March 2021
Geoff Huston

DNS at IETF 110

IETF 110 was held virtually in March 2020. These are some notes I took on the topic of current activities in the area of the Domain Name System and its continuing refinement at IETF 110.

The amount of activity in the DNS in the IETF seems to be growing every meeting. I thought that the best way to illustrate to considerably body of DNS working being undertaken at the IETF these days would be to take a snapshot of DNS activity that was reported to the DNS-related Working Group meetings at IETF 110.

DNSOP - DNS Operations

DNSOP is the general working group for most DNS topics, and has been for some years. This group carries most of the "core" DNS activity for the IETF. The topics considered at IETF 110 are as follows.

RFC 8976 - Zone Digest

It only took some 35 months, and 22 revisions of the internet draft to get this specification published. The concept is simple: It's a resource record that contains the message digest of the DNS zone file. The specification has been crafted very carefully and reflects the extraordinary levels of attention given these days to changes in the DNS repertoire that impact the content of the root zone of the DNS. In this case it's an adjunct to the DNSSEC security framework for DNS. DNSSEC allows individual entries in a DNS zone to be signed, allowing a receiver to verify that the response that they received is authentic, but does not provide a similar level of protection for an entire zone file. Zone Digests fill this space. This is helpful in the work in smearing the root zone (and potentially other zones) over a broader surface, allowing the DNS to scale under increasing growth pressure by using the so-called "Hyperlocal" approach to locally serving the root zone (RFC 7706).

DNS Server Cookies

The DNS has always represented about as good an attack vector as its possible to construct. The DNS protocol is universally supported, it uses UDP, and has the ability to penetrate most firewall filter configurations. The conventional response to attack in the DNS is to pull down the shutters and limit the UDP response rate. Unfortunately this crude response penalises legitimate DNS transactions as well as malicious ones. Cookies are one approach to mitigate this blanket response. A server can identify repeat clients and potentially give them "preferred" service. On the client cookies can be used to distinguish between expected responses from a DNS server and spoofed responses.

The Cookie mechanisms was published in 2016 in RFC 7873, and the work on specifying server cookies started in 2019, and the document in is the final stages of publication (https://datatracker.ietf.org/doc/draft-ietf-dnsop-server-cookies/)

YANG Types for DNS Classes and Resource Record Types

When SNMP gathered favour in the IETF it was the practice to define MIBs for just about everything. We've moved on and now YANG [RFC7950] has become a de facto standard as a language for modeling configuration and state data, as well as specifying management operations and asynchronous notifications. It is reasonable to expect that the approach based on utilizing such data models along with standard management protocols such as NETCONF and RESTCONF can be effectively used in DNS operations. using tools such as NECONF or RESTCONF to manage servers, resolvers and zone data.

Query Name Minimisation

RFC7816 is an experimental specification that proposes a change to DNS resolvers to withhold extraneous information when performing a name resolution. Specifically, it uses a truncated query name at each point in the name resolution process, only exposing the next level label to each server. In response to DNS privacy concerns this approach has been widely implemented, although there are some differences between observed resolver behaviour and the experimental specification in RFC7816. The Working Group is working on a standard specification (https://datatracker.ietf.org/doc/draft-ietf-dnsop-rfc7816bis/).

The changes in this include an upper bound of the number of successive additional single label queries, and a more general way to deal with names with a large label count. The document also considers the query type to use in these queries and suggests that while any query type can be used where the authority always lies below the zone cut, a good type to use is QTYPE=A because of the potential to blend into editing query streams. This matches current resolver behaviour.

NSEC and NSEC3 TTLs

The DNS is a surprising complex specification these days. There is an attempt to illustrate this complexity (https://emaillab.jp/wp/wp-content/uploads/2017/11/RFC-DNS.pdf), illustrating the document dependency state in November 2017. Obviously more have been added since then! It's no surprise that at times the documents contain misleading or contradictory guidance, and one area where this is problematic is in the caching of denial of existence records 9NSEC and NSEC3). As the working draft (https://datatracker.ietf.org/doc/draft-ietf-dnsop-nsec-ttl/) states, "Due to a combination of unfortunate wording in earlier documents, aggressive use of NSEC and NSEC3 records may deny names far beyond the intended lifetime of a denial. This document changes the definition of the NSEC and NSEC3 TTL to correct that situation. This document updates RFC 4034, RFC 4035, RFC 5155, and RFC 8198." This is not the first such clarification update, and definitely won't be the last in the DNS.

Service Binding in the DNS

Work continues to change the DNS from a simple name resolution mechanism to a generalized service rendezvous tool. The Service Binding Record (SVCB) provides both a name alias facility and connection parameters in the DNS. SVCB records allow a service to be provided from multiple alternative endpoints, each with associated parameters (such as transport protocol configuration and keys for encrypting the TLS ClientHello in TLS 1.3). While much of the motivation here lies in the world of hosted content and the requirements of Content Distribution Networks (CDN). It's still early days and it's not clear if this service record will be widely adopted. It has the potential to significantly improve the connection experience for clients and make improvements in the performance of hosted content for CDNs and allow greater diversity of hosting solutions for content publishers. It allows a richer set of control parameters to be used between content and service providers and content and service publishers, and equally allows the client to be provider with a set of connection parameters prior to the connection.
So, if common sense and a shared motivation to improve the speed and versatility of connection establishment drives adoption then this particular refinement of the DNS deserves to be adopted universally. However, it is a complex record, and our experience in populating the DNS with simple address records have not been all that inspiring. The DNSOP Working Group will be doing a Last Call review of this document (https://tools.ietf.org/html/draft-ietf-dnsop-svcb-https-04), and there is some expectation that it will head to IETF review in the coming months.

DNSSEC and Delegation-Only Zones

There are some zones in the DNS that do not contain terminal labels but have delegated zones. These "delegation only" zones have presented some ambiguities for implementation of QName minimization where NXDOMAIN responses were being generated in error. The more subtle issue here relates to DNSSEC, as delegation records in the parent zone are not DNSSEC-signed and the possibility exists for unauthorized manipulation of these unsigned records. While the root and most TLD zones are assumed to be exclusively delegation-only zones, there is currently no interoperable and automatable mechanism for auditing these zones to ensure they behave as a delegation-only zone. This specification (https://tools.ietf.org/html/draft-ietf-dnsop-delegation-only-02) defines a mechanism for delegation-only zone owners to create a DNSKEY that indicates it will only delegate the remainder of the DNS tree to lower-level zones. This allows for easier delegation policy verification and logging and auditing of DNS responses served by their infrastructure.

DNS and TCP

The DNS has a love/hate relationship with its transport protocols. UDP is incredibly efficient and cheap, but at the same time admits a large set of issues with abuse as well as clunky adaptation mechanisms to cope with large data bundles. TCP can cope with large transactions with ease and can act as a counter to abuse based on IP address manipulation in queries. So why doesn't the DNS shift to TCP? The web seems to cope with massive volumes of data transactions and it's all TCP (or QUIC). So why not shift the DNS to use TCP all the time? The major issue is cost. TCP is a higher overhead protocol, and while it is possible to think about a DNS infrastructure based entirely on TCP, there is a cost to be paid in terms of infrastructure and in terms of performance times. So, we use this rather clunky approach where the current default mode is to use UDP for DNS queries and responses and then shift to use TCP only when the response has the truncation flag set, triggering the client to re-query using TCP.

There has been a lot of mythology about the true capabilities of UDP in the public Internet. The introduction of EDNS(0) and a suggested default UDP buffer size setting of 4,096 octets gave the impression that UDP could be used for almost everything in the DNS and TCP was unnecessary. This was an act of bravado unsupported by data, and the issue fragmentation and packet loss has been very prominent in our thinking on this topic in recent years.

The issue with UDP is twofold: Firstly, UDP admits a family of amplification and reflection attacks because of the very nature of UDP. Like it or not RFC2827, published over 20 years ago is simply not widely deployed and source address spoofing is still possible in many, if not most, parts of the public Internet. Secondly, UDP uses IP fragmentation to carry large payloads and IP fragmentation is filtered in many parts of the Internet. Delivery failure of fragmented IP packets appears to impact around one quarter of all Internet users. If we want to continue to have the efficiency dividend of UDP in the DNS, then we have to do something about both of these issues. There has been much work on DNS attack mitigation, while the work on fragmentation is only gathering pace more recently.

The DNS Flag Day 2020 is one possible approach, and it relies on the use of the EDNS(0) buffer size setting to limit the maximal response size that will be used by UDP in DNS servers. A more general discussion of measures that can avoid UDP fragmentation, including reducing the number of name servers, changing the DNSSEC signing algorithm to an Elliptical Curve algorithm and use of the IP DONT FRAGMENT flag are all considered in a current Working Group document "Fragmentation Avoidance in DNS" (https://tools.ietf.org/html/draft-ietf-dnsop-avoid-fragmentation-00)

We've possibly over-reacted with this work, shifting the DNS to use UDP in a "fragmentation-safe" manner by avoiding sending fragmented UDP responses at all. This measure imposes a cost on all to improve the service for a minority of end points. Admittedly it's a big minority of around a quarter of users but nevertheless the solution of cutting over to TCP before the onset of UDP fragmentation has its own associated inefficiencies.

Obviously this is not a solved issue and the conversation will continue. This document (https://tools.ietf.org/html/draft-ietf-dnsop-dns-tcp-requirements-07) has rather modest goals, namely, to remind all that TCP is a part of the DNS and setting up firewall rules to block the use of port 53 over TCP is ill-advised.

DNS Delegation Records

Whenever information is duplicated in distributed systems there is always the problem of which source takes precedence when the sources vary from each other. Both the parent and the child list the delegation records for the child zone. RFC1034 provides the direction that "The administrators of both zones should insure that the NS and glue RRs which mark both sides of the cut are consistent and remain so." but fail to provide clear guidance e what to do when they differ. In the DNS the child NS records are authoritative. On the other hand, the parent has the ability to re-delegate the zone to a different child server, or even remove the delegation completely. So what NS set should a recursive resolver cache?

Evidently, there is wide variability in the way that DNS resolvers handle delegation records. Some resolvers prefer to cache the parent NS set, some prefer the child set, and for others, what they preferentially cache depends on the dynamic state of queries and responses they have processed. This document (https://tools.ietf.org/html/draft-ietf-dnsop-ns-revalidation-00) aims to bring more commonality and predictability by standardizing resolver behavior with respect to delegation records: "When following a referral response from an authoritative server to a child zone, DNS resolvers should explicitly query the authoritative NS RRset at the apex of the child zone and cache this in preference to the NS RRset on the parent side of the zone cut. Resolvers should also periodically revalidate the child delegation by re-querying the parent zone at the expiration of the TTL of the parent side NS RRset.

Terminology

What's the difference between a "Domain Name" and a "Label"? Or the difference between a "Stub Resolver" and a "Recursive Resolver". The DNS has invented its own terminology, and it's reached the point where a list of DNS terms and their definition is very useful. This is an ongoing effort and is intended to replace RFC 8499 (which, in turn, replaced RFC7719) as we invent new terms and subtly re-define existing terms. The current working draft of DNS terminology can be found at https://tools.ietf.org/html/draft-ietf-dnsop-rfc8499bis-01.

With so many documents being considered by this working group its not surprising that a few seem to have expired, presumably temporarily.

Recommendations for DNSSEC Resolvers Operators

This is an interesting informational document (https://tools.ietf.org/html/draft-ietf-dnsop-dnssec-validator-requirements-00) that attempts to clarify the responsibilities of DNS resolver operators when the resolver performs DNSSEC validation, as well as recommended some operational practices. The document appears to reflect upon some of the issues that were found in the 2018 root key roll in relation to the management of the DNSSEC trust anchor material, which is helpful.

Glue In DNS Referral Responses Is Not Optional

This document asserts that there is a widespread misbelief that all additional section records in a DNS response are optional. This document states that NS RRs are to be placed in the additional section of a DNS referral response, and Glue RRs added if the addresses are not available for the authoritative data or the cache. The subtle change added to the specification in this draft (https://tools.ietf.org/html/draft-ietf-dnsop-glue-is-not-optional-00) is to require setting the truncation bit if the glue RRs cannot fir into a UDP response.

Private Use TLD

This is the latest contribution in a long and messy story. I'll try to give an all-too brief personal synopsis here. A long time ago (late '90's) the IETF at the time came to the realization that it was singularly ill-

equipped to deal with the myriad of complex issues associated with determining the policies associated with the distribution of names, after a bruising episode with an IAB-sponsored International Ad Hoc Committee which was supposed to deal with the issues of access to generic top level domains.

The mess was passed over the fledgling ICANN body, and the IETF promised to take a step back from the issues of names and name policies and strictly deal in technical matter thereafter. ICANN headed down an economic rationalist path, and the generic name space was effectively monetized. A number of folk felt excluded by this shift but there was little they could do other than simply use as-of-yet unassigned DNS labels.

However, over time memories and commitments fade and in 2013 the IETF published RFC6762 and unilaterally assigned the tld ".local" for a "special use", based on Apple's multicast DNS technology. Soon after the TOR folk submitted to the DNSOP Working Group a similar case to have ".onion" declared a "special use" tld, and, predictably, a whole line of hopefuls formed behind them. The IETF subsequently backed from this approach (https://www.iab.org/documents/correspondence-reports-documents/2017-2/iab-statement-on-the-registration-of-special-use-names-in-the-arpa-domain/) and once more the only alternative to the ICANN process was simply self-allocation of a tld.

One line of thought in response was that rather than a "wild west" of ranchers and squatters clashing over names it might be helpful to follow the lead of the address system and declare a tld label as a distinguished "uber" tld where self-assignment of 2lds in the name space was not only permitted, but encouraged.

But where to make such a case? ICANN? Or the IETF? Or both? As it has turned out "both" has been the response, but with quite different approaches.

In the ICANN world the Security and Stability Advisory Committee (SSAC) came out with SAC113 (https://www.icann.org/en/system/files/files/sac-113-en.pdf) recommending that "ICANN reserves a string specifically for use as a private-use TLD for namespaces that are not part of the global DNS, but are meant to be resolved using the DNS protocol.".

At the same time a proposal was submitted to the DNSOP Working Group (https://tools.ietf.org/html/draft-ietf-dnsop-private-use-tld-00) proposing to use the "available" (technically, "user-assigned classification points) of the two-letter country code registry (maintained by ISO in the ISO3166-1 registry. In other words, proposing to squat on available space in the two-letter code space rather than reserving a particular TLD for private use.

Where to from here is a confusing question. It would seem that if the DNSOP Working Group were to submit this as an RFC then it would appear to be counter to previous IETF commitments to stay out of this area of name allocation policy and weaken the IETF's commitment to allow ICANN a free hand in the task of such name policies. On the other hand, the proposal for "free" tld in the ICANN space appears to be counter to the entire stance of ICANN in imposing a uniform policy framework on the assignment of top-level domain names, undermining its own policy process.

At least one cynic has observed: Watch this space. Bring popcorn!

NSEC3 Clarifications

NSEC3 records has always been one of the more challenging areas of the DNSSEC secure zone framework. The problem was to create verifiable assertions of non-existence in a zoner, and the NSEC approach was to order the labels in a zone and then sign across the "strides", asserting that no names between two adjacent labels existed. The domain name business is an odd business, and it turns out that the names that don't exist are probably more valuable commercially than those that do for many zones. If you can use NSEC to enumerate all the names that exist in a zone, then it's clear that all the other names don't!

How to prevent this form of zone enumeration? The approach used by NSEC 3 is to hash all the names in a zone using a SHA-1 hash and create striding non-existence records using the order of the hash of the names. Zone enumeration is harder.

Now SHA-1 is not the most robust of hash algorithms and there has been some concern that this was just making life harder for both zone signing and response validation while not imposing a major barrier to efforts to enumerate the zone. For this reason, NSEC3 has added a NSEC3PARAM to the zone file, specifying the number of hash iterations to apply, and a salt value, with the intention of making reverse engineering the hashed name set more challenging. While these attributes are commonly used, do they make if harder to reverse the hash, or do they simply add cost to the signer and the validator without any appreciable benefit.

This document, "Guidance for NSEC3 parameter settings" (https://tools.ietf.org/html/draft-hardaker-dnsop-nsec3-guidance-02) suggests that both measures are ineffectual in most cases and the integration value should be zero, and no salt value should be used.

Clarification of Clarifications

With so many specification documents on the DNS you'd think that it's all nailed down and there is little latitude for creative interpretation of the specification. This is not the case, and many aspects of the DNS would benefit from some further clarification. One of those topics is the "priming query" used by resolvers when booting to obtain the current root zone servers. This was already the subject of RFC 8109, published in 2017, but it seems that there is more to say about the priming query, and this draft (https://tools.ietf.org/html/draft-klh-dnsop-rfc8109bis-01) proposes some further clarification on the priming query.

Multi-Operator DNSSEC

These days the default mode of operation is to pass your domain to somebody to "host" it for you. You can take advantage of a DNS provider to create a server network for your domain across a distributed cloud of servers in anycast configuration with aggregate capacity in the service platform to withstand many forms of hostile attack. But how can you support multiple platform operators without going into contortions over key management? RFC8901 discussed a number of ways multiple operators can provide service to a signed zone. But there was at least one question that was not addressed in this document, namely the way to change providers in a secure manner. The document (https://tools.ietf.org/html/draft-wisser-dnssec-automation-00) outlines a number of scenarios for operators being added or leaving a zone and the way CDS/CDNSKEY and CSYNC records can be managed to effect a secure change in the operators of a zone.

DNS Errors

The repertoire of DNS error code is a rather sparse set of codes that are relatively unhelpful in terms of understanding the basic cause of the issue. This draft (https://datatracker.ietf.org/doc/draft-reddy-dnsop-error-page/?include_text=1) proposes adding a URI to the EDNS(0) option to further elaborate on the reasons for generating a DNS error code. Frankly I'm unconvinced about the value of this approach. Despite Domain Names being constructed on strings of human-friendly labels (with the exception of IDNs of course) the DNS is large a machine-to-machine protocol and exposing a URI with potential explanations of the error condition strikes me as a fundamentally poor idea!

Error Reporting in Reverse

Normally resolvers receive error codes from servers and are not expected to report errors back in the other direction. However, there are some circumstances where this is be best option, such as when an authoritative server is serving stale DNSSEC records. This document (https://tools.ietf.org/html/draft-arends-dns-error-reporting-00) describes a facility that can be used by validating recursive resolvers to report errors in an automated way.

To report an error, the reporting resolver encodes the error report in the QNAME of the reporting query. The reporting resolver builds this QNAME by concatenating the extended error code, the QTYPE and QNAME that resulted in failure, the label "_er", and the reporting agent domain. All this sounds like a re-run of RFC8145, an approach that was intended to permit a recursive resolver to signal its trust anchors to a root server. This approach was not that all that useful for the KSK roll in 2018 and it appears to me that this is not sufficiently different to make the approach any more useful in this content.

DPRIVE - DNS Privacy

The DPRIVE Working Group is working on the specification of technologies to improve the privacy of the DNS. Its work to date has included specifications of DNS over TLS, and DNS over DTLS. DNS over HTTPS was completed in the now concluded DOH Working Group.

DNS over QUIC

The distinction between DNS over TLS (DoT) and DNS over HTTPS (DoH) can be a subtle one. In both cases there is a TCP transport state between client and server and upon this TCP state is laid a TLS session. Within this session DNS packets (queries and responses) are passed. The distinction is that in the case of DoH an HTTP session is established, and the DNS query and response is framed as a HTTP GET or POST method (assuming that we are using HTTPS/2 and using a TCP/TLS substrate) There is a similar subtle distinction between DNS over QUIC (DoQ) and DoH where the version of HTTPS is HTTP/3 and the transport used is QUIC.

This work looks at the base transport case, layering DNS packets directly into a QUIC transport socket. As the DOQ document points out, the goals of the DoQ mapping are to provide the same DNS privacy channel protection as DoT, including the option for the client to authenticate the server, to provide a superior level of source address validation for DNS servers as compared to UDP, to avoid IP fragmentation and UDP issues. In this case the comparison is also DoH, but in this case its a comparison with HTTP/3 with QUIC selected. As with the existing DoT and DoH work its looking at the stub-to-recursive environment. In most other respects this is a conventional use of QUIC as an encrypted transport channel and passing DNS messages in that channel. The comparison between DoQ and DoT are pretty much the same as the comparison of TLS over TCP compared to and TLS over QUIC over UDP. There are opportunities to bypass head-of-line blocking when using QUIC and the relatively conservative setting of the QUIC packet size is intended to bypass path MTU issues.

In the same way that DoT uses a reserved TCP port 853, DoQ is intended to reserve UDP port 8853.

Oblivious DNS Over HTTPS

Channel encryption still admits one other party into the privacy circle: the recursive resolver at the other end of the query. While no others are able to eavesdrop on the conversation between a stub and recursive resolver, the recursive resolver knows the identity of the end system hosting the stub resolver and the queries coming from that end system. In terms of privacy that eliminates one class of risks but still admits a major weakness of being forced to include one other into the query stream.

There have been a number of approaches to improve this situation. A general approach is the onion network (TOR) which attempts to obscure the link between end point identity and the content payload. A rather devious approach was written up as Oblivious DNS (now expired draft https://tools.ietf.org/html/draft-annee-dprive-oblivious-dns-00), where the DNS query was itself encrypted and then passed using the normal DNS infrastructure to a target helper who deciphered the original query name, performed a recursive resolution for the name and then passed the result back using a session key provided by the original querier. The limitation was in the query name length.

This approach uses a conventional DNS query but uses two-layer encryption. The DNS query and the session key is encrypted using the public key of the target Oblivious DNS over HTTPS (ODoH) server. This encrypted payload is then passed as an opaque payload using HTTPS to a ODoH Proxy. The proxy's

task is to pass the encrypted query to the targets and similarly pass encrypted responses back to the client. As long as the Proxy and Target cannot conspire and share information then no party has knowledge of the end entity and the DNS queries (and responses) being made.

There is no clear consensus in the DPRIVE Working Group to adopt this document at the moment. It is an interesting experiment and there may be sufficient interest to pursue a path of an Experimental RFC at the moment.

Encryption between Recursives and Authoritatives

Encrypting the DNS query traffic between stub and recursive resolvers can address a large proportion of the DNS privacy agenda, as an onlooker can associate the identity of the end point with the DNS queries they make. The case to encrypt the cache-miss queries between a recursive resolver and authoritative servers has a different privacy "value". It is not possible to identify the end point of the query (unless Client Subnet has been turned on, which itself is a massive privacy leak). The costs are different in the recursive-to--authoritative scenario ass compared to stub-to-recursive. There is limited session reuse in so far as a recursive will query many authoritative servers so the overheads of setting an encrypted session may need to be amortized over a single query.

A draft has been authored on this scenario (https://tools.ietf.org/html/draft-ietf-dprive-opportunistic-adotq-00). There are some dubious assumptions about this scenario that prob ably would not stand further scrutiny at present, such as "Both parties understand that using encryption costs something but are willing to absorb the costs for the benefit of more Internet traffic being encrypted." The remainder of the document describes the process a recursive e resolver may use to discover if an authoritative server supports queries over DoT, and cache this capability for subsequent use. It is certainly a good demonstration that "this can be done", but that's not really the question for this particular technology. The question is more about incremental costs and benefits and on this crucial topic the draft is somewhat silent.

The opportunistic discovery method described in this draft is a "try it and see" and records the success or otherwise of an attempt to set up a DoT session with an authoritative server. Another draft (https://tools.ietf.org/html/draft-rescorla-dprive-adox-latest-00) advocated use of the SVCB method, using the DNS to convert the capability (and willingness) to support TLS connections.

Other DNS activity at IETF 110

That's not quite the full picture as there is further work in the Adaptive DNS Discovery working group (ADD) addressing issues of discovery and selection of DNS resolvers by DNS clients in a variety of networking environments, supporting both encrypted and unencrypted resolvers.

There is also the Extensions for Scalable DNS Service Discovery working group (DNSSD), looking at automation of service provisioning by using the DNS.

Looking further afield there is range Home Networking working group (homenet), looking at name resolution and service discovery in edge (home) networks.

That's a lot of DNS, and there is every expectation that the level of activity will continue at this level for IETF 111.

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