November 2023
Geoff Huston

DNS at IETF 118

The IETF met in Prague in the first week of November 2023, and, as usual there was a flurry of activity in the DNS-related Working groups. Here's a roundup of those DNS topics I found to be of interest at that meeting.

Re-thinking the DNS

Prior to IETF meetings there is an opportunity for informal discussions, under the label of a "hackathon". The topic in the DNS was to reconsider the DNS. A major constraint in any such discussion is that the installed base of the DNS is highly resistant to many forms of change. The name space definition and the data model of the DNS are fixed, as is the model of collection of delegated authorities assembled in a hierarchical manner. With so much that must be retained, such conversations about a "new" DNS quickly turn to discussions about incremental evolution that preserve backward compatibility.

There are a couple of areas of useful consideration. One is the way that DNS uses the underlying transport layer, and its model of a preference for using UDP for queries and responses and the visible trend to placing more information into DNS responses. The work on DNS over HTTP (DoH) opens up some interesting conversations about using a reliable, authenticated and encrypted streaming transport by default, and also opens up a conversation about push vs pull as a model of information delivery. However, this year's hackathon headed in a different direction, namely the delegation record in the DNS.

Confusion over the NS record has been a significant issue for the DNS over its entire lifespan. It's a piece of data that is held in both the child and parent zones, yet, confusingly, it's the child that is authoritative for this data, while logically in terms of the DNS nameserver discovery process it's the parent's copy of this data that that is used to guide the name resolution process. So the thought process is to consider what a compound *delegation record* might look like. These days such a record may refer to multiple zone hosting providers, it may allow for suggested query transports in a similar manner to the SVCB record, it would likely be authoritative at the parent, and not be part of the child zone (similar to the existing DS record).

It's an interesting area of thought for the DNS as it applies the same consideration that the DNS is a provisioning protocol for accessing named applications and services to the DNS query resolution protocol itself in term of querying the nameservers for a zone.

The hackathon report at the DNSOP Working Group meeting made mention of a new DELEG record that was able to carry this revised form of delegation in the DNS, but the report was also quick to note that further consideration of this idea would probably move further along in the coming months, and what might emerge as a proposal may not necessarily include a DELEG resource record per se. The report noted that the administrative structure of the upper layers of the DNS hierarchy, with shared registry operators, multiple registrars, and a collection of zone publishers has as much influence on the evolution of DNS design as the technical issues of efficiency of name resolution within the DNS protocol. I suspect that admitting this new reality of the administrative complexity of these outsourcing relationships is a major concession within many hard-core DNS circles!

Domain Control Validation

The DNS is used for many things beside the simple translation of a DNS name to an IP address. Some services want to know if the party that is requesting a service for a domain name is the real controller of that name, and the most common way of doing this is to ask the requestor to place a token into the domain name to demonstrate that level of control. It seems to have become a bit of a mess!

```
$ dig +short TXT bbc.co.uk
"google-site-verification=ITX3CwHXxGVfkCmhF4eSwdfo8h2ZGLAZ3zRpYvZi5XA"
"google-site-verification=RaiMXJBIiFvqXHd43kv_ekzmXT2l8ibq5Xy0mulndvU"
"voUGv5zARbEV516E/S8Ugsy9/FOgDGg4n/rpmKZQRROVOj0+2tgzKw3Tk9+Ks6qVbNKU18KTrR5khxTQutDvBg=="
"miro-verification=1a94b0fef7a6d5136a272d5cb425e8dc034e8cfc"
"apple-domain-verification=jFFO0rdS9IrxgWUR"
"msfpkey=3e2918m08bqxp19k63t73fj5b"
"dropbox-domain-verification=l5djk65wpy3z"
"atlassian-domain-
verification=SQsgJ5h/FqwMTXuSG/G4Nd1Gx6uX2keREOsZSa22D5XT46EsEuyaic8Aej4cR4Tr"
"2RLXso9TrRPyhWOEhYggL0U/r1D+g8H7z9RqDBOmcJjSbj88TobGKimtkCrXZNBkDXQDj891S4mDskNOJyWLdg=="
"MSUEdqCJpCtrI1JuH2-U"
"MS=ms10378910"
"docusign=a10ad7b6-cf7e-472d-8157-23061f5b5116"
"adobe-idp-site-
verification=9b850a4a56e3fac19aea1e0ac5db302e5cefab444cd73519dce1c72ccd4db058"
"Huddle"
"docusign=50f10407-e3e4-4f6a-aae4-712d4eb31329"
"docker-verification=aab67462-78f7-4ade-a86b-358645923430"
"J0kgGm0XqA3/6pLD4DHeC5x/dAduzT809P1Iwx/PRCYvVS32rv75RIHKC2aVz47dJxKhPlxGf3h3KXiL6+dyXw=="
"_globalsign-domain-verification=AQ2dURU9RbDOuheuLrx89LSUlA_btgMS6vmFXngBtE"
"v=spf1 a ip4:212.58.224.0/19 ip4:132.185.0.0/16 ip4:78.136.53.80/28 ip4:78.136.14.192/27
ip4:89.234.10.72/29 ip4:89.234.53.236 ip4:212.111.33.181 ip4:78.137.117.8 ip4:46.37.176.74
ip4:185.184.237.181" " ip4:185.119.233.144/30 ip4:185.119.232.158 +include:sf.sis.bbc.co.uk
+include:spf.messagelabs.com ~all"
```

The more this technique is used by various service operators, then the greater the number of TXT records that are stashed at the zone apex. Of course, a querier can't just ask for a particular TXT record. Each queries gets the entire bundle of records!

There is a proposal to pull these bundles of zone apex TXT records apart by using service specific challenge records, creating specialised records such as

```
_foo-challenge.example.com.  IN   TXT  "3419sdqa32453243d206c4"
```

The proposal contains a useful survey of currently used techniques in Appendix A, which is helpful, but otherwise the proposal is unclear as to what problem it is attempting to solve, and why. It should be remembered these days that the DNS is quite convoluted in terms of roles and responsibilities. The entity that is the notional zone holder (or "owner") may not be the same as the zone administrator who maintains the zone content, who, in turn, may not be the zone publisher (or publishers). When we talk about "validation" which of these parties is providing validation? Should a validation structure make these distinctions in various name administrative roles explicit?

The techniques we use today for name validation certainly have their idiosyncrasies, but in many ways it's a simple hack that works in an adequate manner. It's unclear what additional value this proposal is bringing to the table.

Generalised Notify

There are a number of theories about why DNSSEC is not being taken up with widespread enthusiasm.

Aside from the obvious issues with the efficiency of name resolution with the issues of larger responses and the consequent issues of dealing with UDP truncation and switching to TCP , there is also the added overhead of the additional queries associated with DNSSEC validation, and the added issue that very few stub resolvers at the network's user edge perform validation in any case, so the last hop in DNS resolution is essentially unprotected. Aside from these rather obvious (and quite compelling) performance and efficiency reasons to hold off DNSSEC-signing a zone, there remains a persistent school of thought that

the real showstopper for DNSSEC adoption is the lack of a standard way for a delegated zone to communicate the DS record to the parent zone.

Almost ten years ago, in September 2014, RFC 7344 proposed an automated way of doing this. This RFC defined a new pair of resource records, CDS and CDNSKEY, which are respectively a digest of the zone's DNSKEY in the same format as a DS record, and the zone's DNSKEY records. Like other entries in the delegated zone, these records are signed with the zone's key. The basic approach is that the parent regularly checks the child zone for published CDS records, and if a new value detected then the parent lifts a copy of the records from the child zone, performs a standard DNSSEC validation on the value, and then adds them into the parent zone as a new DS record. The parent may use the CDS record and copy it to the parent zone DS record, or it may prefer to use its own hash function over the child zone key, in which case it would use the CDNSKEY and perform the hash operation on that key value.

A number of domain admins have taken up this approach over the past decade. On the positive side it eliminates a whole new set of administrative processes to pass a value from the child zone to the parent, as the parent simply performs a regular poll against the the child zone's CDS record to detect a change in value.

But in the DNS, nothing is ever simple. A key question is how responsive should this polling system be? A polling interval of months is clearly just too infrequent, while a polling interval of fractions of a second is also clearly going too far the other way! A highly responsive system can imply a very high polling load if the parent zone contains many delegations, while a more infrequent polling rate causes slower convergence between parent and child zone for DNSSEC key transitions. More generally, polling is a highly inefficient way of communicating change.

This line of thought leads to the concept of borrowing the DNS mechanism used by a primary nameserver to inform its secondary servers that the zone has changed, namely the NOTIFY mechanism (RFC 1996). In a more general case of extending these NOTIFY messages, the child zone could send NOTIFY(CDS) and NOTIFY(CDNSKEY) messages to the parent to trigger a poll for new data from the child zone. This could also include NOTIFY(CSYNC) messages to allow the child to directly signal NS record changes to the parent. This *generalized NOTIFY* is described in a draft that was introduced to the DNSOP working group in this week's meeting. This work also proposes a new DNS record type, tentatively referred to as NOTIFY record, which is used to in the parent zone to publish details about where such generalized notifications should be sent for each delegation.

Johan Stenstam has pointed out that this could all be simplified if instead of coupling a NOTIFY message to trigger a pull, we could just use the DNS UPDATE mechanism (DNS Dynamic Updates), which could generalise this dynamic UPDATE to include NS as well as DS/DNSKEY records, without a prerequisite of a DNSSEC-signed zone. This is not a novel idea, and was described in a draft from Mark Andrews back in 2013. It appears that having new thoughts in the DNS is harder than it looks! Johan's point is that the original work required some effort to understand where to send the UPDATE, and for this reason with work was abandoned, yet the same problem is addressed in the generalized NOTIFY draft. NOTIFY and UPDATE are alternate methods for parent synchronisation. The key insight here is that sending the UPDATE to a DNS "service" destination rather than the primary nameserver allows the parent zone admin to implement its own checks on the contents of the UPDATE before applying them. It also can work for both signed and unsigned child zones.

There is no doubt that the area of the DNS that uses shared data is one of the more operationally troubled areas of misconfiguration in the DNS. In the case of delegation name server records (NS) the data used by resolvers is the parent's non-authoritative copy of such records in the downward traversal of delegations to find the authoritative name servers for the domain name, while the child's copy of the NS records is the authoritative version of the data. In the case of the delegation signer records (DS) the authoritative data is published by the parent, but the data that is derived from the DNSKEY key value that is authoritative in the child domain. Whenever there is replicated data in distributed system the key

question is: What should a client do when the various sources of the data disagree with each other? The DNS has no answers here.

The DNS is getting to a point where the collection of signalling and action tools is large enough to mean that there are multiple ways to customise these existing tools to achieve a desired outcome. The parent domain might be happy using polling to detect changes in the child domain for DNSSEC-signed domains, or they might prefer that the child domain signals the change of key with a NOTIFY signal, and then leave it to the parent to pick up the new data via a query, or there is the option of the child passing the new data to the parent via an UPDATE.

Deepspace DNS

When the work on protocol development work for an "InterPlanetary Internet" first started it was a DARPA-funded project supporting work NASA and MITRE and others to go beyond simple point-to-point long distance communications and examine how to support a network of communicating devices operating in the very high delay environment of deep space. The work evolved into the more generic work of "delay-tolerant networks" and the IETF published RFC 4838 and RFC 5050 as experimental individual contributions in 2007. In many ways it's a form of returning to the past of dial-up store and forward relay networking. The Bundle approach was taken up by the Delay-Tolerant Networking Working Group revising RFC 5050 with a set of RFCs: 9171, 9172, 9173 and 9174, in January 2022.

Of course, many of the networking issues remain in such hostile environments, including naming and name resolution. Marc Blanchet has been working on the issues that lurk behind a desire to map the DNS into such deepspace environments. Approaches include pre-loading local caches with resolution outcomes, a mapping of needed names and RR values into a "special" version of a deepspace zone, a new root zone for this name space, or some form of split horizon DNS.

The question posed by the presentation of DNS in Deepspace is whether this is topic of interest to the DNSOP working group. It does seem to be a very esoteric area of application of the DNS and somehow, I just can't see DNSOP spending its collective time and effort on this. But my track record of guessing what Working Groups choose to do, and choose not to undertake is pretty poor, so any outcome is possible!

Compact Denial of Existence in DNSSEC

There two fundamental performance and efficiency criticisms of DNSSEC. The first is that the inclusion of signed records in DNS responses bloats the DNS response size and this makes DNS over UDP an uncomfortable situation that requires UDP fragmentation or a rapid transition to TCP. The second is that validation of DNSSEC-signed responses can be very time consuming when using incremental queriers to assemble the validation path.

The DNS response size has been seen as a challenge, particularly in the case of signed NXDOMAIN responses. These negative responses contain the zone's SOA record (to let the querier know how long to cache the negative answer) and its RRSIG signature, an NSEC record that spans the query name, and its RRSIG signature and an NSEC record for a wildcard, to indicated that there is no wildcard in the zone,, and it's RRSIG signature. That's 3 resource records and 3 signature resource records. And at this point large responses are more likely, particularly with RSA-2048 crypto.

Work by Cloudflare in 2016 pointed out that responding in that manner to a query where the query name does not exist is perhaps more information than what was strictly asked for. The query contains a query name and a query type. A very minimal negative response to such a query would be to indicate that the requested query type for not exist for this name (NODATA). The subtle shift in this NODATA response is to indicate that all query types other than the type in the query itself exists in the zone.

This approach has been taken up in DNSOP under the name of "Compact Denial of Existence". The draft defines NXNAME, a pseudo Resource Record type, used in the NSEC type bit map for non-existent names.

This approach to denial of existence, coupled with elliptical curve crypto, can reduce the size of DNSSEC-signed responses to the point that signed responses should fit into UDP and front-end response signers can work even without complete zone knowledge.

Now, what can we do about the time taken to perform validation?

SVCB and DANE

There's little doubt that the DNS is the new steering protocol for the Internet (see https://www.potaroo.net/ispcol/2023-09/service-routing.html), and these days it appears that the two most useful records to achieve this in the DNS are the name translation record (CNAME), with its ability to transfer control of an individual name out of its original zone to a target zone that is operated by a service hosting provider, and the service binding record (SVCB), with its ability to define the parameters of a service connection without performing additional DNS queries.

There are a number of RFCs describing the use of Domain Name keys in the DNS, including RFC 7671 which describes DNAME itself, RFC 7672 which describes DNAME and MX records and RFC 7673 for DNAME and SDRV records. This work is performing the same function for DANE and SVCB records.

The DANE specification contains the advice that a client should follow the CNAME alias chain to its ultimate target host name, and use that target name as the base name for a TLSA query, but where no TLSA records exist for that name, it should also query for a TLSA record using the initial domain name. This work suggests that this is not the desired behaviour and only the target name should be used, as in the following example:

```
example.com.            HTTPS 0 xyz.provider.example.
www.example.com.        CNAME xyz.provider.example.
xyz.provider.example.   HTTPS 1 . alpn=h2,h3 ...
xyz.provider.example.   A     192.0.2.1

_443._tcp.xyz.provider.example.  TLSA …
_443._quic.xyz.provider.example. TLSA …
```

Some protocols that can run over TLS, such as HTTP/0.9 and HTTP/1.0, do not confirm the name of the service after connecting. With DANE, these protocols are subject to an Unknown Key Share (UKS) attack, which allows an attacker to deceive one peer of a secure communication as to the identity of the remote peer.

Current Focus of DNS Activity

The focus points of DNS activity are a constantly moving target. The frenetic pace of work on channel encryption for the DNS appears to have calmed down and attention has shifted to the server side of the DNS, looking at the issues related to the role of the DNS in content hosting.

The extreme end of the DNS on the client side, the stub resolvers appear to be highly resistant to change. They use DNS over UDP queries in the clear and the overwhelming majority of these queries are directed to the recursive e resolver operated by the client's ISP. Enterprise users have a higher level of willingness to use non-local open DNS resolvers, notably Google's Public DNS service, and in this space there is also a higher level of use of DNS over HTTPS and DNS over TLS. However, the measurements do not show any dramatic movement in these patterns of use, and the efforts to add channel encryption between the recursive resolver and the authoritative name servers appear to not have gathered much momentum.

These days there is more activity on the server side of the DNS, looking at ways to make administration of delegated zones more efficient, and also looking at ways to make tine initial rendezvous process more efficient by adding service profile data into the DNS.

I am interested to watch the evolution of the combination of DANE, CNAME, and SVCB constructs in the DNS as they apply to encrypting the last open peephole in TLS, the encryption of the SNI field. It strikes me that progress in this area has a pretty high barrier to surmount, given that a true end-to-end approach that does not entail trusted third-party intermediaries will need DNSSEC validation all the way to the stub resolver. In today's DNS that looks like it's still a big ask.

Disclaimer

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

Author

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

www.potaroo.net