The ISP Column
A monthly column on things Internet

May 2019
Geoff Huston

Report: DNS OARC 30 Meeting

DNS OARC held its 30th meeting in Bangkok on the 12th and 13th May. Here's what attracted my interest from two full days of DNS presentations and conversations, together with a summary of the other material that was presented at this workshop.

Some Bad News for DANE (and DNSSEC)

For many years the Domain Name X509 certification system, or WebPKI, has been the weak point of Internet security. By "weak point" you could as easily substitute "festering, rancid, underbelly" and you would still be pretty much right on the mark! The massively distributed trust system has proved to be unmanageable in terms of integrity and there is a regular flow of stories of falsely issued certificates which have been used to perform intrusion attacks, eavesdrop on users, corrupt data and many other forms of malicious behaviours.

The efforts of the CAB Forum [https://cabforum.org] to instill some level of additional trust in the system appear to be about as effective as sticking one's fingers into a leaking dyke. The number of trusted CAs has extended conventional credibility well beyond the normal boundaries and has pushed the unsuspecting user into a fragile state of credulity. Efforts to improve this mess, such as Extended Validation (EV) certificates, have gained no traction with users, as they are largely immune to subtle changes in the content and colors of the browser's navigation bars, and certificate transparency logs appear to be completely, ineffectual of catching CA-related name hijack events in real time.

The effort to define DANE, or Domain Keys in the DNS, was an effort to provide a different mechanism of name-based assurance, by using the DNS to convey credentials to the user rather than a third party-operated X.509 PKI infrastructure. DNSSEC provided a way to allow any entity to directly assure itself that the response it had received from the DNS, relating to a record held in the DNS, was indeed precisely that DNS record at that time. If the entire objective of the Web PKI and all these domain name certificate issuers was, in the end, to associate the control of a key pair with the control of a delegated domain name in the DNS, then DANE would cut out this morass of intermediaries and allow the domain holder to store the name operator's public key in the DNS in a manner that would be hard for attackers to corrupt.

Shumon Huque is of the view that DANE was the reason why DNSSEC was worth the effort (and I agree with him!). This was the way to finally bring some robust security into the use of the name system and allow applications to ensure themselves that they are indeed connecting to the genuine named service.

A basic sticking point for browsers has been the extended time taken to perform DNSSEC validation. There was a concerted effort to address this through a mechanism called "DNSSEC Chain Extension" [https://tools.ietf.org/html/draft-ietf-tls-dnssec-chain-extension-07] that was to be "stapled" to the TLS material in the credential exchange. The document describing this approach was initially approved in the IETF's TLS Working Group as a candidate standard track document in March 2018, and an implementation was funded and planned for Mozilla. But this effort crumbled in what was described by Shumon as a "huge fight" in the working group, and the draft was abandoned. The result is that DANE is effectively dead for browsers for the time being.

It is incredibly frustrating to see these developments. The intentions behind both DANE and free domain name certificates were laudable, as affordable high-quality security for all was the intended result. But what we are left with is no better than before, and possibly worse. Truly reliable robust security is even more of a luxury good than ever.

The Modality of Mortality of Domain Names

Are all new domain name registrations basically junk? Do they live Hobbesian lives that are "nasty, brutish and short"? How are these short-lived names destroyed? Farsight's Paul Vixie reported on a study of the mortality of domain names that exist in the DNS for less than one week.

The study observed a creation rate of around 2 per second, or 150,000 new domain delegations per day and the creation of new host names at a far higher rate of some 300 per second, o3r some 12 million names per day. They took a six-month window and studied some 23.8M newly delegated domain names. a little under 10% of these names had died within 7 days. And of these, most die within the first 5 hours, and 60% of these short-lived delegations die within 24 hours.

The major cause of this early demise is blacklisting of the domain name. Blacklisting is a very rapid response, with some 80% of blacklisted domains entered into the lists within 24 hours of the time of first use of the name. More than 30% of blacklisting occurs within 1 hour. A second cause is removal of the delegation record. This form of name removal al takes linger, with a median of some 2 days. Only 20% of the names that are removed in this way are removed in under 1 day. Another cause is the removal of the name's authoritative name servers, and here while one quarter of the name removal events occur within 1 day, the median time of death by this cause is some 4 days. The longer time here may be an artefact of credit card transaction clawback or similar.

The majority of the short-lived names were observed in the gTLD space, and here blacklisting is the primary cause of name death. This was also observed in those ccTLDs that are used as generic TLDs. Overall, some 8% of new names die within 7 days.

The observation from this study is that we appear to be spending a huge set of resources to remove names that should never have existed in the first place. If further rounds of new gTLD rounds turn out to be little more than an exercise to offer more choices for spammers, then why are we doing this to ourselves?

Hyper-Hyper-Local Roots

RFC7706 describes how a recursive resolver can configure a local copy of the root zone and use this local copy as a fast alternative to performing queries directed to a root server.

Ray Bellis described the RFC 7706 approach be being too prescriptive. There is no need to put the root into every recursive resolver, and if a network operator wanted to go down this path a local root resolver should be capable of supporting many recursive resolver clients. To illustrate this, Ray used the ldns DNS library to implement a fast root server in a tiny hardware platform. He used pre-compiled answers by generating pre-computed compression offsets. He uses raw sockets and "stateless" TCP [http://www.potaroo.net/ispcol/2009-11/stateless.html] to speed up the server's TCP performance. It's blindingly fast on small processors, and Ray achieved 15,000 queries per second on a Raspberry Pie 3B. It has a very economical 13Mb ram footprint.

More generally, it's possible to generalize this approach and take relatively small zones and use this technique to tune them to offer very high-performance DNS servers on extremely small devices.

Deploying Authoritative Servers

What's the best way to set up a zone's authoritative name servers? Is many better than just 1 or 2? Is anycast useful for authoritative name servers? Is the design of the root zone server infrastructure with 13

named servers with associated anycast services something that we should all copy, or should something less ambitious be entirely adequate for the job? In many ways the design of an authoritative server system represents the outcome of balancing several factors. There is a consideration of server availability, server performance for both positive and negative answers, and the behavior of recursive name servers.

The IETF's standards point to a strong preference for zones to have at least two authoritative name servers, and preferably disperse them so that they do not fate share, and justify this preference as being robust in the face of individual failures. As a result, many zones including those considered critical to many enterprises operate with a large number of NS records per zone.

If a zone is served by a number of name servers in the form of multiple NS records, how do recursive resolvers choose a name server to query? There is a widely held belief that a recursive resolver will regularly sample the time to query each authoritative name server and then use the fastest server for the next sample period. Work by Akamai's Kyle Schomp looking at queries to an Akamai zone largely bears this out, but with a few important caveats. The issue is that the concentration of use of resolvers is highly skewed, and while a small subset of these resolvers perform a high volume of queries that allows them to cache responsiveness per zone per server, the rest have a far lower overall query volume and the server selection algorithm gives inconsistent results in such circumstances.

If you thought that many distributed authoritative name servers for a zone gives faster overall name resolution performance, then this work challenges that assumption, to some extent. A large name server collection will work well for some resolvers who will make the best choice from the available set, but not for many others. Perhaps anycast is a better approach for optimizing the server set in terms of query times and at the same time offering a line of defense against DOS attacks.

Short Notes

Resolver Testbed
Paul Hoffmann of ICANN reported on an effort to build a test framework using a virtualbox VM filled with resolvers, a simulated root server and a mechanism to generate particular resolver to root query interactions simulations. Code is available at [https://github.com/icann/resolver-testbed].

DNS Security
Ralk Weber presented a historical perspective of security issues in the DNS, including efforts to corrupt the DNS via cache poisoning, and later by the Kaminsky attack. There were DNS DOS amplification attacks, DNS Changer, and random subdomain name attacks on authoritative servers. These days we are seeing orchestrated multi-part attacks that exploit weaknesses in domain name registrar systems to hijack a domain name.

DNS Interception
The rise of open DNS resolvers as an alternative to the ISP-provided resolvers has been a prominent feature of recent years. Such resolvers have been around for some decades, such as the DNS service behind 4.4.4.4 and that operated by OpenDNS. It gained more attention with the launch of Google's service, which has been promoted as a fast and 'honest' service, in that it does not filter or alter responses, and does not perform NXDOMAIN substitution. But such moves to bypass ISP-provided DNS resolvers have inevitably provoked a reaction. We hear of ISPs advertising the anycast IP addresses of these open servers in order to intercept such DNS queries and redirect them back to the original resolution environment. Other ISPs perform DNS interception, where all UDP (and most times TCP) traffic to port 53 is passed to a local DNS resolver irrespective of the IP packet's destination address. How prevalent is this practice? This presentation described an experiment that attempted to measure the extent to which DNS interception is taking place. It is a challenging measurement to perform at scale, and while various probe-based test platforms (such as Atlas probes) can perform these DNS tests ther issue with these particular platforms is an issue of scale and selection bias. So yes, DNS interception

happens but it's not clear how many users in the entire Internet have their DNS intercepted in this manner.

Multi-Signer DNSSEC Management
The attack on the DYN service in October 2016 in the US had a number of consequences. One of these was the realization that using a single service provider to run your DNS authoritative name service is not necessarily a good idea. But outsourcing the serving of a DNSSEC-signed zone to multiple service providers can present some challenges. If the service providers also performs various customized responses (what is often called "stupid DNS tricks") and use their own keypair and perform on-the-fly signing then a multi-provider DNS service model can be made to work. Shumon Huque's presentation explored how various permutations of shared and per-provider KSK and ZSK keys can be made to work in a reliable manner.

Unsupported DNSSEC Algorithms
The world of cryptography is one of constant change. New algorithms appear and existing algorithms are deprecated. What happens with DNSSEC tools when unsupported algorithms are used in the various parts of the zone signing, serving and validation processes. Matthijs Mekking reported on the results of testing a number of widely used DNS signers, servers and resolvers to investigate their behavior when unsupported algorithms are encountered. In general, the tools work as expected, treating unsupported algorithms in the same manner as unsigned data in general. Some tool crashes and anomalous behaviours were observed in some cases.

Offline KSK in Knot
Jaromír Talíř reported on how the .CZ domain was signed in the past and the introduction of a Knot DNS signer allowed the use of an offline KSK in the zone signing process.

DNS Flag Day
When the extension mechanisms for DNS (EDNS(0)) were introduced DNS resolvers adopted a conservative stance. If a query containing EDNS(0) options did not elicit a response from an the authoritative server, the resolver uses a number of workarounds, requerying without EDNS(0) options and requerying using TCP. The DNS Flag Day was the "stop day" when resolvers no longer supported this workaround behavior, and authoritative servers needed to correctly response to EDNS(0) queries. The flag day was largely deemed to be a success.

This has prompted consideration of another of these flag days for the future as a means of improving the robustness of the DNS. For the next DNS Flag Day, the objective is evidently somewhat more ambitious in scope, as the plan is to address the current issues with large DNS responses over UDP and the problems with reliability of IP fragmentation of the large UDP packet, particularly in the case of IPv6.

The Ultimate Stub Resolver
As Olafur Gudmundsson explained, the original stub resolver was implemented as a simple call into an operating system module that performed DNS resolution with a limited query repertoire. This model was refined with language-specific libraries, such as DNSjava, DNSpython and similar. These modules were 'unpacked' into DNS libraries and APIs to provide an application with greater flexibility and control over the DNS resolution function.

Cloudflare's experience with DNSdist is interesting. The tool is positioned as a DNS load balancer across multiple DNS servers, whereas it is actually a highly effective traffic steering device with caching. All stub resolvers should have this level of functionality! The same approach works for the so-called recursive resolver farms, where traffic steering by query name, coupled with caching, allows each member of the farm to operate exclusively across a set of query name and query types, eliminating the need to share query responses across the entire farm.

OpenINTEL
In the same way that search engines repeatedly crawl the web space to assemble their index data, it is possible to crawl parts of the DNS space. This project does precisely that, repeating the crawl on a daily basis, and the resultant data set becomes in effect a long-term record of the name space and its evolution. They query some 216 million domains per day, collecting some 2.3 billion DNS records per day in the process. One illustration of the use of this tool was an analysis of authoritative servers before and after the October 2016 DNS attack. Many key customers of DNS providers switched from using one provider to multiple providers in the aftermath of the attack.

DNSKEY queries and the KSK Roll
Ray Bellis of ICS reported on a look at the volume of DNSKEY queries and RFC8145 queries seen at E and F root servers over the KSK roll. The installation of the KSK-2017 into the DNSKEY record did not generate a visible change in DNSKEY query levels seen by these root server clusters. The KSK roll itself did generate a 3x increase in observed DNSKEY queries. The absolute level of queries was nmot a concern, but the reasons for the higher query rate were not clear. The revocation of KSK-2010 in January 2019 saw a further 5x increase in query levels. The removal of the revocation entry saw the query levels revert to the post-roll level, and subsequent investigation pointed to a bug in earlier versions of the Bind resolve r that caused query repetition. But we have still yet to see query levels drop to the levels seen before the KSK roll, and the reasons for this are again unclear.

What part of "NO" is so hard to understand?
I presented on the queries seen when the server's response is "no such domain" (or NXDOMAIN). Instead of a single query we observer an average of 2.4 queries seen by the zone's authoritative server when the domain name itself does not exist. The presentation attempts to explain this, looking at happy eyeballs, DNSSEC signed vs unsigned, the impact of DNS resolver farms and the curious observation that NZDOMAIN elicits more queries than a positive response. The overall behavior of the DNS is sometimes rather difficult to fully explain given the interaction between various independent timers and various resolver architectures.

Incentivizing the Adoption of New Standards
It was reported that the reason behind the large number of DNSSEC-signed zones in .se was a financial incentive to registrars where a signed zone was charged a lesser registration fee. A similar program was used in .nl and this has been extremely successful. They are now using financial incentives to promote the adoption of IPv6 DNS servers, DMARC and STARTTLS, promoting IPv6 and tools to support secure mail.

DNS Fragment Attack
Kazunori Fujiwara of JPRS described a cache poisoning attack using IP fragment substitution. His presentation described how spoofed PTMU ICMP messages can prompt fragmentation and how an attacker can then attempt to insert fragments into the DNS response. He proposed some techniques to protect against this attack vector.

Flamethrower – DNS load and functional testing
An alternative tool to DNSperf with realistic query rate patterns. Code is available [https://github.com/DNS-OARC/flamethrower].

Respdiff – Regression and interoperability testing
A tool to generate and send queries to m,any name server instances and compare the responses. Code is available [https://gitlab.labs.nic.cz/knot/respdiff]

Meeting Materials

The full agenda and presentation materials for the 2019 symposium can currently be found at https://indico.dns-oarc.net/event/31

Disclaimer

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

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