Security Now! #764 - 04-28-20

RPKI

This week on Security Now!

This week we update on the Apple/Google contact tracing technology. We also 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. And we look at some recently spotted dangers of public repositories. We have a bit of miscellany, a SpinRite update and some useful feedback from a listener regarding Oracle's VirtualBox VM system. Then 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.

This may not be as effective as she hopes...

Security News

Contact Tracing Update

The joint Apple/Google virus contact tracing that was our main focus two weeks ago has been gaining traction. I'm already on record saying that based upon my reading of the technology, which I shared with our listeners two weeks ago, I believe that the system was clearly and cleverly designed to achieve its stated goal of allowing someone who has chosen to use this system to obtain a notification if they were probably in proximity to another user of the system during a time when that other user may have been contagious with the novel coronavirus.

And, most importantly, that this system enables this minimal service while providing maximal protection of the privacy of all participants.

This week, Apple and Google are on schedule to be providing initial beta support for the system, and as a result of feedback from around the world, last Friday they disclosed a number of changes they were making to further enhance the system's existing privacy protections and accuracy:

- The term "contact tracing" has been changed to "exposure notification," which Apple and Google feel better describes the functionality of their upcoming API. 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.

- 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. ***[This is nonsense, but... okay.]***

- Bluetooth metadata will be encrypted, making it more difficult for someone to try and use that information to identify a person. ***[That seems like a useful enhancement.]***

- Exposure time will be recorded in five minute intervals, with the maximum reported exposure time capped at 30 minutes. ***[Again, not a biggie. Slightly more granular information.]***

- 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 the RSSI ("Received Signal Strength Indication") to more accurately estimate the distance between two phones when contact was made. ***[Ah! That's a clever addition. The Transmission power level.]***

- Apple and Google will allow developers to specify signal strength and duration thresholds for exposure events.

- 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.

- The API's encryption algorithm is switching from HMAC to AES. Many devices have built-in hardware for accelerating AES encryption, so this change should help performance and efficiency on phones. ***[That's nonsense, too, but... okay.]***

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 "pencil and paper" contact tracing. It's going to be interesting to see how that fares. But we do have technology. 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 there's certainly an intriguing possibility for automating at least some of the contact tracing challenge. Frankly, I think the whole contact tracing goal is completely wrongheaded, like making everyone now walk through a magnetometer at the airport. But that's another discussion. People appear to think it will be useful, and I salute Apple and Google for so quickly offering the technological foundation of a truly good solution. Which is what I'm sure this is.

But, as the saying goes… "It's complicated." And many of those who are opining on this complex technical issue, and presumably being read by those looking for some guidance, are getting important facts very wrong. For example, 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 health care 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 highly ephemeral. 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* ***in****accuracy are two-fold. 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 ten minutes."*

Similarly, the Brookings Institution, in their piece titled: "Contact-tracing apps are not a solution to the COVID-19 crisis" wrote:

> *"Some of the contact-tracing frameworks have been designed with security and privacy in mind, to some degree. 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. 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. The smartphones are not broadcasting a beacon declaring that its owner is 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 works, the public's natural and healthy skepticism is being primed to say no thanks. And that's unfortunate.

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 work hard to subvert the system by adding the user's identity to the uploaded reports? Yes, of course. This is why I so strongly wish that Apple and Google were going further and were, themselves, providing the turn-key apps that run on top of those APIs. I would be surprised if the Chinese government's apps running atop this API did NOT also embed their citizen's identity into the uploaded content. 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, yes... the stakes are high and everyone is freaked out right now. And at the very least this 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. So 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 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, iPadOS and WatchOS to natively incorporate Apple's trusted and trustworthy contact tracing app. It would sit there, unused but ready for the next time.

Two crashes were discovered in iOS native eMail app. Was there more to it?
Thanks to the inflammatory nature of the disclosure by an obscure security firm, the tech press had a probably unjustified field day over this one. For example...

> Zero-Day Warning: It's Possible to Hack iPhones Just by Sending Emails
>
> Watch out Apple users! The default mailing app pre-installed on millions of iPhones and iPads has been found vulnerable to two critical flaws that attackers are exploiting in the wild, at least, from the last two years to spy on high-profile victims. The flaws could eventually let remote hackers secretly take complete control over Apple devices just by sending an email to any targeted individual with his email account logged-in to the vulnerable app.
>
> According to cybersecurity researchers at ZecOps, the bugs in question are remote code execution flaws that reside in the MIME library of Apple's mail app—first, due to an out-of-bounds write bug and second, is a heap overflow issue.

The trouble here is that, while there ARE many legitimate security concerns, this almost certainly was not one of them. And this sort of headline grabbing crying wolf damages the security industry's credibility.

We also had: *"iPhone zero day – don't panic! Here's what you need to know"*

"A critical iPhone and iPad bug that lurked for 8 years may be under active attack
Malicious emails require little or no interaction; exploits active since at least 2018."

"Apple investigating report of a new iOS exploit being used in the wild. Cyber-security firm ZecOps said today it detected attacks against high-profile targets using a new iOS email exploit."

The full disclosure by ZecOps -- which, by the way, in very poor form, **has preceded Apple's release of a patch for iOS** (which is expected soon) -- 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:

- Individuals from a Fortune 500 organization in North America
- An executive from a carrier in Japan
- A VIP from Germany
- MSSPs (managed security service providers) from Saudi Arabia and Israel
- A Journalist in Europe
- Suspected: An executive from a Swiss enterprise

In fairness, it's true that forensic backtracking and analysis often leaves 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. But that was never shown. In their own disclosure FAQ they asked and answered: https://blog.zecops.com/vulnerabilities/youve-got-0-click-mail/#post-faq

> Q: Why are you disclosing these bugs before a patch is available?
>
> A: It's important to understand the following:
>
> - These bugs alone cannot cause harm to iOS users – since the attackers would require an additional infoleak bug & a kernel bug afterwards for full control over the targeted device.
>
> - Both bugs were already disclosed during the publicly available beta update. The attackers are already aware that the golden opportunity with MobileMail/maild is almost over and they will likely use the time until a patch is available to attack as many devices as possible.
>
> - With very limited data we were able to see that at least six organizations were impacted by this vulnerability – and the potential abuse of this vulnerability is enormous. We are confident that a patch must be provided for such issues with public triggers ASAP.
>
> 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 to use 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.

However, 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 said:

> "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."

The ZecOps research sparked other dissenting opinions 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 are many dubious statements and claims. For example, their report said: "Few of the suspicious events even included strings commonly used by hackers (e.g. 414141…4141)" except that hex 41 (0x41) is uppercase 'A' which is what you get when you Base64-encode a region of nulls or zeroes... and the MIME encoding used by eMail uses Base64 encoding for binary data. So long runs of 41's is not "frequently used by hackers" it "frequently appears in MIME encodings."

The ZecOps research based its assumption of the existence of in-the-wild exploitation on crash logs found on the device they were inspecting where the crash logs were interpreted as failed attempts to trigger the bug. ZecOps said that the failed exploitation 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 why then leave the crash logs behind as evidence of the previous failed attempts?

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.

Something for VPN users to perhaps consider???
https://www.techradar.com/news/exclusive-millions-of-vpn-users-endangered-by-this-cross-border-intelligence-pact

We've often covered various aspects of intelligence sharing which involves the so-called Five Eyes alliance. It was originally just two eyes, established between the US and the UK in the 1940s to share the intelligence gathered by each other's national intelligence services. It 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.

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 the Fourteen Eyes pact which is also known as SIGINT Seniors Europe. So now, although somewhat less formal and official, the members of the Fourteen 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 VPN's, 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 and we know that the information collected and logged could include websites visited, connection timestamps, bandwidth usage, server location and even the client's original IP address - all of which is sharable among members of the 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, in a country that does not fall under the Fourteen Eyes alliance.

They note that two popular services, Express VPN and Nord VPN 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 is a long running sponsor of TWiT and has our recommendation. https://www.expressvpn.com/twit

There are attacking hackers everywhere.
The so-called "TypoSquatting" attacks deserve a mention and a caution.

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 that it is quite popular and has, over the years, built up a strong following.

And as with any popular language today, there is a large and growing public repository of prepackaged add-on modules that can be freely downloaded and incorporated into Ruby projects. In the case of Ruby, these are, of course, called "RubyGems".

A RubyGems package is obtained, downloaded and installed from the public repository by entering the command: “gem install {package name}.” The problem is, typos in the package's name. It turns out that if instead of entering “gem install atlas_client” to obtain the correct package, the unwitting Ruby coder enters “gem install atlas-client” the package that's downloaded and installed is the original intended “atlas_client” with malware added.

The security firm, Reversing Labs, found that the malicious RubyGem had added an apparent image file named aaa.png. And when the package 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 -- such as 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. As we've seen before with these clipboard monitors, 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... 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, to reload it whenever Windows restarts.

Reversing Labs found more than 725 malicious typo-squatting instances of this particular malware within the RubyGems repository and the repository maintainers have removed all those that were found. But who knows what else might still be lurking there, and elsewhere?

So this should serve as a useful cautionary warning for those who routinely download packages from online repositories. They are free, but unless you're careful you may be getting more than your money's worth.

Miscellany

Back at the beginning of all this novel Coronavirus news I reminded our listeners about the possible importance of Vitamin D. I received a tweet from a listener pointing me to an interesting article on the IrishHealth website. It referred to a Longitudinal Study on Ageing conducted by Trinity College Dublin.

Trinity College Dublin: The Irish Longitudinal Study on Ageing (TILDA): **https://tilda.tcd.ie/**

https://tilda.tcd.ie/publications/reports/Covid19VitaminD/index.php
https://tilda.tcd.ie/publications/reports/pdf/Report_Covid19VitaminD.pdf

"Vitamin D linked to COVID-19 mortality"
http://www.irishhealth.com/article.html?id=27163

> Vitamin D may be an important factor in determining the severity of COVID-19 infections, new research from the Irish Longitudinal Study on Ageing (TILDA) at Trinity College Dublin has found.
>
> According to Prof Rose Anne Kenny principal investigator of TILDA, vitamin D benefits bone health, muscle health and the immune system, "in addition to a potentially critical role in suppression of the severe pro-inflammatory response which characterises severe COVID-19 complications."
>
> As a result of their findings, the researchers are recommending that all nursing home residents in Ireland take Vitamin D.

We have established that no one should drink or inject any sort of cleanser into their bodies. But one thing everyone can safely do -- though by all means check with your physician before making any change to your supplement regimen -- is to take a useful amount of Vitamin D3. And by useful amount I mean at least several thousand IU of vitamin D3 per day. We all know that I'm not a doctor and that I have no formal medical training. But I am well read on a number of health-related topics, Vitamin D among them. I've been taking 5,000 IU of Vitamin D3 per day without fail ever since I learned of its importance to human health. And I encourage all of my friends to do the same thing. You'll find that capsules of more than 5,000 UI are available. But no one should take more than 5,000 IU per day without keeping a close watch on their blood levels. As an experiment I've done that, taking 10,000 IU per day. And over the course of several months my measured blood level slowly crept upward. My curiosity was satisfied and I've settled on 5000 IU.

Closing the Loop

Blake Helms @helmsb
Hi Steve! During the latest episode you mentioned VirtualBox. 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 support as well as 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 isn't 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 with Oracle for $600k because they had several 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.

SpinRite

Work is proceeding nicely on SpinRite. I want to make testing as easy as possible for those who are interested in participating. Since SpinTest, like SpinRite, boots and runs on DOS, I have 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 USB thumb drive, or create an ISO file for burning to an optical disc:

So, I am currently working to get the first release of SpinTest ready for packaging in its boot-prep installer. I hope that for next week's podcast I'll have some sense for how compatible this first AHCI driver is with all of our tester's motherboards.

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 which the router uses to lookup the best interface to send any packet out of which arrives on another interface.

All of these routers also speak BGP, the Border Gateway Protocol. They maintain persistent BGP TCP connections with each of their peering routers and using BGP over these links they share the news of any updates to the routing tables that they are given or receive from other routers.

However, as we know, since routing changes are shared, they propagate across the Internet. And if bogus routes are either accidentally or deliberately introduced the Internet will break. The idea of "breaking the Internet" is something of a meme... 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:

How BGP can break the Internet

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 1-day crisis where many people across the world were not able to access some or all of the Internet.

Similarly, in 2008 a Pakistani ISP attempted to use a BGP route to block Pakistani users from visiting YouTube. The ISP then accidentally advertised these routes with its neighboring AS's 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.

These 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.

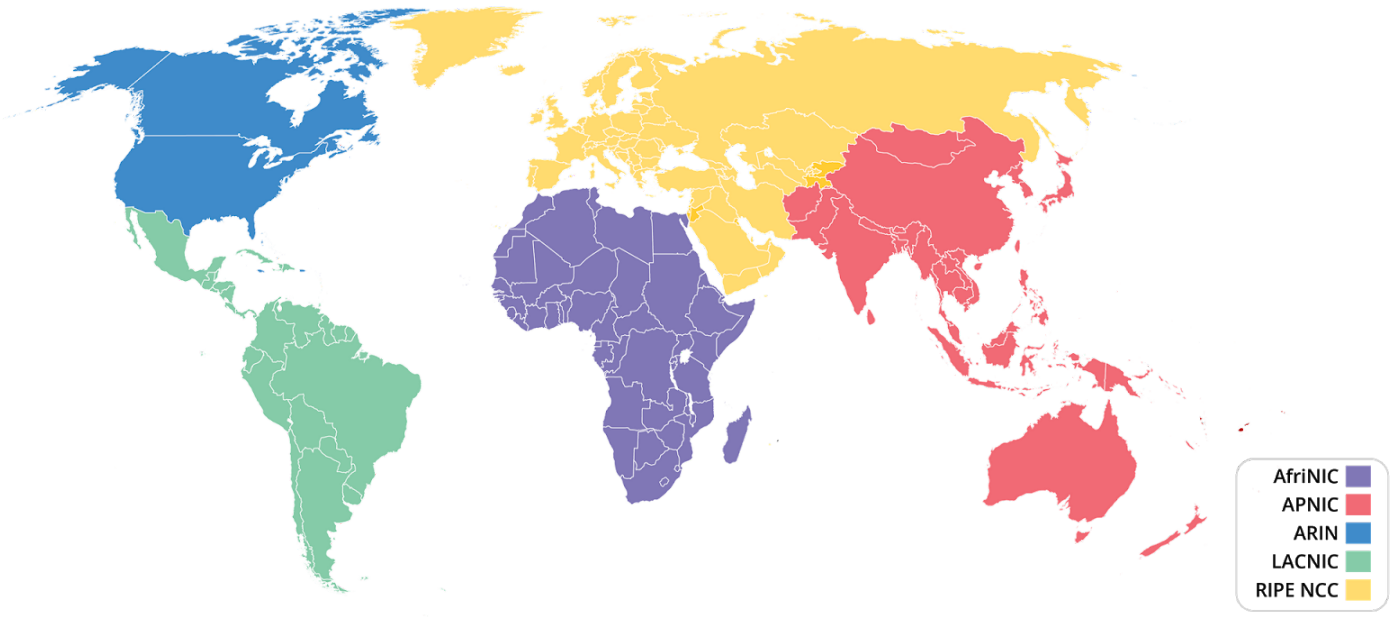
Incidents like these can happen because the route-sharing function of BGP relies on trust, and autonomous systems implicitly trust the routes that are shared with them. While there have been a number of ambitious proposals intended to make BGP more secure, these are hard to implement because they would 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 into place anytime soon.

The company today known as "BBN Technologies" was originally Bolt, Beranek and Newman Inc., 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 -- like each of the early Internet originators had -- also have what's known as an Autonomous System (as AS) number. BBN was AS1. And a BBN employee by the name of Ray Tomlinson is credited with the invention of Internet e-mail. He's the guy who chose the '@' sign as the separator between an account and an mail domain name. My point is, old timers all know of Bolt, Beranek and Newman, now just BBN Tech.

BGP security is an ongoing problem. So 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: https://tools.ietf.org/html/rfc6480

> 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 (AS) 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 to originate routes to that address space. Such verifiable authorizations could be used, for example, to more securely construct BGP route filters.



The necessary flexibility of BGP allows any route to be originated and announced by any random network, independent of its rights to announce that route. 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 (RPKI) 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. 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 an all or nothing situation. We need broad, widespread and thorough adoption of RPKI.

Internet Routing Registries (IRRs) are the entities that assign Autonomous System numbers and blocks of IP space. There are five of these regional registries with familiar names: FRINIC, APNIC, ARIN, LACNIC & RIPE. Their respective territories are shown on the map above.

All five already provide a means for the registrants to take IP/ASN 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 -- route origin authorization -- is signed by one of the five registries operating as a TA -- a "Trust Anchor" -- and attesting to that Autonomous System's ownership of one or more blocks of IP space.

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 allows an Autonomous System to authenticate the routes that it is advertising to the world over BGP.

Strong support for RPKI 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 -- Mutually Agreed Norms for Routing Security. Not surprisingly, this is one of the things the MANRS group is working toward.

Cloudflare, a recent addition to The Internet Society's MANRS effort wrote:

> 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. **That all said, we think it's simply too-polite an effort as it doesn't have enough teeth to quickly change how networks behave.**

To put a bit more political pressure on recalcitrant Internet Service Providers, Cloudflare has created a BGP shaming website called: "Is BGP Safe Yet?" https://isbgpsafeyet.com/

Simply go there with your Internet connection and click the "Test Your ISP" buttons and in a few seconds you'll find out whether your ISP is safely ignoring invalid prefixes. In my case it said:

Your ISP (Cox Communications, AS22773) implements BGP safely. It correctly drops invalid prefixes

And I was given a button to tweet these results to the world.

RPKI is not a bullet-proof solution to securing all routing on the Internet, however it represents the first milestone in moving from 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 to follow them in a large scale deployment. And they are suggesting 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 nudge in the right direction.

One last note: A guy named Job Snijders from NTT will be presenting a free RPKI 101 webinar

On May 14th, Job Snijders from NTT will present a free RPKI 101 webinar in two weeks and two days, on May 14th at 8am Pacific Time. This guy is the real deal. His short Bio reads:

> Job Snijders is IP Development Engineer at NTT, where he analyzes and architects NTT's Global IP Network (GIN) for future growth. He has been actively involved in the Internet community in an engineering and architectural capacity, as a frequent presenter at network operator events such as NANOG, ITNOG, DKNOG, RIPE, NLNOG & APRICOT, and in a number of community projects for over 10 years. Job is co-chair of the IETF GROW working group, founder & director of the NLNOG Foundation, contributor to the OpenBSD project, and vice president of PeeringDB.
>
> Job's 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 or contributed to RFCs and Internet-Drafts.

Job has contributed to the following Internet-Drafts and RFCs:

RFC 8327 (author) - Mitigating the Negative Impact of Maintenance through BGP Session Culling
RFC 8326 (contributor) - Graceful BGP Session Shutdown
RFC 8212 (author) - Default External BGP (EBGP) Route Propagation Behavior without Policies
RFC 8203 (author) - BGP Administrative Shutdown Communication
RFC 8195 (author) - Use of BGP Large Communities
RFC 8111 (contributor) - LISP Delegated Database Tree
RFC 8093 (author) - Deprecation of BGP Path Attribute values 30, 31, 129, 241, 242, and 243
RFC 8092 (author) - BGP Large Communities
RFC 8060 (author) - LISP Canonical Address Format (LCAF)
RFC 7999 (author) - BGP BLACKHOLE Community
RFC 7908 (contributor) - Problem Definition and Classification of BGP Route Leaks
RFC 7789 (contributor) - Impact of BGP Filtering on Inter-Domain Routing Policies
RFC 7059 (contributor) - A Comparison of IPv6-over-IPv4 Tunnel Mechanisms
RFC 6830 (contributor) - Locator/ID Separation Protocol (LISP)

RPKI 101 with Job Snijders:
https://www.brighttalk.com/webcast/5648/396013/rpki-101-with-job-snijders

Next Week

Next Week: We take a deep dive into China’s extremely controversial plan for a next-generation Internet. It changes many assumptions we have come to take for granted.