December 2021
Geoff Huston

Some Notes from RIPE 83

The RIPE community held a meeting in November. Like most community meetings in these Covid-blighted times it was a virtual meeting. Here's my notes from a few presentations that piqued my interest. All the material presented at the meeting can be found at https://ripe83.ripe.net/.

Vulnerability Disclosure

Responsible Disclosure is a vexed topic these days. Humans are pretty shocking programmers at the best of times, but when the environments into which this code is deployed is unpredictable then the chances of errors in the code rises to very quickly to 100% probable! As Quinn Norton said in 2014 (https://medium.com/message/everything-is-broken-81e5f33a24e1) in a painfully frank, but depressingly accurate, article about just how totally broken this mess had become by then (and it's only got worse since): "When we tell you to apply updates we are not telling you to mend your ship. We are telling you to keep bailing before the water gets to your neck."

What should you do if you find a bug in some widely deployed code? Do you a) keep quiet, b) write an exploit to test just how effective it is, and bring down large parts of the Internet, c) join the dark side and sell it as a 0day exploit to some cyber crims, or perhaps to a interested government or two (https://www.nytimes.com/2013/07/14/world/europe/nations-buying-as-hackers-sell-computer-flaws.html), or d) quietly inform the code author/maintainer/vendor in the hope that they are listening and want to fix it, e) claim a bug bounty from a willing vendor, f) inform your local CERT or g) tell everyone as loudly as you can and let everyone else scramble to fix the exposed vulnerability. All these approaches have been tried in the past and the results have been mixed.

As Giovane Moura pointed out in his presentation on this topic, there is a wealth of conflicting advice as to what is the best course of action here. Bruce Schneier advocated back in 2007 that full public disclosure of vulnerabilities is a damned good idea, on the basis that it only really gets attention and gets fixed if the vulnerability is widely known, and patches are released in a timely fashion (https://www.schneier.com/essays/archives/2007/01/schneier_full_disclo.html). Of course, you can inform the software vendor and let them control the timetable of disclosure and undertake not to tell anyone else yourself. Again, good luck with that. Few vendors come out and claim responsibility for their mistakes voluntarily, even if it is, their message is invariably associated with the dubiously reassuring message that "We've patched a bug. We are so good!" And of course, the individual who spent the time and effort finding the problem gets no credit for their work. But is the current practice of advanced warning and timed disclosure any better? "Hi, you have a problem in your code. You have until Monday next week to deploy a patch to the millions of devices running your code because that's when we go public. Have a good day!" I don't think there is a good path through all this. I'm not even sure that there is a least bad path.

Bruce added: "I don't want to live in a world where companies can sell me software they know is full of holes or where the government can implement security measures without accountability." Sadly, we are all living in this world of knowingly shitty software, replete with unpatched bugs and full of as-yet undiscovered vulnerabilities. And we still have no real idea of how to cope. If you look hard enough at

traffic on the Internet, you will still find the echoes of supposedly long-gone viruses and worms. Conficker is still out there, as is Slammer even though these were supposedly patched more than a decade ago.

I'm with Quinn Norton: Everything is broken. So broken that we really have no idea how to fix it. All we appear to be doing is our best to make it even worse!

Why QUIC?

For those who came in late, QUIC is a novel way of packaging the transport protocol for the Internet. Instead of using an unencrypted TCP transport header in an IP packet QUIC uses a simple UDP packet header and places the TCP control frame inside the UDP payload. And then QUIC encrypts the entire UDP payload. Not only does this make the data opaque top the network, it makes the TCP control plane opaque to the network. By using encryption, QUIC restores the now venerable network stack model that said that the transport protocol is none of the network's business.

Why are we so enthusiastic about QUIC? Why are we looking at DNS over QUIC, BGP over QUIC and of course HTTP over QUIC? Because as far as we can tell it's the best response the application layer has available to it in order to keep intrusive middleware and meddlesome networks out. It neatly leaps over today's messy middleware fiasco and the consequent growing ossification of networks that made the deployment of new end-to-end protocols completely impossible in the public Internet.

It was unsurprising to hear a presentation from a large mobile network service operator, Orange, decrying this situation where their visibility into user session traffic patterns has been slammed shut with QUIC. Some 30% of Orange's traffic is now passed over QUIC. The presentation pointed to a new so-called balance of power within the IETF where privacy concerns strongly expressed by Facebook and Mozilla, with support from Google, Microsoft and Apple appear to completely dominate the consideration of all proposals that are bought to the IETF. This presentation bemoans the likely fate of their modest proposal to increase the level of visibility into the QUIC session flow with a 2 bit "loss indicator", as a complement to the earlier "spin bit".

One view is that 2 more transport bits in the clear does not represent a major incursion into the privacy properties of QUIC, and this request for further clarity of QUIC behaviour as a packet traverses networks is helpful to both the network operator, the user and the application that uses QUIC. A contrary view is that these two bits represent one more step along a path of progressive compromises that once more ends up gratuitously exposing user's actions to the network and various onlookers.

QUIC was a response to increasing sensitivity about the widespread eavesdropping abuse of users in the Snowden papers. Admittedly, it's a quite extreme response, as it brings the shutters down in a pretty comprehensive way, but at the same time the network operators were caught out not only looking, but also caught out attempting to extract information from these visible network flows and monetising it in all kinds of ways, all without the users' knowledge let alone informed consent.

While much of the Internet's original architecture was based on pervasive concepts of mutual trust (just look at routing) we are now re-drafting this architecture. Once trust is abused, then all trust is over. Applications no longer have any desire to expose any more than the minimum set of essential items of information to any other party. Not to other applications, not to the application platform and certainly not to the network. In this case it's the endpoint IP addresses and UDP port numbers is all that is exposed to the network. Any more exposed information is just too much.

Developing World Infrastructure

There was an interesting statistic in a presentation about Janata Wifi in Bangladesh, but first let me show a slightly different side of this story, on the retail cost of Mobile Data in a number of economies from a presentation by Benedict Evans (Slide 107, The Great Unbundling, January 2021, https://www.ben-

evans.com/presentations). It appears that when compares to China and the US the cost of mobile data in India is far cheaper (Figure 1). As Benedict then points out this lower retail cost is associated with far higher relative use of mobile data services.

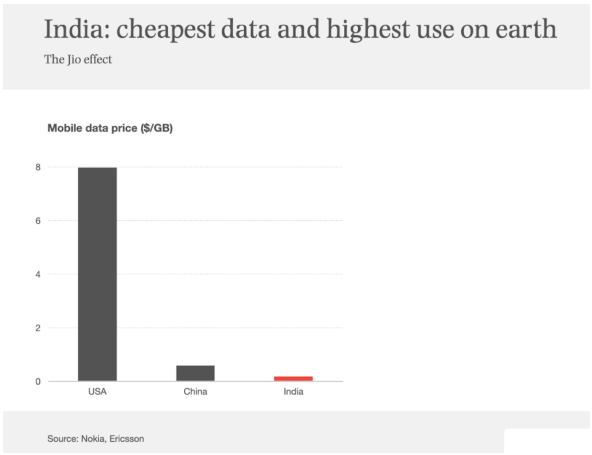

Figure 1 – 2020 Retail cost of Mobile Data in US, China and India, from Benedict Evans, The Great Unbundling

What makes this retail price comparison so notable is the comparison of the radio spectrum costs, shown in Figure 2.

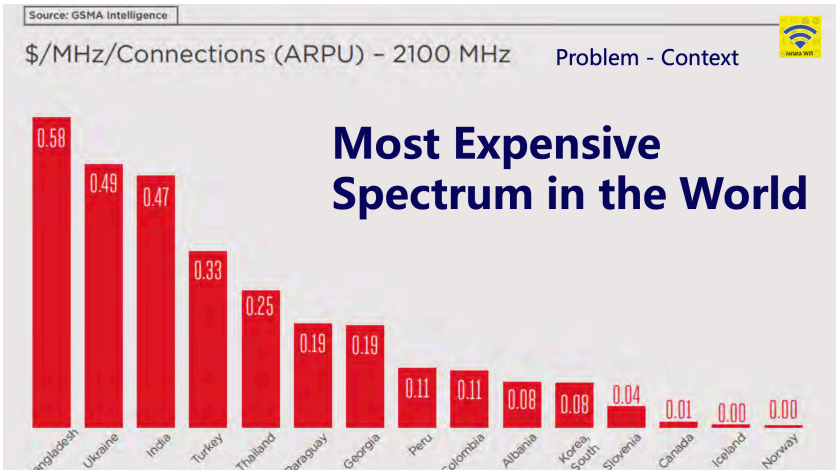

Figure 2 – 2.1Ghz Spectrum license costs, from Sheikh MD Seum, Janata Wifi

In India it appears that the network operators run their networks with slender margins and can convert one of the higher spectrum costs to a retail service that operates at a low retail price, that is affordable for many Indians. In Bangladesh there is a different solution that uses the fixed line infrastructure and WiFi-based access for data services and uses the mobile network for simple access token control and provisioning. In their case using tea stall locations there is no pre-existing dense existing WiFi deployments, so with little or no interference they can easily achieve 30Mbps – 40Mbps download speeds, compared to 10Mbps over 4G. And they are 10 times cheaper than 4G by using WiFi. In addition, the country has adopted a mobile money system called bKash that offers a low-cost secure monetary service using SMS/USSD services.

National Regulatory Practices in Internet Resources

One of the consequences in the domain name space with the two-letter country code top level domain names was that these names were readily swept up in various national regulations if the nation so wished. The same could not be said of the so-called generic top-level domains which operate within a framework of contractual obligations with ICANN with more nebulous enforcement capabilities.

What's the story with IP addresses? Are IP addresses that have been allocated or assigned to entities that operate with the context of a nation state the property of that nation state? For more than a decade the answer in South Korea is a clear assertion of national control. In Australia there is a similar capability, phrased as a reserve power that can be exercised by the government. In many other countries the situation is not so clear. A recent regulatory initiative has occurred in Russia that now places IP addresses, IP address and DNS names within the regulated space. There are reporting requirements for Autonomous System number holders, defined interactions with Law Enforcement Authorities and traffic localization. There is also a requirement to provide a local BGP from the AS, as well as a Netflow feed.

This might sound like a heavy-handed regulatory move on the part of the Russian authorities, but such detailed oversight at a technical level of public ISPs is not unheard of in other countries. For example, broadly similar measures have been in place in New Zealand for some years where notification of certain technical changes in the network's configuration to the regulatory body has been a requirement for public ISPs for some years, although they do not, as yet, require detailed usage data such as Netflow data sets.

Are such regulatory moves a brutal assault on the very concepts of the freedom of cyberspace? Or is this a more mundane case of restoration of more conventional forms of regulation and state oversight of public spaces and public activities? The Internet is not Facebook's private fiefdom and despite Facebook's apparent assumptions to the contrary, and they simply cannot do whatever comes to Mark Zuckerberg's mind each day without a due process of accountability to a broader public interest. Yes, this ill-defined balance between public interests and commercially driven imperatives can be, and has been, abused by private sector actors as much as it has been abused by state actors. I can understand the public sector's desire to regain some lost ground in the regulatory space, and this public sector desire is as apparent in the EU, the US and elsewhere as much as it is happening in Russia.

Some two decades ago, when the Clinton administration in the US was exerting significant international pressure to keep the Internet out of the domain of the ITU-T's administered international regulatory regime, the onus was placed on the public sector to engage in this environment on the terms defined by the Internet's actors. Following this US lead the public sector in many other countries duly followed along at the time and showed some hesitancy to pursue a larger agenda lest they scare off further private sector investment in the Internet and imperil the envisaged economic benefits that the Internet would surely bring.

We are now seeing a more confident and assertive public sector, perhaps in response to the largely unchecked rise of the global digital behemoths that are the other side of this conversation.

It seems now that this is just going to play out as a mix of international geopolitics clashing with massive business interests, and the best you and I can do is bring popcorn!

Route Flap Damping

Route Flap Damping (RFD) allows BGP speakers to "penalise" route that exhibit a large number of updates by accumulating a penalty count for each route and suppressing the route when the accumulated penalty exceeds a given suppression threshold. The penalty decays over time until it reaches a release threshold at which point the route is unsuppressed. This is a response to an earlier observation that much of the load issues in BGP were the result of processing these updates. It's had a chequered history and periodically goes in and out of favour. In 2006 RIPE-378 recommended that RFD not be used. In 2013 RIPE-580 revived it. In a recent re-evaluation of RFD the study presented at the meeting recommended that RFD settings remain as advocated in RIPE-580. However, it must be noted that BGP has always had a huge skew in updates, where a small number of prefixes (around 3%) are responsible for more than half the updates. Just 50 originating AS's account for slightly more than one quarter of all BGP updates. I've not seen a detailed analysis to determine if what we see today in BGP updates if the result of a sparse uptake of RFD, or if RFD is already widely deployed. I strongly suspect the former.

Route Refresh and RoV

In theory, BGP implementations need to keep a local copy of all the information received from each BGP neighbour in a structure called "Adj-RIB-In" (Adjacency Routing Information Base fo Incoming information). The motivation for this was to permit the local BGP instance to adjust to a change in the local best path selection policies that select entries from the Adj-RIB-In and load them into the local FIB (Forwarding Information Base). With a fully populated Adj-RIB-In these changes in policy could be readily accommodated and a new set of best paths could be selected (Figure 3).

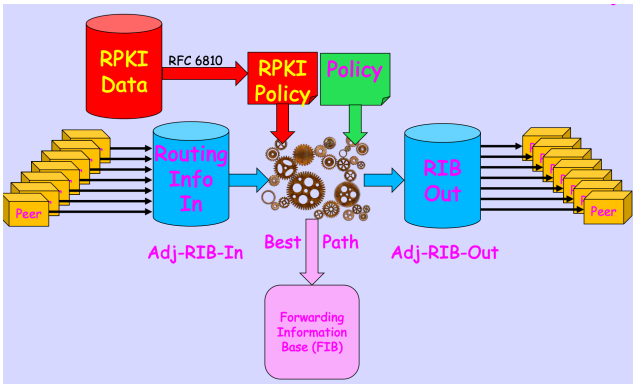

Figure 3 – Idealised Internal BGP data structures, from RPKI-Based Policy without Route Refresh, Randy Bush et al.

But if policy changes are infrequent this Adj-RIB-IN becomes a little used mechanism that consumes router memory and if you are a router vendor, it also adds to the cost of the unit. A way around this is to drop the fully populated Adj-RIB-In and perform an on-demand polling of all peers for their announced routes via a Route Refresh BGP command. It replaces a just-in-case provisioning policy with an on-demand policy. Which is a reasonable trade-off if local routing policy changes are infrequent (Figure 4).

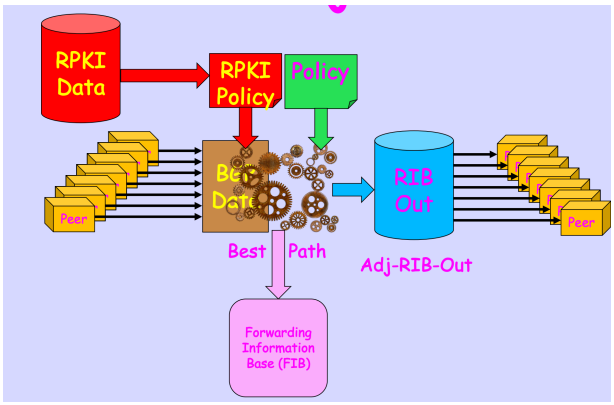

Figure 4 – Collapsed Adj-RIB-Ins, from RPKI-Based Policy without Route Refresh, Randy Bush et al.

But there were some wayward outcomes in IETF politics a little over a decade ago, when router vendors, notably Cisco and Juniper, were pressuring the IETF to stop making seemingly endless changes to BGP because that was hurting their business by inflating their product maintenance costs for their BGP implementations. The IETF was just finishing up a rather messy debate over "kitchen sink" BGP proposals where a large set of proposals were presented to the IETF to extend the use of BGP and its reliable flooding algorithm for all kinds of novel, imaginative and in some cases just plain silly, proposed uses. The instructions to the SIDR Working Group to develop specifications for adding secure mechanisms to BGP came with the stern admonition: "Don't, whatever you do, touch BGP. Just don't!"

This implied that the secure origination function was implemented outside of BGP, and the results of that function were passed to a router using an externally controlled route policy filter. Just how short-

sighted, silly and ultimately broken this non-protocol design constraint has proved to be for using RPKI to secure aspects of BGP operation, is properly a topic that deserves a more detailed analysis elsewhere, assuming that we will get around to performing a post-mortem autopsy on this work!

So, a change in the secure origination status of one or more routes becomes a change in the local policy set in the router, which implies that the router must sort through its Adj-RIB-In collection to find a new best path. Which is fine if you have it. And you will have an Adj-RIB-In data structure either because you BGP implementation included it by design, or if you configured some variant of the "soft inbound reconfiguration" configuration directive to your router, assuming it supports that directive. And if not, then you need to get a full route refresh from all of your BGP peers. For every Route Origination Validation change. Which is a very effective DDOS attack on your router, and possibly on your BGP neighbours as well. If your BGP-speaking equipment does not have this Adj-RIB-In data structure, then just turning on Route Origination Validation and dropping invalid routes might not be the best idea you've had today. Look before you leap and make sure that your configuration has full Adj-RIB-In support so that it can manage these routing policy changes without generating large volumes pf route refresh operations.

Now wouldn't it have been better if we have been permitted to integrate these security functions directly into BGP when we were embarking on the SIDR specification work in the IETF?

Measuring the Impact of a Route Hijack

I must admit that I am struggling with this presentation. It's a research project that is attempting to measure the impact of a routing hijack. The reason why I'm struggling with this is that the intention of the routing system is to propagate routing information to all parts of the network. If the hijack is intended to be generally effective, then the best way to achieve this is to use a more specific prefix. If the hijack uses the same prefix length as the target, in which case this is no different to the topic of anycast catchment areas. The problem with this latter issue is that the inter-AS topology of the Internet is not sparse and loosely connected but compact and densely interconnected. This means that anycast catchment areas have a large component of uncertainty where the AS path lengths of two of more competing advertisements have the same AS Path length, in which case the timing of the updates is a major factor.

IPv6 and ULAs

In the original concept of the IPv6 address architecture IPv6 addresses were meant to be so plentiful that every network, large and small, would be able to use a unique IPv6 prefix to address their network irrespective of whether they connecter to the public Internet or not. Now IPv6 did not have NATs (or at least the IETF was not prepared to countenance the use of NATs in IPv6 and refused to factor the use of NATS into the IPv6 architecture) so a potential consequence of using a fine-grained addressing architecture is that it would place an untenable level of stress on the routing system, as we did not device a new routing architecture for IPv6. It's just the same old routing tools with the same old scaling issues.

Perhaps instead we could make use of multi-addressing in IPv6, where a local network could use a local address prefix and a provider-based prefix at the same time. When a host was communicating with an external entity if could use a source address drawn from the provider prefix, and when it was communicating with a local entity if could use source address from the local address prefix.

The obvious question is: why bother with the local address prefix at all? Why not just use provider-based prefixes for everything? All this sounds fine if you are a provider and you are searching for hooks and barbs to make it difficult for your customers to desert you and head to another provider, but it doesn't look so good from the customer's perspective. And what should you do if you run a detached network? Yes, you could use any IPv6 prefix you want, but if you subsequently attach into the larger network there is a renumbering event looming in your future. You could head to a Regional Internet Registry but frankly they are not well suited to your requirements. They will lease you an address prefix for as long as you are

willing to pay a fee, and once you cease to pay the lease fee then a painful renumbering event is looming in your future.

Perhaps the IPv4 approach of a private-use address space and the definition of private-use AS numbers weren't so bad after all, and a comparable approach would assist IPv6 users. The original approach was the reservation of the prefix fec0::/10 for *site-local* addresses, which could be used for internal use within sites, and because it was internal there was no real issue with collisions in use across sites. It was all meant to be totally automated. Which was all well and good except that no one could figure out what a "site" was! The IETF backed off and deprecated site-local addresses. In their place they defined a more conventional local-use address prefix, fc00::/7 (RFC4193), called Unique Local Addresses (ULA).

Now the top half of this address prefix, fd00::/8, was defined such that anyone could simply use one or more /48 prefixes at random from this block and just use it. If you've been paying attention, then you would realise that such prefixes are simply not unique. If you use a decent random algorithm to generate a 40-bit number it is likely not to collide with any other random selection, but the probability of collision rises faster than the number of selections. So in RFC4193 they made provision to have a global registry in the lower half of this address prefix (fc00::/8) so that folk could avail themselves of a local use prefic that was assuredly unique. Except that they then didn't then complete that work. As RFC4193 states: "Another method, indicated by clearing the L bit, may be defined later."

Into this gap jumped SiXX, which ran a uniqueness registry for ULA prefixes. You generated a random 40-bit number and informed SiXX of the number and your details and they listed it in a public registry. For free. However, they stopped doing this in 2017. Now Ungleich in Switzerland has taken up this registry and is continuing the ULA prefix registry work.

I'm still not sure why. You can't route these prefixes and any collisions should be completely invisible in any case!

Other Topics

Over five days of meetings there were of course many other topics that were discussed at the RIPE 83 meeting, including the DNS, IoT, Open Source, Measurement, and various governance-related issues. All the material presented at the meeting can be found at https://ripe83.ripe.net/.

Disclaimer

The above views do not necessarily represent the views or positions of the Asia Pacific Network Information Centre.

Author

Geoff Huston AM, B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region.

www.potaroo.net