The ISP Column
A monthly column on things Internet

Geoff Huston
April 2016

DNS OARC 24

DNS OARC held a two day workshop in Buenos Aires prior to IETF 95 at the end of March 2016. Here are my impressions of this meeting.

For a supposedly simply query response protocol that maps names to IP addresses there a huge amount going on under the hood with the DNS. Some good, some not so good.

In the not so good category Anand Buddhdev from the RIPE NCC reported on some attacks against the authoritative nameserver infrastructure operated by the RIPE NCC. They operate an anycast cluster with 9 servers in 3 locations at AMSIX (Amsterdam), LINX (London) and NetNOD (Stockholm). They serve some 5,000 zones, with a mix of reverse address-to-name servers and secondary services for a number of other domains, including a number of country domains. They experienced a high query load targetting the Turkey (.tr) secondary nameserver. The initial attempts to filter this traffic lead to changes in the attack profile, and the response then lead to a kernel filter to perform incoming packet drop based on query patterns. This lead to a response of more intense incoming traffic volumes that threatened to overwhelm the ingress paths to the servers, leading to a distributed routing blackhole response. This is another illustration of the acute level of toxicity in today's Internet which now have to capacity to overwhelm all but the very largest of the service providers. The RIPE NCC is looking to take a path of outsourcing its DNS services into facilities that can absorb multi-gigabit traffic attacks. The attacks are temporal, but the ensuring defensive measures are expensive, and the limited capacity to deploy such services tend to concentrate service delivery into a small number or providers, while the rest of the Internet experiences ever rising toxic levels of background radiation through these attacks.

Also in the not so good category, Matt Weinberg and Duane Wessels from Verisign reported on A root and J root traffic, and in particular attacks against the root in November 2015 and December 2015. These attacks generated a lot of discussion including the tech press at the time. There were two attack events, on November 30 and December 1, and 10 out of 13 members of the root server constellations received the attack traffic. D, L and M received no attack traffic, and it appears that the traffic was being sent to the old IP addresses for these servers). The traffic was UDP only and IPv4 only. They saw the command and control traffic as well as the attack traffic. All the queries were for `www.336901.com` (evidently a Chinese game site) with 5.2M query per second (qp)s at A root and J root. The second attack was 24 hours later for `www.916yy.com` at the same 5M qps query rates. This used 895 M unique source IPs (obviously spoofed), but 68% of the queries were from 200 IP addresses - i.e. there was a strong core of attack query traffic from a small set of attackers. This attack caused service impact on the 10 root server clusters, as seen by RIPEmon monitors. The attack traffic was evenly spread across the major A-root server sites. J root use 90 smaller sites and packet loss was seen at some of these smaller sites. Verisign use Response Rate Limiting (RRL) and this dropped up to 60% of the attack traffic automatically (corresponding to the clustered attack traffic). However, this does not stop all the traffic and regional sites with low query volumes did not trip the RRL threshold. The presentation included an effective graphic using Hilbert curve display of IPv4 address space to show the systematic way in which source addresses were spoofed. The clear evidence of a number of distinct source address "walkers" was seen as evidence of a number of distinct attack generation engines at work. This is known malware used for generating DNS attacks. The underlying motivation od the attack is not clearly understood. This damage was to the root systems, so the query target sites were not harmed. It is possible that this was flag planting or possibly a diversionary exercise. In this case RRL responded well given the reuse of certain IP addresses. One interesting area of speculation is the response of CGNs when a source address of a packet is spoofed. The data tends to suggest that a widespread CGN behaviour is to perform the NAT mapping irrespective of whether the source address is spoofed or not!

There is a second form of unwanted DNS query traffic, but this query traffic is not malicious. This is the detritus of the DNS itself. When query is passed into the DNS system for resolution it does not go completely unnoticed. I reported to the workshop on an analysis of DNS queries seen as part of an unrelated measurement exercise. Out of 44 billion DNS queries seen by the measurement's authoritative name servers in a 3 month period, some 25%, or 11 billion queries, were 'junk' queries that reused old experiment names. Further analysis of these zombie queries revealed one group of behaviours linked to be DNS recursive resolvers performing some form of local cache refreshing operating over long times. A distinct query behaviour looks to be DNS stalkers that replay previous DNS queries. And there is a small group of of totally deranged resolvers that appear to have some form of very tight loop of repeat queries!

A second theme of the workshop was on key and key generation algorithm changes for DNSSEC.

This is a topic with some level of background. The first time this gained wider attention was some years ago when each time a key rolled the query levels to the authoritative name servers increased (http://www.potaroo.net/ispcol/2010-02/rollover.html). Since then resolver behaviour has been moderated and we don't observe that particular broken behaviour any more, but that does not mean that the process is now completely trouble free. There are two topic of interest at this OARC workshop: change of algorithm and change of the Root Zone keys.

Annand Buddhdev of the RIPE NCC also spoke on the subject of crypto algorithm rolls, looking at at the experience of the RIPE NCC in rolling the algorithm of their signed zones. The RIPE NCC was an early adopter of DNSSEC, and some 10 years ago the use of SHA-1 was a commendable decision: it was probably the strongest algorithm available at the time. These days its a point of vulnerability, so the decision to change to use SHA-2 was entirely reasonable. However, they use Secure-64 units to manage their signed zones, and these units were incapable to handling an algorithm change. Their initial workaround plan, to go from signed with SHA-1 to insecure to signed with SHA-2 was not well received by their community, so they had to wait for algorithm roll support from Secure 64 to implement this capability. The plan was to introduce the SHA2 RRSIGS, wait for twice the TTL, then introduce the new ZSK and KSK values. again wait for twice the TTL then withdraw the old ZSK , and following a DS update at the parent, to withdraw the KSK. The encountered problems with the ZSK withdrawal. Section 5.10 of RFC 6840 requires that all the algorithms present in the DS and DNSKEY records be used to sign the entirety of the signed records in the zone. This was being strictly enforced by both Unbound and the Versign resolvers which caused their problems. They were forced to re-instate the ZSK, and reverse the planned order, withdrawing the old KSK first and only after the DS is updated, withdraw the old ZSK.

I reported on an update of the level of support for the ECDSA P-256 in DNSSEC validating resolvers. When this measurement was first undertaken over a small sample set of some 3 million user queries in 2014, some 1 in 3 of users were behind validating resolvers that recognized the RSA algorithm did not recognize the ECDSA algorithm. This experiment was repeated in early 2016, this time with a sample size of some 765 million sample queries. The results show that some 83% of all the queries that clearly showed the use of RSA-validating resolvers also clearly showed the recognition of the ECDSA algorithm, while the other 17% showed that the resolver was not recognizing ECDSA as an algorithm it could cope with. This is a failure rate of 1 in 6, which represents a significant improvement over the 2014 figures. It was possible to identify common resolvers who fell into this failing category, and it was noted that there was a large representation of mobile phone data service providers in this set of failing resolvers. The other rather attractive aspect of this protocol change is that validating resolvers that do not recognize the signing protocol fail by regarding the data as unsigned. This means that resolvers that fail to recognize the algorithm at least still function as resolvers!

There was a very interesting panel discussion about DNSSEC algorithm agility. It's probably a discussion that used to be held within circles of cryptographers, but these days the DNS is a large scale user of crypto technology, and the algorithms that are being used are important. The widespread reliance on prime number cryptography with the RSA algorithms is not a good enough long term strategy these days, if it ever was. RSA has a relatively poor cryptographic efficiency, requiring relatively large key sizes to provide security that provides adequate protection against cracing efforts using current technologies. Alternative approaches exist, including the currently promising areas of elliptical curve cryptography. But how can these more recent cryptographic algorithms be introduced into mass use? We already see the slow and drawn out process to introduce support for ECC-P256. What chance do we have of being able to completely withdraw from an algorithm, or do we have to carry support for all these algorithms indefinitely in all validating resolvers? At the same time the barriers to adoption of new algorithms is certainly quite substantial. Certification of a crypto software library can be a painfully slow process, and the process of upgrading deployed software is incredibly painful. There is a long tail of old versions of software in active use. Managed devices such as personal computers fare much better in this respect than completely unattended or embedded devices, where if there is an in situ update process its likely to be exercised highly erratically if at all. We are leaving in our wake a trail of what we would like to think are other people's legacy compatibility problems. The DNS, as it stands today, is caught up in this. The protocol expects signed material to head out into caches and other forms of intermediaries, so the concept of a capability negotiation between the zone publisher and a resolver is somewhat of a far fetched notion in today's DNSSEC environment. So it seems that we are caught up in a process that exhibits a high level of inertia. There are many barriers to both adopting new crypto approaches and similarly many barriers that prevent us walking away from old and potentially outdated approaches.

Duane Wessels of Verisign reported on current plans to change the size of the Zone Signing Key (ZSK) of the Root Zone. The plan is to increase the size of the ZSK to 2048 bits. Their plan is to pre-publish the longer key in the DNS Root Zone commencing 21 September 2016. On 1 October the zone will be signed by this longer ZSK, while the old 1024 bit ZSK will be still published in the root zone for a further 30 days. This is 20 days longer than the usual 10 day post-publish period. The Key Signing Key (KSK) of the root zone is already a 2048 bit RSA key, so there are no substantive concerns over resolvers' acceptance of a 2048-bit key. However, the larger key will increase the size of DNS responses for the root zone's DNSKEY record by 128 octets. The standard size response will increase from 736 octets to 864 octets. During the regular ZSK key roll there will be 2 of these 2048-bit ZSKs, which will entail a 256 octet increase from today's 883 octet response to 1,139 octets for the ZSK roll. This may encounter some small levels of trouble in some cases with some validating resolvers indicating that they are unable to handle a 1,139 octet UDP response (through their offer of a smaller buffer size in the EDNS0 buffer size in their queries) and at the same time are unable to establish a TCP connection.

The meeting also heard a report on the plans to roll the Root Zone's Key Signing Key (KSK). A report from an ICANN-commissioned Design Team was released at the start of March. This report calls for a 3-month process of introducing the new KSK into the Root Zone, followed by a cutover to the new KSK at the start of the second 3-month period, and the revocation of the old KSK in the last 3-month period. The report called for the preparation of the new key material in 2016 and the roll of the KSK in the first 9 months of 2017. It seems likely that the roll of the ZSK, currently planned for 2016, may push back the KSK roll activities, and the timetable proposed in the Design Team's report may be pushed back by 2 or 3 quarters so that the KSK key roll may well occur at the end of 2017 and moving in to early 2018. There are two major considerations with respect to this change in the key. The first is that there are transitory periods when the response to a DNSKEY query to the root zone will be larger. The final phase of pre-announcement will see a response size that is 1,414 octets, as it will contain two of the 2048-bit ZSKs and two of the 2048-bit KSKs, signed by a single KSK. The second critical phase is the period of revocation of the old KSK. This will contain one ZSK, two KSKs and two KSK signatures. The response size of this packet will be 1,425 octets. It is anticipated that in both cases a certain number of DNSSEC-validating resolvers will have problems in receiving this response due to the response size. The second problem is the issue of manual configuration of the trust key set. We cannot tell in advance of the key roll how many resolvers have set themselves with a manually configured trust anchor key. Or, perhaps a little more concerning, we don't know how many secure DNS validation packages that are packaged with manually set trust anchor key values have been distributed. In this case its not clear that the system administrator is even aware of the package setting of manual keys. These resolvers will be caught up by a key roll of the KSK, and at the point of the roll will no longer be able to resolve any name, whether signed or unsigned. While it is possible to estimate in advance the number of validating resolvers who encounter problems with DNS responses of 1,425 octets, and even estimate the number of users behind these resolvers, it is simply impossible to perform any advance measurement of resolvers with manually set trust anchors. Unfortunately, we will only know the extent of the issue when the key rolls, and not before.

There are many other aspects of evolution of the DNS, and some of these were evident at this workshop.

The world of resolvers has expanded from the BIND software to include a number of products, including unbound, Power DNS and others. The most recent entrant in this space is the Knot DNS Resolver, from the CZNIC folk. Ondrej Surej of CZNIC talked of the developments with their Knot DNS resolver project (https://www.knot-resolver.cz). It is an open source platform for a recursive DNS resolver, written in C and LuaJIT. The code is constructed on top of the Knot DNS libraries. Knot DNS allows flexible programming constructs in the configuration files. It implements IPv6 "happy eyeballs" with a 20ms head start bias towards making DNS queries over IPv6 transport. It is a general purpose resolver with intended application at both large and small scale. It has flexible shared cache back ends that allows multiple daemons to run from a common cache. It supports RFC7646 negative trust anchors, and can perform prefetching. It performs QNAME minimization. It supports persistent caching and a Tinyweb module. It's really good to see the ecosystem of the DNS branch out and have some real diversity and choice these days in the resolvers we use.

The conventional model of DNS resolution is that this is a task that is outsourced to dedicated servers. But what if an application wants to use DNSSEC directly, and is not willing to trust intermediaries, even the platform on which the application is running? Sarah Dickenson and Willem Toorop reported on the getdns api work (https://getdnsapi.net/) This is an api designed by applications developers as a natural evolution from the getaddrinfo() api. getdns is an asynchronous interface into the DNS so that the local library can operate either as a stub resolver or a full recursive resolver (using the unbound resolver code). The intention is to allow applications to have direct visibility into DNS behaviour and DNSSEC validation, so that there is a better potential to integrate services such as DANE into application behaviours. It supports DNSSEC validation in both stub and resolver modes, and the recent work on DNS over TLS as part of the DNS Privacy efforts. The api is asynchronous by default, organised around events (libevent, libev and libuv). It uses a JSON dictionary model of responses and response types. They are seeking tighter integration into OS distributions and applications.

There were many other interesting presentations in a very packed two day workshop. The best I can do is to point the reader to the workshop web site: https://indico.dns-oarc.net/event/22/ where all of the workshop presentations can be retrieved.

Author

Geoff Huston B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for building the Internet within the Australian academic and research sector in the early 1990's. He is author of a number of Internet-related books, and was a member of the Internet Architecture Board from 1999 until 2005, and served on the Board of Trustees of the Internet Society from 1992 until 2001 and chaired a number of IETF Working Groups. He has worked as an Internet researcher, as an ISP systems architect and a network operator at various times.

www.potaroo.net

Disclaimer

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