing all routing on the Internet. However, it represents what I think is the first milestone in moving from purely trust-based to authentication-based routing. Cloudflare explained that their intention is to demonstrate that it can be done simply and cost-efficiently, and they are inviting operators of critical Internet infrastructure, which is what any Autonomous System is a part of, to follow them in a large-scale deployment of RPKI. And they're suggesting that with this effort, Is BGP Safe Yet, that we lowly end users might help a bit by checking to see how our ISP is doing and perhaps giving them a little public shame or an attaboy with a tweet in the right direction.

And one last note. A guy named Job Snijders, I guess, J-O-B, and his last name is S-N-I-J-D-E-R-S, so I would guess Job Snijders...

Leo: It sounds Dutch, which means it's not even close.

Steve: Yeah, yeah, okay. You know, I'm sorry, Job, but I tried. He's with NTT, and we're going to be hearing his voice, anyone who's interested, because he's presenting a free RPKI 101 webinar. And the good news it's a ways away, two weeks and two days. On

May 14th he, however you pronounce his name from NTT, will present a free webinar. It's May 14th at 8:00 a.m. Pacific time. And this guy is the real deal. His short bio reads: "Job Snijders" - however you pronounce his name - "is IP development engineer at NTT, where he analyzes and architects NTT's Global IP Network for future growth." And of course NTT is a major player. They're one of the big guys. "He's been actively involved in the Internet community in an engineering and architectural capacity" - I almost said "archeological," but we're not that old yet - "architectural capacity as a frequent presenter at network operator events such as NANOG, ITNOG, DKNOG" - these NOGs all stand for Network Operator Group, by the way - "RIPE, NLNOG, and APRICOT, and in a number of community projects over 10 years.

"Job is co-chair of the IETF GROW working group, founder and director of the NLNOG Foundation, contributor to the OpenBSD project, and vice president of PeeringDB. His special interests are routing policy, routing security, and large-scale BGP deployments. He maintains several tools such as 'irrtree' and 'irrexplorer,' and is active in the IETF, where he has co-authored and contributed to RFCs and Internet Drafts."

And is that an understatement. Let's see. I have a list of them. For example, he's the author, RFC 8327, "Mitigating the Negative Impact of Maintenance through BGP Session Culling"; the author, RFC 8212, "Default External BGP (EBGP) Route Propagation Behavior Without Policies"; the author of "BGP Administration Shutdown Communication"; the author of the RFC "The Use of BGP Large Communities"; the author of "Deprecation of BGP Path Attribute values 30, 31, 129, 241, 242, and 243." Anyway, you get the point. And he's like the author of another half dozen. So if this interests you, this kind of BGP arcana, you can listen to him talk about RPKI two weeks and two days from now on May 14th at 8:00 a.m. Pacific time. I'll be there. Sounds like fun.

Leo: Cool. Is this one of those gotchas, though, where they're, you know, it's easy - we've seen a lot of these testers. It's easy to say, oh, everybody should be doing this. And then when it comes down to it it's like maybe a little bit hard to implement, or there's compelling reasons not to implement it, that kind of thing.

Steve: I don't think there's compelling reasons not to. I do think it's probably difficult. And, you know, especially in the middle of coronavirus where everybody seems to be rolling back everything that they were planning to do, this is probably not the time it's going to happen. But, I mean, this looks like the only way to secure what we've got now. As it happens, next week's topic is a deep dive into China's extremely controversial plan for a next-generation Internet.

Leo: Oh. I was wondering if you were going to cover this one.

Steve: Yup.

Leo: This is the one with the kill switch.

Steve: That's right. It changes many assumptions we have come to take for granted. And I've got the documentation at the technical level, so we're going to do a deep dive into what China is planning. It's probably never going to happen, but it'll be interesting to see, I mean, they're certainly big.

Leo: Well, they presented it to the ITU. I think they understand they can't just unilaterally do it unless they want to be cut off from the rest of the world.

Steve: Right. Yeah, they're proposing it. It's like, hey, how about this?

Leo: What if we did this? We've got this problem with dissidents and people we don't like publishing pictures of Winnie the Pooh, comparing it to President Xi. We wish we could just flip a switch and turn those sites off. How about it, guys?

Steve: That pesky freedom of expression; you know?

Leo: I hate it when that happens.

Steve: When those packets come, and you don't know where they came from.

Leo: Good, I can't wait. I'm actually very interested. I read about that a few weeks ago, and I was hoping you would. See, as I'm browsing the Internet, I always go, oh, I hope Steve talks about that. Oh, I'd like to know more about that. I count on you, Steve. You're the one who explains all this. That's why we...

Steve: Leo, I have a feeling you're not alone. And for that I really thank our listeners every week.

Leo: Explainer in Chief, this guy right here. You'll find Steve at his website, @SGgrc. No, no, that's your Twitter handle, @SGgrc. His website is just GRC.com. You'll find all sorts of great stuff there, including 16Kb versions of the show, squinched heavily to fit into your briefcase. There's also beautifully written transcripts by Elaine Farris so you can read along as you listen. There's also 64Kb audio, if you have a bigger briefcase. You can find audio and video at our website, TWiT.tv/sn.

When you're at GRC.com, though, do take a look at SpinRite. Now would be a really good time to get in on SpinRite 6 so you can participate in the testing for the next generation, which as you heard is imminent. That's exciting.

Steve: Well, and I've told people, if they want to make sure that the next SpinRite runs on what they have...

Leo: Yeah, that's a good point.

Steve: ...then this is a way to do it. Because if it doesn't, I'll make sure it does.

Leo: Don't make promises you don't want to keep, Steve. There's probably a few things, you know, I've got this disk pack, it's a 5MB IBM Bernoulli.

Steve: Well, actually we were recently just talking about Zip and Jaz drives and their operation.

Leo: As long as there are sectors, right, you can read them.

Steve: Yeah.

Leo: What else is there? Oh, ShieldsUP!. More Vitamin D info. There's a ton of stuff there: GRC.com. That @SGgrc, I keep saying that, that's his Twitter. And somebody was asking in the chatroom earlier, how do I email Steve? I said, don't email Steve. That's an exercise in futility. You either go to GRC.com/feedback and leave a message there, or you tweet him. He takes Direct Messages from anybody, crazy guy. So you can leave a DM for Steve at @SGgrc on the Twitter.

We do this show every Tuesday, 1:30 Pacific, that's 4:30 Eastern, 20:30 UTC. If you want to watch us do it live, TWiT.tv/live is the place. There's audio and video streams there. You can also ask your Amazon Echo, "Play TWiT Live on TuneIn," and it'll play. I do that all day so I can listen all day and see what's going on. It's kind of nice to have that in the background, especially as we're stuck at home.

Subscriptions to our podcasts are also welcome. It helps us because it gives a more consistent number of downloads every week. So all you have to do is find your favorite podcast app - Stitcher, Slacker, Pocket Casts, Overcast, Google Podcasts, Apple Podcasts, you know. You know the drill. Subscribe to Security Now!. You also want every copy. Even if you can't listen this week, you could listen to two next week. It's always good to have them all. One daisy chains to another. Each show builds upon the knowledge gained from the previous show. You need them all is my point.

Steve, I hope you have a wonderful week. Enjoy - what is it you're going to watch? "Devs" and...

Steve: "Devs" and "Under the Loop" or "Above the Loop" or something about the loop on Amazon Prime. I will have reviews of those next week because...

Leo: "Tales From the Loop."

Steve: "Tales From the Loop" is on Amazon Prime, available on Amazon Prime. It looks great. And you can speak to "Devs." We were talking about a couple weeks ago.

Leo: Yeah. Couple of good sci-fi shows, worth watching. All right, Steve. Have a great week.

Steve: Okay, my friend. Right-o.

Leo: Talk to you next week. Stay safe.

Steve: Bye.