Geoff Huston
May 2017

RIPE 74

RIPE 74 was held in May in Budapest, and as usual it was a meeting that mixed a diverse set of conversations and topics into a very busy week. Here are my impressions of the meeting drawn from a number of presentations that I found to be of personal interest. All the presentations can be found at https://ripe74.ripe.net

The DNS

Rolling the Root Key of the DNS

It was more than seven years ago that many folk, including the RIPE community, were very clear in making the case that it was time to stop talking and get moving and sign the root zone of the DNS with DNSSEC. This work was duly completed on the 15th July 2010, and the root zone of the DNS was signed with a root zone Key-Signing Key.

No cryptographic key should live forever. We cannot foretell when and how techniques may appear that show vulnerabilities in our current cryptographic tools and algorithms, or when unforeseen events occur that either compromise the key value, or lock us out from further access to the key. So it's wise to plan ahead and develop operational procedures to roll the key to a new value, and exercise these procedures on a regular basis to ensure that we are all familiar with what is going on.

ICANN's Ed Lewis explained the process that is now underway, and highlighted the critical date, the 11th October 2017, when the current key will be swapped out in favour of the new key.

There are many moving parts to this story and some aspects of the risks involved in this key roll cannot be quantified in advance. If you run a DNS resolver that performs DNSSEC validation you should be aware of this process, and ensure that your DNS resolver configuration uses automated key management to follow the change of the root key.

Security and the DNS

The DNS is a remarkably chatty protocol, and perhaps we pay too little attention to the extent to which much of our online actions are signalled by the DNS queries that are made on our behalf. An analysis of the way in which The Onion Routing network (TOR) makes DNS queries reveals that the DNS requests use the same exit relay as the subsequent traffic. and this can be used to compromise the expectation of anonymity that is commonly associated with the use of TOR. Perhaps the point here is that the DNS itself is the point of weakness, rather than TOR, in which case this becomes a conversation about DNS privacy. How can we improve the privacy of the DNS and not make it an open channel to allow other to view our online activity. The presentation on the effect of the DNS on TOR's anonymity did not go into DNS privacy, but it is a subject of active study in DNS circles, and one well worth following.

A related presentation by Benno Overeinder of NLnet Labs provided a high level summary of the state of current DNS privacy efforts. This seems a little odd given that the DNS is in itself a public database and the information in the DNS is not an intrinsic secret per se. However, what is sensitive the queries

that you and I make of the DNS. Given that almost every Internet transaction starts with a call to the DNS, then the record of my DNS queries is tantamount to a record of my online activities. Can we make these transactions of queries a little harder to eavesdrop? One approach is to encrypt the traffic between the application and its local DNS library and the recursive resolver. This way the queries that overtly identify the end user are not readily accessible to third party snoopers. Of course, this does mean that you should exercise some care as to which recursive resolver you use, as this resolver will still be privy to all your DNS secrets.

The most popular approach to achieve this is to borrow a page from HTTPS book and use the TLS security protocol, and create a stack of using DNS queries over TLS over TCP. TCP Fast open and TLS session resumption make this look quite achievable and with pipelining and out of order processing this could be as fast as the current UDP-based query approach. while this appears to be a good fit for clients talking to recursive resolvers it is perhaps not so good a fit with recursive resolvers querying authoritative name servers. But it must be remembered that for as long as EDNS0 Client Subnet options are not being used, then the recursive resolver does not pass on the details of the end user to the authoritative name server.

However, the query name can be a useful data snippet, and the DNS behaviour of performing a top-down search for a name using the full query name at each level can be considered an unnecessary information leak. So qname minimisation is also a necessary tool for recursive resolvers to use, ensuring that the servers at the upper levels of the DNS hierarchy, including the much-studied root servers, are not given details of the complete query name, but only a truncated form of the name that allows it to respond with the desired name server records and no more.

It may not be a complete 'solution' to DNS privacy, but it is a large step forward for the DNS.

The Digital Object Architecture

Alain Durand of the ICANN Office of the CTO gave a very informative briefing on the state of another digital naming architecture, DOA. While the DNS is the dominant naming scheme for the Internet it by no means the only such scheme. First described in 1995 as the Handle System, it has been taken up by the publication and media industry as the Digital Object Identifier (DOI) scheme. This has subsequently assumed the name of the Digital Object Architecture (DOA), supported by a foundation, the Digital Object Network Architecture (DONA).

The DOA architecture is very simple: an identifier consists of a unique consist of a prefix which identifies a "naming authority" and a suffix which gives the "local name" of a resource, which is unique only within the context of the prefix. Naming authorities can have segments, delineated by a period. So "1" can be a naming authority, and "1.2" identifies the segment "2" within naming authority "1", and so on. The suffix is a string of characters drawn from the Unicode UCS-2 character set. The naming authority is separated from the suffix by a slash "/" character. Example DOI's include: 11738/ithi and 10.1038/nphys1170.

There are a number of aspect of this naming scheme that appear attractive: the naming authority label is usually a number, and the cachet of the DNS top level domain names is missing. This implies that a name authority is just that. It also appears that common use so far has continued with the local name of a resource. Common use interprets this local name as an index identifier. It is generally a "flat" space without named values, in a manner similar to the ISBN catalogue number for books.

The data associated with DOIs is not as limited as DNS Resource Records, and there are extensible indexed types that allow considerable freedom as to how to associate data with the DOI.

It is interesting to observe that many collections have used catalogue numbers, including books, periodicals, newspapers and similar. These collection indices are in many ways independent of the manner of their implementation, and are not intrinsically bound to the properties of certain digital network behaviours. Such systems appear to name the generic object, rather than identify a particular

instance of a digital object via its network location. Given the ubiquitous use of the DNS i's not surprising to see a number of DOI to DNS gateways that translate a DOI to a particular DNS name where an instance of the named object can be found. For example, the DOI 10.1038/ng571 can be accessed in DNS-based web browsers using the URL http://dx.doi.org/10.1038/ng571. Of course such mappings are not completely equivalent. The DOI effectively names the object, and could allow for numerous digital instances of the object to exist. The DNS effectively names the object via it's network location. If that location changes then the DNS label needs to reflect the new location, while the DOI name remains constant and the mapping information in the DOI resolution process is updated. If we really wanted this level of location independence in the DNS and use persistent names then we do this in the DNS through the use of NAPTR or SNAPTR records, where a generic, or permanent DNS name, is dynamically translated to a location-derived name. However, for some reason we do not use the DNS in this way, and today DOIs are used for some applications in some sectors and DNS names in others.

Routing

Studying Routing

The group at Roma Tre University has an outstanding track record of study into the Internet's routing system, and two lightning talks highlighted two aspects of this work. A study of BGP updates shows that BGP AS paths tend to shift within a bounded space, and most AS path changes shift between a small number of semi-stable states. One of these presentations took this to one level of finer granularity of detail, and looked at the hop-by-hop paths as shown by the RIPE Atlas traceroute repository. The study searched for periodicity in traceroute-reported paths, and a similar component of periodic behaviour. In some ways, this is not a surprising result, in that the topology of the Internet is highly constrained and the available states between two points are limited.

The second presentation from this looked at the issue of partial visibility of prefixes in the inter domain routing system. The observation motivating this presentation is that routing issues are often piecemeal. In other words, the routing issue may be that a route to a particular address prefix is not propagated to all parts of the Internet, so only a subset of the Internet can reach a given address. The tool described in this presentation, stream graph, attempts to summarise this partial visibility, using the vantage points of the peers of the Routing Information Service (RIS) to generate the base reachability information.

RPKI Filtering

These days it seems popular to measure the uptake of the use of ROAs to inform route selection. One such approach was explained by Andreas Reuter. One potential approach is to search for pairs of announcements from the same origin AS where one announcement can be validated by a published ROA and the other is either not validated or invalid. If the AS paths of the two announcements differ, then there is a potential inference that there is some form of ROA filtering and the candidate AS performing the filtering is announced as part of the AS Path in the valid announcement but not in the other. As the presentation explains this is not a reliable indicator. There are many reasons why AS Paths differ in the Internet, including various forms of traffic engineering and policy controls, so drawing the conclusion that such AS Path differences are the result of invalid ROA filtering is prone to error.

An alternative approach is conduct active experiments by deliberating injecting valid and invalid route advertisements into BGP. In this case they extended the experiment further by periodically flipping the validity state of one of the announcements between invalid and valid. Their hypothesis is that if there are AS's that implement filtering of invalid routes, the flipping of the ROA state should cause the AS path to also change at that time. This technique found 3 ASs that appear to actively filter invalid route advertisements. Obviously this is a rather underwhelming outcome at this point in time.

IPv6

There are always a number of interesting IPv6 presentations at RIPE meetings, and this meeting was no exception.

IPv6 Deployment Stories

Martin Levy from Cloudflare reported on their IPv6 efforts. They are now a considerable presence in the CDN space, with over a hundred data centres, all dual stack, and some 6 million customers. In recent months, they have been enabling all the Cloudflare cached sites to serve data on IPv6, which has made a visible impact on the metrics of the number of web sites reachable via IPv6. They are also proposing in the IETF to change the response to a query for an IPv4 address to add a IPv6 address to the response, saving the additional time to make a seance query for the increasing number of dual stack end users.

Rabobank reported on their decision process for adopting IPv6. While they did not directly feel the pressure to adopt IPv6 for their online services for themselves, they felt that by staying with a IPv4-only service they were seeing more of their customers using CGNs to reach them. This caused some issues for their security operations centre, where blocking an IP address from access to a service may entangle other customers, and also raised problems with tracking of the effects of phishing and other malware. By deploying IPv6 then they are in a position to respond using IPv6 as customers are are provided with IPv6 by their network service providers.

While an IPv6 deployment addresses some of the issues with end user attribution in an environment of shared IPv4 address, there are other security issues that are introduced with IPv6.

Challenges with IPv6 Security

Many presentations at meetings such as these tend to repeat common values. However, from time to time some presentations deliberately challenge those values and confront our common preconceptions. Enno Ray's presentation on IPv6 Security. He starts by observing that "Many [of the topics he discusses] go back to decisions (or lack thereof in the relevant IPv6 WGs at the IETF. We've all heard about the creep of self-interest, politics, etc, into voluntary organisations, which effectively undermine their original purpose. I think this point has long been reached in certain IETF circles, namely in 6man." Strong stuff!

He criticises the unclear relationship between SLAAC and DHCPv6, the relationship between ND (neighbour discovery) and MLD (multicast listener discovery), the interaction between RA flags, routing tables and address selector mechanisms, and MTU issues.

In retrospect it is challenging to justify the shift from ARP to multicast-based neighbour discovery. If the number and size of RFCs that describe and subsequently attempt to clarify a technology , then by that metric ARP compares very favourably with the 94 pages of RFC4861 and the subsequent 6 clarifying RFCs!

The concept of multiple IP addresses and address scope (think "link local") has been a sources of considerable confusion and ambiguity and appears to serve no particular useful function. It increases the decision structure for local hosts and increases the opportunity for code to make mistakes. Yet again, several RFCs attempt to clarify IPv6 address scoping and address selection.

The use of Extension Headers is similarly criticised. The concept of an 'extensible' protocol certainly appears attractive as an attempt allow for future evolution to be accommodated in the protocol, but it comes at considerable cost. It adds complexity, increases code decisions and probably also increases variability. Extension headers in an IPv6 datagram come with variable types, varying sizes, variable order, variable number of occurrences of each type, and varying field values. How is a protocol implementation meant to cope with this level of variability? Jon Postel is quoted from RFC761 to "be conservative in what you do be liberal in what you accept from others" in the Robustness Principle.

Eric Allman published a "reconsideration" of this principle in 2011 (http://queue.acm.org/detail.cfm?id=1999945). He postulates that "Perhaps it is more robust in the long run to be conservative in what you generate and even more conservative in what you accept.", and concludes that "And like everything else, the Robustness Principle must be applied in moderation." Extension Headers are central to this issue. Should an IPv6 implementation pass on packets with unknown extension headers? Or drop them on the basis that it is better not to pass on a packet whose entire header is not clearly understood by the code? The ambitious treatment of IPv6 packets with Extension headers would not be such a fraught problem were it not for the decision to remove packet fragment control from the IP packet and put it into an extension header!

IPv6 also changes the role of a router. Not only is it a gateway to remote networks but it is now an intrinsic part of the network provisioning architecture. These days there are all kinds of local networks, and on public systems we should not simply trust a device that purports to be a local router. This is of course unwise, and we now have RA GUARD to identify and defend us against such rogue elements in a network. But, as pointed out in RFC 6890 the interactions between packet fragmentation and RA packets present continuing vulnerabilities in this area.

A more general observation that Enno makes is that a protocol that continues to evolve leaves a trail of deployed systems that implement older versions of the protocol. Whether it's the way in which local interface identifiers are generated on Ipv6 hosts, neighbour discovery, or address selection for outbound packets, IPv6 has specified different actions at different points along its evolutionary path. The result is that Ipv6 is considerably more complex that Ipv4, and this complexity is expressed both in the code to implement the protocol and the effort required to operate robust production networks that support Ipv6.

IPv6 Addressing

In some ways "too much" appears to be just as big a challenge as "not enough". Prefix lengths appears to be a continuing debate and the presentation on the work on operational practices for IPv6 prefix assignment shows that we are by no means agreed on what are the best practices for IPv6 address plans. For example, some years back, the IETF decided that some form of local address prefix for use outside the context of the public Internet was desirable, and the concept of "ULAs" or Unique Local Address prefixes was set up. This is a pool of self-assignable IPv6 address prefixes that operate in a manner analogous to private address space in IPv4. These days, according to this work, the use of such ULAs is "strongly discouraged". What about a point-to-point link? In IPv4 we often use a /30, allowing for each end to be addressed, and a sub-link broadcast address to address both ends at once. Well IPv6 has done away with broadcast addresses so why not use a /127 for such links? Again, these days, the fashion is changed to recommend the use of a /64 for such links. And, despite some considerable debate from time to time over the consumption implications, the current advice is to assign a /48 to "everybody". I guess that includes my mobile phone!

I guess this is another aspect of the evolutionary nature of IPv6 where we keep on making changes to aspects of the protocol over time. The problem here is that we leave in our wake the accumulated legacy of all the previous decisions, so that the situation at any point is perhaps more chaotic than it would've been had we not attempted to impose such rules in the first place! The best advice one can offer to implementors of IPv6 software and hardware is to assume absolutely nothing about prefix sizes.

To 64NAT or 464XLAT?

It is perhaps unfortunate that we provided choices in the IPv6 transition environment. The outcome of providing choices is invariably some deployments make one choice and others make the different choice! It was in the mobile sector with the deployment of IPv6. Many large network providers reached the point that they could no longer offer universal dual stack service on all of their mobile service platforms due to a lack of further IPv4 addresses. The obvious answer was that they needed to operate

their network as an IPv6 network, but the challenge was to continue to provide a working IPv4 service to the attached devices.

One model is effectively a tunnelled NAT, where the device has a locally addressed IPv4 environment but the device provides a virtual IPv4 interface that performs an encapsulation into IPv6 to pass the packet to the mobile carriage network. The mobile operator passes these packets to a decapsulating CGN and native IPv4 packets are passed out to the Internet. The device believes it is a conventional dual-stack host and no changes are required to applications. Android supports 464XLAT.

Apple's iOS does not. It uses DNS64. Here, the device is connected to a service that is only IPv6, so it thinks it is an IPv6-only device. Which will be great in some future world where the Internet only supports Ipv6, but right now there are a whole lot of services and users that don't. So the way they fake it is to call on the DNS for assistance. Whenever the device queries for a service name that has no IPv6 address then it generates an IPv6 address that points to the operator's 6to4 protocol translator with the IPv4 address encoded into the address. When the device sends an IPv6 packet to this protocol translator it essentially perform a a header transform to IPv4 and also uses a NAT-style operation to generate the public IPv4 address. This is nowhere as straightforward as 464XLAT, as the presentation on nat64check related. A major part of the issue here is that we are sending out mixed messages about the DNS. On the one hand we are pushing the message that DNS needs security and privacy, and we are providing tools and solutions to address this. On the other hand DNS64 relies on lies in the DNS and also relies on passing all client DNS queries to the operator's DNS resolver, which may have significant privacy implications. Sometimes choice is not the best outcome.

Other Topics

Quantum Networking

One of the more memorable presentations at RIPE 74 was Stephanie Wehner's presentation on Quantum Networking. Quantum entanglement is a physical phenomenon such that for two entangled particles, when the state of one particle is revealed the state of the other is determined at exactly the same time. Quantum entanglement effects have been demonstrated experimentally with photons, neutrinos, electrons, molecules the size of buckyballs, and even small diamonds. One of the applications for quantum entanglement is key distribution for encryption algorithms, as the entanglement properties are essentially private and cannot be tampered with. The work remains very much in the area of high frontier research and the quantum devices are still of a size of 6 qubits, so there is much to do here. However, there are some fascinating possibilities to come when looking at "classical" computer science problems and applying quantum computing techniques. But that's a topic for post-quantum cryptography, at another time.

Fast

There is a certain level of obsession about speed in the Internet and measuring it. Speedtest.net is a tried and true favourite for many, but now there is also Netflix' Fast (fast.com). The goal os fast was simplicity of use and reliability of the reported speeds. The objective here is to measure the last mile of Internet access, and not to have this measurement distorted by long haul paths across the transit of the network, so fast uses a distribution of so-called Open Connect Appliances as the target for these speed tests.

The fast.com client code downloads a list of appliances, and makes a choice as to what it believes is the 'best' appliance to use for the test. It then negotiates with the appliance to undertake a number of download tests of varying sizes. The TCP ramp-up time is not included, but when the unit believes that TCP is operating in some "steady state" mode, it reports on the average speed it can achieve.

The only aspect about this that I'm uncertain about is the exact nature of the TCP flow control protocol used to undertake these tests. It is evident that a TCP speed test can produce a wide variety of answers depending on the level of cross traffic and the interaction of the flow control algorithms.

Different flow control algorithms can exert different levels of flow pressure on concurrent flows in order to claim some notion of its "fair share", so in some ways there is no single notion of speed in a shared network. But maybe I'm quibbling at this point. It is indeed a simple test that appears to work well!

A Saga of Crowdfunding

The folk at cz.nic seem to be leading lights in the area of open source projects. They are responsible for the Knot DNS server, the Knot DNS resolver, the Bird implementation of BGP and the open source access router, Turris.

Turris was a grand attempt to respond to the overwhelmingly crappy world of computer premises equipment by producing a home gateway router that did it properly. IPv6, DNSSEC support, good security, and a means of automated updates, and extensible. The first project for Czech users was so successful that they succumbed to temptation to do it on a bigger scale using crowd-source funding via IndieGoGo. The conservative target was $100,000 - which was reached in 21 hours! The total project raised some $1.1M in funding for some 4,400 routers. However, they started adding options with the "stretch goals" with colours, WiFi, memory, power sources, and when you added t-Shirt sizes there were some 32,000 possible combinations within the orders! The production encountered some false starts with PCB manufacturers, so the project slipped as they moved to production.

But while they are a little more wary of venturing into hardware a second time, the light still shines brightly for open source software projects at CZ.NIC!

Caught Between Security and Time Pressure

There is are a lot of presentations at these meetings and a wide variety of presentation formats. For me the most memorable presentation was by Constant Dietrich of the Beuth Hochschule für Technik & Technische Universität Berlin. The topic was system misconfigurations and the security implications that may result. The message was to recognise that such misconfigurations do happen, and a clearer understanding of the causes may help us avoid them more effectively. What really pushed this message home was the brilliant slide presentation format she used to convey the material. It's well worth a look at https://ripe74.ripe.net/wp-content/uploads/presentations/53-LATEST-RIPE-Misconfiguration-Slides.pdf

Author

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

www.potaroo.net

Disclaimer

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