July 2024
Geoff Huston

Bytes from IETF 120 – DNS Topics

DELEG

This is a newly formed Working Group to look at the mechanisms for delegation in the DNS, intending to define a delegation with a richer set of functions than what we have with the NS delegation record. There are a number of shortcomings with the current form of delegation in the DNS:

- In the process of top-down name server discovery the NS delegation record is used from the parent zone, but the parent zone server is not authoritative when serving this resource record. These records cannot be DNSSEC-signed and therefore cannot be validated.
- NS records can only specify the DNS name of the same servers and not their IP address. Over the years this has been a source of vulnerabilities, operational confusion and additional resolution delays associated with "glue" records.
- This constraint in the target of NS records also restricts the use of CNAME alias records in delegation.
- Delegation records do not specify any transport protocol that can be used to query the delegated zone's nameservers. A record must probe for support for alternative DNS transport protocols.

It appears that with the evolution of the DNS with a richer set of transport options, including channel privacy, the DNS security framework, and a desire to continually improve the efficiency and speed of name resolution we are wanting to take the basic signal of a delegation of a domain in the DNS and augment that with a richer set of information about how to contact the delegated zone's name servers.

The DELEG proposal from 2023 is by no means the first such proposal in this space, and here the approach is to build upon these previous proposals by taking the rich set of service capabilities that can be described in a service binding (SVCB) DNS resource record, shifting the point of authority to the parent zone, thereby allowing DNSSEC-signing of this data, and also permit aliasing.

If this was a clean slate and this new form of zone delegation did not have to coexist with any existing DNS infrastructure, then such an approach appears to offer a feasible way forward. But this situation is complicated by the need to coexist with the existing infrastructure of the DNS, including not only NS records, but also the entire issue of provisioning, the distinguishing of roles between DNS zone management and operational management of serving the zone's content.

The typical IETF response to such issues is to take a step back and consider what are the functional requirements for this enhanced delegation function, and the consider the parameters for evaluation of such requirements.

Having the parent zone be the point of authority for a DELEG record allows the record to be signed by the parent's key, which has the potential to improve the efficiency of DNSSEC-validating the sequence of delegations. Having the DELEG record also include the equivalent of the glue record (the IP addresses of the name servers) in the form of SVCB's ip hints. It also improves the level of trust in these records in terms of their authenticity as the entire SVCB RRset can be DNSSEC-validated. That implies that the risk of having the recursive resolver being misled by false delegation information is countered in this approach. But is this of value to the resolver? DNSSEC is not exactly popular out there in the public

Internet and the case for validation of delegation records with the associated time and processing penalties that such a validation function would exact on name resolution seems to be extraordinarily tenuous. DNSSEC is conventionally used to validate DNS resolution outcomes as being authentic or not, and the way in which the DNS record was resolved is not material to the DNSSEC validation outcome. If using DNSSEC to validate resolution outcomes has not been a spectacular success in terms of adoption, then adding to the DNSSEC validation load by also validating these referral responses, essentially validating each delegation step, seems to me to take an unrealistic view of the role and value of DNSSEC in the public DNS infrastructure.

The issue of specification of the transport protocol used to query these name servers is also a questionable requirement. This is not conventionally a query made by the client resolver. This is a query made by the recursive resolver to an authoritative server for a zone. The client's details are not a part of these queries, and the query is only made if the recursive resolver does not already have unexpired locally cached information that it could use to answer the query directly in any case. The case that justified the overheads of using an encrypted transport to make DNS queries, namely that the client's IP identity, the DNS query that they are making, and the time of the query are accessible to a third-party onlooker under certain circumstances, is a far weaker case when the client identity is no longer part of the query. The scale of the potential privacy compromise is far smaller in this recursive-to-authoritative context than the previous work on stub-to-recursive.

So that leaves us with the current inability to replace a name server name with a CNAME in the NS model of delegation. Aliases are used extensively in the DNS operations area, as it allows a service name to be lifted out of the locus of control of the zone administrator and placed into the zone of control of the service operator. It appears to be a common way to pass control of a DNS name over to a CDN operator for named web servers and similar, and performing the same form of alias in a zone's name server context provides a similar way of shifting the locus of control of a name server function to a specialised DNS operator.

In addition, there are set of constraints that need to be carefully considered. In the diverse world of recursive resolvers and authoritative name servers change is slow and incremental. A new resource record will take time to be accepted by resolvers. DELEG records cannot replace NS records in the parent zone, or at least not for the present, so perhaps an EDNS(0) signal in the query to signal that the resolver can accept DELEG records would ensure that DELEG referrals responses are only passed to resolvers who signal their capability to accept them (similar to the DO bit in queries for DNSSEC signatures).

There is undoubtedly more work for this working group to do, more RFCs to generate on requirements, behaviours, modifications and corrections, code to develop, deploy, correct, vulnerabilities to discover and exploit. Yet if the lasting incremental benefit in the DNS as most of us use it is some marginal change in the use of alias name forms in referral responses then one has to wonder if anyone really remembers the DNS Camel from IETF 101, some eight years ago. If this is a new term to you, then it's probably a good time to pause here and look up Bert's presentation and then wonder if DELEG is a stunningly useful additional to the DNS, or just another straw to load onto the DNS Camel's back.

DNSOP

Over in the general area of DNS specifications, the DNS Camel is still under stress from the ever-increasing load of new work!

Since IETF 119 there have been two more published RFCs. RFC 9615 updates the specification for bootstrapping DNSSEC, using in-band publication of the proposed DS records in the DNS rather than relying on out-of-band methods that are generally used for NS records. RFC 9619 is a short reminder that the query count field in DNS queries is a maximum of one.

There are a number of specifications that are well down the RFC-to-be track, including a revision of "DNSSEC Trust Anchor Publication for the Root Zone," a revision of the priming query used by DNS

resolvers and "zone version," a way for authoritative DNS servers to provide information regarding the version of the zone from which a response is being generated.

In Working Group Last Call is "avoid fragmentation," the draft which advises DNS operators, both servers and resolver clients, to avoid the situation of passing DNS over UDP packets where IP level fragmentation is likely to be invoked. The issues involved in fragmentation are not only the cases of discarding of IP fragments but also the ability to inject trailing fragments that alter the DNS response and thereby attempt to poison the resolver's cache.

Also ready for Working Group Last Call is "compact denial of existence," which appears to build upon the concepts described in RFC 4407. The situation arises with a DNS-signed domain that uses a front-end signer rather than serve a pre-signed zone. Here the resolver will indicate that there is no resource record data for this name (NODATA), as distinct from responding that there is no such name at all in the zone (NXDOMAIN). These synthetic NODATA responses are smaller and have fewer signatures. To distinguish between non-existent names and empty non-terminal names (names that are a delegation), the NSEC type bitmap for non-existent names will include a setting for the pseudo-type NXNAME, while empty non-terminals will not. Admittedly, this work is only applicable in some cases, but the use of front-end signers in DNSSEC is common with the larger DNS operators, and having a robust way of handling negative responses that avoids the pitfalls of zone enumeration and avoids large responses of NSEC and NSEC3 denial of existence responses does make a lot of sense. However, nothing comes for free, and this mode of specific responses for non-existent labels within a domain opens up the authoritative server set to large scale so-called "Water Torture" attacks where random names in the domain are queried, either directly via botnets or across a wide range of public resolver services, in order to intentionally generate non-existence responses from the domain's authoritative servers. If the authoritative servers cannot synthesize their own NXDOMAIN responses (using Aggressive Use of DNSSEC-Validated Cache (RFC 8198)), they must pass all queries on to the domain's authority servers, making resource exhaustion more likely at those latter servers.

The specification "domain verification techniques" aims to circumvent the bloat in domain validation records at the apex of a service name by proposing the use of provider-specific challenge records (such as `_foo-challenge.example.com IN TXT <challenge record>`.)

The draft on "generalized notifications" is an effort to take the existing DNS Notify mechanism used by a primary server to notify all the secondary servers that the zone has been updated, defined in RFC 1996, and define a more general approach that can be used in other contexts of the DNS where data synchronization is useful. The particular motivation for this work has been the child-to-parent communication of delegation and DNSSEC keying information in the CSYNC, CDS and CDNSKEY records. The current method of polling for changes in these records suffers from all the inefficiencies of uncoordinated polling, where the desire for a responsive process that quickly recognises change causes high frequency polling. Notification allows the publisher of the changed record to notify the dependent zone operator of a change in the data. The issue considered in this work is a structured method to define the target of the notification in a DNS record. It is proposed to use a new resource record, DSYNC, that describes the endpoint as a scheme, port and target server name as a 3-tuple. It is useful to consider whether this is another case where the SVCB server record may be more useful in this context, as it would allow IP address resolution data, port data, DNS transport level protocol and the notify type and scheme to the notification target description. The uses for SVCB keep on popping up these days with regularity in DNS work. It seems as if we've been waiting for this bundled information model in the DNS for quite some time, and now that it's here its being adopted with much enthusiasm!

"Delegation Revalidation." It was back in 1990 that it was recognised that the DNS was vulnerable to cache poisoning attacks. An attacker could implant false name-to-address mapping information into a DNS resolver that the resolver would then hold in its local cache and use to answer subsequent queries. At the time this was a significant problem for the network because the network's trust framework was based on the integrity of the mapping of DNS names to IP addresses and the integrity of the routing system in directing IP packets to the "right" destination. To put it simply, as long as you sent packets to

the right IP address you necessarily trusted the response as authentic. This DNS attack corrupted the mapping of a DNS name to the corresponding IP address in a way that was impossible for an end user to detect. In response the IETF embarked on the path that became DNSSEC. As was recognised at the time, DNSSEC is only a partial solution, in that it cannot authenticate the service that a user subsequently connects to. We turned to SSL and then TLS, which did not "protect" the DNS resolution outcome per se but was able to provide what turned out to be considered adequate assurances of authenticity about the service that a user was about to connect to. DNSSEC was left to languish, and without DNSSEC it is still possible to inject incorrect data into the name space. That has not stopped numerous efforts to improve this situation while at the same time eschewing DNSSEC. I can't help but think such efforts are of marginal benefit, if any. If you want users to detect inauthentic responses from DNS resolution queries, then DNSSEC is the best way we know how to achieve that. If you want the same outcome without DNSSEC, then the results generally take more time and provide little in the way of effective assurance at all, and delegation revalidation is no exception to this general observation. Here the parent zone nameservers are ranked lower in credibility to the name servers listed in the child zone. Yes, it is possible to refresh a resolver's cache of nameserver records by querying on the child size of the zone cut, but without DNSSEC the trust involved here is back to the basic proposition: "If I send a packet to the right IP address, I'll trust that the response I get is the right response." Yes, the world of deployed DNS has largely rejected the use of DNSSEC, and the result is that we are left trying to make relative judgements between what are essentially poor outcomes in any case.

"Nameserver Selection." It's been drilled into the minds of DNS admins that all zones must be served by two or more name servers. The rationale is ostensibly one of service resilience, and when a DNS resolver is given multiple name servers for a zone then if one server is unresponsive then another server will be queried before giving up. As always, when a service is performed by multiple servers, other motivations surface. Service operators would like to spread the query load across the multiple servers in order to improve the overall efficiency of the server set. Recursive resolvers would like to use the server with the fastest response. Nameserver selection has long been considered part of the "black art" of the internals of DNS resolution infrastructure. This draft, nameserver selection, describes the server selection strategies used by various popular resolver implementations, and describes the ways in which such approaches can be subverted and exploited. The draft contains some recommended practices to address vulnerabilities.

DNS Load Balancing

A side-meeting was also held in the week on the related topic of DNS-based load balancing. Once you get over the confusion of whether the objective is to distribute the DNS query load across multiple authoritative name servers, or to use the DNS to create customised responses with the objective of distributing the consequent service load across multiple service delivery platforms. Or whether these customized DNS responses are intended to steer clients to the "best" server in terms of proximity and current availability service delivery capacity is an open question. Should we use an approach the further refines Client Subnet, where the client provides some information as to its location and the authoritative server provides a response which has performed the server selection function on the server side, or whether the server returns the set of available choices and pushes the decision as to which to use back to the client (as in the case in the selection of an authoritative server in DNS resolution). It is unclear where standardized specifications would fit in here.

Responding to Key Trap

There has also been some discussion about the Key Trap vulnerability and related zone configuration issues which create undue levels of work for resolvers. It's all well and good to say that a zone operator should not use multiple keys with the same hash value, but if the zone is operated by a party that wants to create a trap for resolvers, then unless the resolver has some internal defence then it's still vulnerable. At present a resolver may elect to abandon resolving a name at its discretion, and simply return the SERVFAIL code in such circumstances. This response may be triggered by spending too long to resolve

a name, too many processor cycles, too many secondary queries of any form or any other activity. At some point implementors and infrastructure operators should have some discretion to define the bounds of work that their systems should perform in response to a service query. A perhaps complementary response is to define a minimum configuration that resolvers should reasonably support, such as a maximum of 2 keys with the same tag value, or 2 levels of indirection of name server records without associated glue records, and similar.

The State of DNS

As usual this DNS activity is operating at a relatively rapid pace as we pick up an idea and run with it with some enthusiasm for a short while until a new distraction appears as a novel idea and we chase after that instead. At the moment it appears that the service binding construct (SVCB) is the shiny new idea, and there is no shortage of where it can be applied in the DNS. Service binding allows for a means of bundling information in a single DNS transaction, and compared to dynamic discovery through multiple queries and multiple connection probes to discover service capabilities it has much going for it a present. How long this will last and what will replace it is still somewhat unknown at present.

It also appears that every major DNS operator and service provider is placing their own sets of pressures for evolution in the DNS and the cumulative outcome often appears fragmentary and unfocussed. The words of conservative caution a few years ago about continued bloat in the DNS specification under the general theme of the "DNS Camel" appears to have been largely forgotten. And even the efforts to track the number of DNS specification documents, and the volume of these specifications in terms of a simple word count appear to have lapsed. The result is a resumption of broad and apparently unfocussed activity in almost every part of the DNS.

As a result, I have absolutely no idea if there is a confident answer to the question: "Where are we going with the DNS?"

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