The ISP Column
A column on things Internet

June 2024
Geoff Huston

DNS Topics at RIPE 88

RIPE 88 was held in May 2024 at Krakow, Poland. Here's as summary of some of the DNS topics that were presented at that meeting that I found to be of interest.

DNSSEC Bootstrapping

How can you start up a DNSSEC relationship between the parent zone and the delegated child zone in a simple, robust and fully automated fashion? RFC 8078 describes a way to automate the management of the parent-side DS record by using CDS/CDNSKEY records in the child zone, but the essential trust element that allows the parent to authenticate the CDS/CDNSKEY is DNSSEC itself. Once a DNSSEC relationship between parent and child has been set up, then it can work effectively. However, the question remains as to how can this trust relationship be established for the first time?

An approach being proposed by deSEC's Peter Tomassen is to use the nameservers that serve this unsecured zone. The assumptions are that:

- the zone that contains these nameserver names is not the unsecured zone,
- the zone that contains these nameserver names is itself already DNSSEC signed, and
- there is some trusted relationship between the operator of this nameserver zone and the unsecured zone.

The initial CDS/CDNSKEY RRSET is published in this nameserver zone, using a DNS name that is the concatenation of the unsecured zone name and the nameserver zone name, with service labels to glue them together. For example, to load an initial DS value for the zone `example.co.uk`, served by the nameserver `ns1.example.net`, the operator could add the record to the `example.net` zone

```
_dsboot.example.co.uk._signal.ns1.example.net  IN CDS <value>
                                               IN RRSIG <signature record>
```

There is an element of a leap of faith in this design, namely that there is a trustable relationship between the zone operator of the unsecured zone's nameserver names and the unsecured zone. This trust relationship is neither visible nor verifiable by any relying party, and the same question applies to the nameserver zone about how it was bootstrapped into a DNSSEC-validated state in the first place. Its turtles all the way down!

Right now, it appears that most of these processes associated with DNSSEC management are managed through custom API's. I suspect that the DNS industry is comfortable with these APIs and there is no screaming rush into automated processes. However, some DNS implementations and some DNS operators have implemented this automatable process. The details of this approach can be found at draft-ietf-dnsop-dnssec-bootstrapping.

Presentation link

DELEG Update

A proposal to add additional details to delegation records has been circulating in the IETF since late 2023. Various extension to the DNS, including encrypted transports, service bindings and such, have been a source of increasing complexity in the implementation of DNS resolvers, particularly in the area of delegation. This DELEG proposal has been to define an extensible parent-side delegation record that can not only flag the existence of a zone cut point and a delegation, but also describe the attributes of the nameservers that are serving the delegated zone. It builds upon recent work with service binding records in the DNS.

The concept of delegation is central to the design of the DNS, and working on changes to the manner of delegation in the DNS defines a path for the evolution of the DNS that is not without its attendant risks. Design-by-committee has a woeful track record, and the temptation to kitchen-sink a design is often overwhelming, adding adornments and tweaks which may be mutually contradictory and often unused in operational environments. Committee work has to strive extremely hard to arrive at a point of elegant minimalism!

DELEG evidently attempts to address three shortcomings of the current DNS delegation process.

The first is that the parent's presentation of the NS records to a resolver is unsigned and is therefore prone to various forms of substitution attacks. DELEG, like a number of other proposals, attempts to address this by proposing a new delegation resource record that is authoritative in the parent zone, so that can be served with a DNSSEC signature. This is an extension to DNSSEC-signing in the DNS, where it's not just the DNS response that it being authenticated, but the resolution path being used. The marginal benefit of this additional function is difficult to assess, given that a resolution path that misleads the recursive resolver will more than likely end with a response that will fail DNSSEC validation in any case.

The second is an inability to specify an alternative transport protocol to use to query the authoritative servers of the delegated zone. Again, the general benefit of this is unclear. The encrypted channel transport protocols need to perform an initial packet exchange to set up the shared encryption state. In the stub-to-recursive resolver situation this overhead can be offset by the subsequent queries that use the established channel, and there is a marginal gain in avoiding the delays in switching over from UDP to TCP for large responses. However, this is not necessarily the case in the recursive-to-authoritative scenario, as a single recursive resolver may query a large number of authoritative servers in a relatively short span of time.

It's the final shortcoming where the DELEG resource record may offer a useful function. The inability to use a CNAME record as a target of an NS record means that it's hard to move a DNS name server name away from the control of the child zone operator. If a DNS operator wishes to dynamically generate name server records that can perform load balancing or offer records that are located close to the querier to improve DNS performance, then this is practically impossible in the current NS record structure.

DELEG looks promising, but there is the residual consideration of the backward compatibility of the proposed DELEG structure. Do the NS and glue records that are also packaged in the delegation response have to match the DELEG record contents? Or do the DELEG records specify additional name servers that may be queried in a higher level of preference to the servers listed in the NS records? Do we really need this somewhat clumsy stuffing of referral responses with both DELEG and NS records in any case? Why not just borrow from the earlier REFER proposal and respond with only DELEG records if the query indicates via an EDNS(0) option setting that it can handle DELEG referrals? Otherwise the server would respond with NS and Glue records as it does today to indicate a zone cut and referral.

Many more words will be generated on topic without doubt. As to whether the effort can maintain focus and be used to produce a useful extension to the DNS, that remains to be seen.

Presentation link.

DNS Energy Consumption

When asked about the cost of using encryption, a conventional response is to quantify this cost in terms of additional delay. But in today's world it is also instructive to consider this cost in terms of its overall carbon footprint. At Afnic, the French domain name registry, they have calculated the cost of hosting a domain name for a year at 147g of CO_2, down from 252g some 5 years ago.

In this study they are looking at the energy consumption in resolving a DNS name. They are interested in the energy consumption profiles of using the DNS over UDP, TCP, TLS, and HTTP, and also looking at the incremental cost of using DNSSEC and DNSSEC validation. The approach they are using is to look at total energy consumption of a particular configuration, look at the traffic profile that was serviced at the cost of this level of energy consumption, and derive some form of unit energy consumption.

There have been some quite remarkable improvements in the energy consumption of computation over the past couple of decades, but I suspect that the major motivating factor was not necessarily one of reducing the carbon footprint of computation. I suspect that the major motivation was to extend the battery life of mobile devices, packing more features requiring more computation into small lightweight packages that did not run at scalding temperatures!

Presentation link.

DNS Resolver BCP

When the DNS4EU program was being aired, the RIPE DNS community decided that what they could do to help was to generate a "best current practices" on operating a large-scale DNS recursive resolver system. This document is RIPE 823, DNS Resolver Recommendation.

This session compared some of the recommendations in the BCP document with operational practices used by a small selection of large-scale recursive resolver operators.

Some of the interesting topics raised included the question of whether a large-scale operator operates multiple DNS server implementations or operates a DNS platform monoculture. Multiple implementations can generate situations that exacerbate fragility and complexity in the operational environment and are often avoided for this reason. Other operators deliberately use multiple different implementations to remove a single point of vulnerability and use the DNS's failover capability or particular operational controls to perform rapid repair in the event of single unit failure. It’s an interesting trade-off that perhaps is a salient factor when looking at large-scale service infrastructure.

Should these resolvers be provisioned on dedicated hardware platforms, or operate on virtualised cloud platforms? With dedicated cloud platform that allow the operator some level of control over the shared resource allocation there is little to discern between could and dedicated metal platforms.

Should each instance in a large anycast-style configuration also take steps to ensure high availability at each site? It appears that this quickly becomes a consideration of cost and performance. Anycast can be configured to support rapid failover across sites, and the larger multi-site setup is its own implementation of high availability, which tends to suggest that per-site high availability measures are an additional cost which is not so clearly justifiable. On the other hand, switching clients for a nearby service instance to a remote instance has performance implications, and if the marginal cost of per-site high availability engineering is sufficiently low, then in many scenarios this multi-level high availability engineering can be justified. Again, there is no clear single answer here.

The question of EDNS Client Subnet (ECS) support is a difficult one. As the specification of Client Subnet (RFC 7871) states, "We recommend that the feature be turned off by default in all nameserver software, and that operators only enable it explicitly in those circumstances where it provides a clear benefit for their clients." For large scale resolver scenarios which operate as an open service, the situation where the client and the resolver are at some distance from each other can arise, and when the service being queried operates some form of steering using the DNS, then poor outcomes can arise. On the other hand, the privacy considerations are a matter of legitimate concern. Some open resolver operators make ECS an option, by placing ECS at a different resolver IP address, while others set ECS to on or off for the entire service. Of course, this is less of a consideration in very dense service configurations, as the entire objective of such dense configurations is to locate the service close to the client in the first place.

Should a service operator use RPKI? "We generate ROAs but we do not drop invalid routes" is a common response. There is some sense to this approach given that the networks that host such resolver services are themselves stub and not transit networks, and it makes some operational sense for the invalid route filter functionality to be deployed in transit networks.

The question of using "negative trust anchors" is a thorny one. DNSSEC is an unforgiving security tool, and if you misconfigure DNSSEC then validating resolvers cannot resolve any names in your zone. "Negative trust anchors" is a fancy term for a DNSSEC validation exception list that causes the resolver to ignore DNSSEC validation errors. It's a a pretty crude "solution" to the underlying issue of the fragility of DNSSEC, but we've not been able to come up with anything better to allow the DNS to distinguish between operational mishaps and deliberate tampering with the DNS.

Should an open DNS resolver service use channel encryption (DoT, DoH, and DoA)? There is a mixed response here. From a security perspective there is a strong case to be made to use both the server authentication and channel encryption properties of these forms of DNS transport, as the DNS traffic between the client and the resolver service is likely to traverse across the open Internet where various forms of traffic inspection and interception are possible. However, it does come with an incremental cost to the service platform and the client may need to perform additional configuration steps.

Presentation link.

An Introduction to the Root Server System

One of the poorly understood aspects of the entire infrastructure of the DNS is the root server system, and ISC's Jeff Osborn presented on this topic explaining some actions that have been undertaken in the root service community to try and bridge this lack of appreciation of the role of the root servers in the Internet landscape.

Presentation link.

Subsequent discussion on this topic pointed out that while there are some evident gaps in the understanding of the role and operation of the root server system in many public policy venues, there may well be similar gaps in understanding on the part of the service operators about the nature of the public policy concerns. The concept of shared common infrastructure is common to many activity sectors, but when the critical nature of this common infrastructure is pointed out and when it is observed that for many nation states this vital common infrastructure is operated by foreign entities who are not answerable in any way to their own national environment, either in a regulatory sense or in the sense of service delivery, then clearly there is some mismatch of expectation and understanding going on.

Making a case that a largely unblemished operational record of operating the common root of the name system for some four decades is sufficient to reassure policy makes that the system will never be in a position that it can be abused by actors to the detriment of national infrastructure was never going to be an easy proposition.

A conventional historical response to such international issues has been in the adoption of international treaties, binding signatory nation states to a set of behaviours that may well allay some of the concerns arising from such dependence on foreign entities. An international treaty framework lies behind the international postal system, and the same is the case for international telephony. Yet the Internet is an outcome of the deregulation of the national communications systems in a number of major economies and from the outset the Internet effort eschewed the incorporation of specific roles for governments. Its infrastructure, namely the single name space, the common address space and the common routing domain, is not necessarily recognised within international treaty frameworks, and neither is the operation of this common infrastructure a protected activity.

Personally, I suspect that this form of national disquiet is not going to go away in international policy circles. As the world seems to lurch from one international crisis to another, the concern that the common infrastructure of the Internet could be abused with the acquiescence or even with the active engagement of some nation states with the intention of disrupting the communications infrastructure in other nation states in such ways seems to be a continuing aspect of our environment. I'm not sure that reassurance of the form that: "well, nothing really bad has happened so far." is enough. Pointing to the use of international treaty structures used in other international communications realms seems like a rather ineffectual response that will incur its own set of consequences, foreseen or otherwise, which tends to argue against such a move. Making some very major technology changes to the nature of the root of the DNS and the manner of its operation that may try to address this level of inter-dependence presents its own issues with acceptance, even assuming that we could devise such changes to the way the root zone is used in DNS name resolution.

Which probably goes a long way to describe why the operational framework of serving the root zone of the DNS seems to be frozen in a 40 year-old time warp!

Disclaimer

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

Author

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

www.potaroo.net