August 2023
Geoff Huston

DNSOP at IETF 117

This is part of a personal commentary on the meetings at the July 2023 meeting of the Internet Engineering Task Force (IETF 117). If you want to know what was presented and the recordings of the sessions, see the IETF 117 meeting agenda[1].

After the flurry of work in various aspects of DNS privacy, the IETF's agenda for DNS has shifted towards more maintenance and update. This does not mean that the volume of work has abated in any way, but it has dropped the more focussed stance of previous meetings.

Here's a run-down of the current work in the DNSOP Working group.

Just Gone out the Door!

In terms of recent work, RFC9432[2] has been published. This is a Standards Track specification of a mechanism termed *Catalogue Zones* which simplifies zone provisioning by orchestrating zones on secondary name servers from a single data source: the Catalog, which itself is a DNS zone consisting of a list of zones, specified as PTR records.

Going to the Exit

In the RFC Editor queue is the SVCB HTTPS specification draft-ietf-dnsop-svcb-https[3], which provides clients with complete instructions for access to an HTTPS service. This allows the DNS to translate the name of a service into the connection parameters that allow a client to access the service and do so in a single DNS transaction. Not only does this reduce a sequence of DNS service discovery queries into a single query and response, but it can also replace the connection capability negotiation phase, such as the capability of the server to support a connection using HTTP/3. It makes the entire process faster by reducing the number of exchanges to set up the service with the client.

The alt-TLD draft draft-ietf-dnsop-alt-tld[4] is also in the RFC Editor queue. This is another step in a rather convoluted dance between the IETF and ICANN over the governance of the DNS namespace. The establishment of the IANA Special Use Names registry and its initial population with non-DNS top level domain names was not without some elevation of eyebrows and the issues created by this IETF action were document in RFC8244[5]. This document proposes the reservation of the top-level label **"alt"** (short for "alternative") as a special-use domain name (RFC6761[6]). This top-level label can be used to signify that the name is not rooted in the global DNS, and that it should not be resolved using the DNS protocol. The intention is to provide an unmanaged space within the DNS name space that allows alternate approaches to name definition and resolution to be located, while attempting to avoid name conflicts with the regular DNS context.

---

[1] https://datatracker.ietf.org/meeting/117/agenda
[2] https://www.rfc-editor.org/rfc/rfc9432.html
[3] https://datatracker.ietf.org/doc/draft-ietf-dnsop-svcb-https/
[4] https://datatracker.ietf.org/doc/draft-ietf-dnsop-alt-tld/
[5] https://www.rfc-editor.org/rfc/rfc8244.html
[6] https://www.rfc-editor.org/rfc/rfc6761.html

The consistent handling of DNS glue records has been subject of diverse interpretations by DNS server implementations. The RFC editor queue contains a draft that attempts to clarify the role of such glue records draft-ietf-dnsop-glue-is-not-optional[7], namely that when a name server generates a referral response, it must include all available glue records for in-domain name servers in the additional section, or must set the truncated response bit (TC=1) if constrained by UDP message size. This clarification is intended to improve the resilience of the DNS, because, as the document notes: "At the time of this writing, the behaviour of most DNS server implementations is to set the TC flag only if none of the available glue records fit in a response over UDP transport."

Held up!

A number of documents are in some form of pause mode, awaiting some further action before proceeding, or simply allowing a moment to reflect if the working group should proceed to publication of the document as an RFC.

One such paused document is DNSSEC validator requirements: draft-ietf-dnsop-dnssec-validator-requirements[8] which does not appear to introduce any new requirements per se, but is more along the lines of collecting the various constraints and recommendations from the existing DNSSEC specifications and publishing them in a single document. Such documents are useful in terms of collating the information that is spread across many documents (this particular draft has 20 normative references to other RFCs and drafts!). On the other hand, such digest documents are necessarily a snapshot in time, and they quickly become just one more document that needs to be updated every time any of the other specification documents that touch upon DNSSEC validation is revised.

Another document, on the use of the GOST 2012 Signature Algorithm, draft-ietf-dnsop-rfc5933-bis[9], updating RFC5933 and RFC8624 is also paused to allow some time to reflect on where to go with this specification. The normative inclusion of national cryptographic algorithms in international standards has not been without its elements of controversy from time to time, and in this case the proposal to publish the specification of the use of GOST in DNSSEC has now been passed to the Independent Submissions Editor.

There is also the specification of caching resolution failures, draft-ietf-dnsop-caching-resolution-failures[10] which is paused in the IESG queue. The early DNS standards required resolvers to cache the non-existence of the requested data (where the requested type does not exist, or the name itself does not exist), but caching of the failure of the resolution process itself was an optional requirement in RFC2308. This has impacted a number of operational incidents, such as when the Facebook.com servers failed in October 2021, when query traffic on the **.com/.net** server infrastructure was reported to have increased from 7,000 to 900,000 queries per second. The answer to this hundred-fold increase in query rates in the face of certain service failures is to cache resolution failure, to allow the recursive resolvers to absorb the stub queries in times of failure and not pass them inward to the failed DNS authoritative servers. The proposed cache timer settings are a compromise in that longer cache times may needlessly protract the failure that resulted from a server misconfiguration, while shorter times may still lead to large query volumes during times to failure. This draft specifies a minimum of 1 second, suggests a minimum initial cache of 5 seconds and maximum cache lifetime of 5 minutes. The document is held in the IESG awaiting more operational feedback on these proposed initial timer settings, as perhaps a shorter initial cache value may be a better match to transient resolution failures, allowing a faster recovery of service without exacerbating the failure with excessive queries. It's likely that this is a temporary pause, and the document will be passed to the IESG in the very near future.

[7] https://datatracker.ietf.org/doc/html/draft-ietf-dnsop-glue-is-not-optional

[8] https://datatracker.ietf.org/doc/html/draft-ietf-dnsop-dnssec-validator-requirements

[9] https://datatracker.ietf.org/doc/draft-ietf-dnsop-rfc5933-bis/

[10] https://datatracker.ietf.org/doc/draft-ietf-dnsop-caching-resolution-failures/

Going Out!

All the DNSOP documents mentioned so far are in the latter stages of the publication process one way or another. There are a set of documents that are in their last stages of Working Group review, in Working Group Last Call.

There is the latest revision of the DNS terminology document, draft-ietf-dnsop-rfc8499bis[11] which appears to be struck on the term "lame delegation", both because it's not clear what conditions are encompassed by this term, and secondly due to the inappropriate use of the word *lame* in this context.

There is the "zone version" EDNS query option specification draft-ietf-dnsop-zoneversion[12], which directs a DNS authoritative server to add an EDNS option in the answer to query with a token field representing the version of the zone file associated with the answered Resource Record, such as the SOA serial field in zones when this number corresponds to the zone version. Of course, this interpretation of the SOA serial field reflects a notion of a zone as being defined in a single file with distinct batched update operations applied to it that cause an update of the serial number. It's likely that very large zones, or dynamically computed zones, or hosted zones do not necessarily have such a clear underlying notion of batched updates to a zone file, so the interpretation of the SOA value reflecting distinct "generations" of a master zone file is somewhat unclear. If the DNS continues its shift to a more dynamic signalling protocol, then the value of this additional data field will become somewhat questionable in the coming years, as it reflects a more traditional "zone snapshot" view of zone publication in the DNS. Nevertheless, the draft has passed a Working Group Last Call and is awaiting a write up to send it on its way to the IESG.

The DNS Error Reporting draft draft-ietf-dnsop-dns-error-reporting[13] describes a lightweight reporting mechanism that provides the operator of an authoritative server with reports on DNS resource records that fail to resolve or validate. The reports are based on extended DNS errors as described in RFC8914. The server puts in a report channel server in its responses, and the reporting client reports an error via a DNS query for a TXT record. The query name used for such reports is a concatenation of the queried name, the query type and the error code, prepended to the reporting DNS name. It reflects the increasing role specialising in the DNS where responsibilities for various aspects of a zone's DNS deployment are split across several entities, so that the channel to report DNS failures may not be the same point as the SOA contact, for example. This document is on its way to the IESG.

The issue of DNS over UDP transport and the related issue of IP fragmentation has been a vexed one for some years. The DNS is an efficient protocol due to its use of UDP, allowing the use of stateless servers that treat each incoming DNS query in a UDP packet as a separate transaction. But with the increasing amount of data being placed into DNS records we are seeing the size of DNS responses increasing. It's not just DNSSEC signatures, although that is a big component of this added volume, but all the other additional fields, qualifiers and flags. EDNS has proved to be the kitchen sink of the DNS, and we are working on figuring out if there is any limit to the variety and volume of information that we can stuff into the EDNS bucket! However, we encounter some issues with this approach. In general the Internet operates on a 1,500 octet maximum packet size, and any attempt to use a larger IP packet will encounter fragmentation. Fragmentation compromises the robustness and the integrity of DNS over UDP. Dropping of fragmented IP packets is a common firewall practice, causing a DNS client to timeout awaiting a response. Path MTU mismatch is a related issue, where in IPv6 the ICMP diagnostic message needs to be sent back to the server and the client needs to timeout and resend the DNS query. There is also the issue of potential substitution in fragmented responses, resulting in DNS cache poisoning from off-path attackers. Maybe we should just move all DNS to TCP. Right? Well, not so fast! DNS over TCP is slower due to the initial TCP handshake, and it will impose a higher load on the server in terms of storing active TCP session control blocks. Some estimates of the difference in load indicate that we would

[11] https://datatracker.ietf.org/doc/draft-ietf-dnsop-rfc8499bis/

[12] https://datatracker.ietf.org/doc/draft-ietf-dnsop-zoneversion/

[13] https://datatracker.ietf.org/doc/draft-ietf-dnsop-dns-error-reporting/

need the operate DNS infrastructure at some three times the UDP-based capacity if the DNS was operated entirely in TCP. It's not surprising that we continue to favour UDP if the response is small enough to fit into a single UDP packet without fragmentation. Larger DNS responses will be sent in a single UDP response with the Truncated Flag set up, causing the client to re-query over TCP. Yes, this is slower than UDP, but more robust for the small proportion of DNS answers that are just too big to be sent in UDP without fragmentation. The advice to avoid fragmentation in the DNS, as provided in draft-ietf-dnsop-avoid-fragmentation[14], is not a terribly long and detailed document. The point is simple: servers really should not fragment DNS over UDP responses! The draft is currently in Working Group Last Call.

Another DNSOP draft in Working Group Last Call is draft-ietf-dnsop-domain-verification-techniques[15]. It is a common question in the provision of DNS infrastructure to ask: "is this really your domain?". An automated way to answer this is to ask the querier to insert a particular record into the DNS. The common approach is to use the TXT record type and the outcome is a rapid bloating of the TXT fields at the zone apex point, where there may be many TXT records for each of the different zone validation functions. This overloading of the TXT record is a bit of brute-force solution, and this draft considers alternative approaches, including some part of the challenge token value in the query name as well as the value in the response. It's a useful document for anyone seeking to perform DNS validation with a survey of existing techniques in an appendix as well as some general suggestions about how to structure such domain name challenges for validation.

Ready?

We can now more on to the next set of documents, namely those in active consideration in the Working Group that are likely to be ready for a Working Group Last Call.

There is this theory that what is holding up further use of DNSSEC to sign DNS zones is the difficulty of the processes around the management of DNSSEC credentials, and if we could devise some automated procedures to managed DNSSEC credentials, then DNSSEC deployment would blossom. These automated mechanisms are described in RFC7344 and RFC8078, where the child can publish the hash of their entry point zone key in their own zone, and the parent can automatically pick this up by polling the zone, validate the record, using DNSSEC of course, and then add this record into their zone file as a DS record. All well and good, but can we use a similar method for a delegated zone to transition from unsigned to signed? An enrolment process was described in RFC8078, but it's proposed to deprecate that process, replacing it with a mutually trusted third party, as described in draft-ietf-dnsop-dnssec-bootstrapping[16]. To bootstrap into a signed status the first instance of the child CDS/CDNSKEY records are placed into this third-party zone and signed within the context of that zone. The parent server can then lift these records from this trusted domain. While it all seems to be quite convoluted, it reflects on increasing levels of role specialisation in the DNS environment, where zone publication, authentication enrolment, name registration are not only distinct functions but also can be performed by distinct actors.

The DNS-Based Authentication of Named Entities (DANE) specification (RFC7671) explains how clients locate the TLSA record for a service of interest, starting with knowledge of the service's hostname, transport, and port number. The document draft-ietf-dnsop-svcb-dane[17] describes the interaction of DANE with indirection via Service Binding, using the SVCB and HTTPS resource records. It also explains how to use DANE with new TLS-based transports such as QUIC. In some ways this is a simple update to bring QUIC, DTLS and SCTP services into the realm of service bindings in the DNS. However, this is an important level of abstraction in the name system to allow secure mappings to be

[14] https://datatracker.ietf.org/doc/draft-ietf-dnsop-avoid-fragmentation
[15] https://datatracker.ietf.org/doc/draft-ietf-dnsop-domain-verification-techniques/
[16] https://datatracker.ietf.org/doc/draft-ietf-dnsop-dnssec-bootstrapping/
[17] https://datatracker.ietf.org/doc/draft-ietf-dnsop-svcb-dane/

formed between the service name and the provider-managed names and credentials using the DNS as mapping substrate and allows the public key value associated with domain name control to be made known to a client without using the Web PKI and its diverse trust model. When using the DNS over an encrypted channel the DANE exchange can happen in an entirely private manner as well. It appears that this specification is largely done in the Working Group and is ready for a Last Call.

Active!

The DNS is a loosely coherent system, and to support operational resilience there are many levels of redundancy. There is typically more than one authoritative server for a domain. This may be the outcome of the use of multiple DNS publication platforms, so that these servers may be operated by more than one entity. The obvious question to ask is what should a client do if there are multiple different values for the same CDS/CDNSKEY records? The solution proposed in draft-ietf-dnsop-cds-consistency[18] is to require that all authoritative servers be queried, and all responses must match before accepting these values and installing new DS/DNSKEY records. The good news is that it is not necessary to poll every authoritative name server all of the time for changes to CDS/CDNSKEY, but once a change in the CDS/CDNSKEY values is detected then a consistency check should be performed across all the name servers to ensure that they all are providing the same values.

I must admit that I thought the original RFCs on CDS and CDNSKEY as a means of automated provisioning of the DS record in the parent zone was functionally adequate but the number of drafts seeking to clarify and elaborate on certain aspects of this functionality mean that either my impressions were incorrect and there were missing parts in the original specification, or many folk just can't resist the temptation to twiddle the bits. For me the specification draft-ietf-dnsop-dnssec-automation[19] strikes me as heading into a corner space. It describes the situation of using CDS/CDNSKEY records in the situation where there are not just multiple publishers of a zone, but also multiple signers of a DNS zone each with their own keys. It also strikes me as slightly contradictory that the reason to use multiple signers and multiple zones publishers is to create a more resilient environment, yet the advice in this draft appears to paint a different picture: “Every step of the multi-signer algorithms has to be carefully executed at the right time and date. Any failure could resolve in the loss of resolution for the domain." It's hard to tell if that’s just a warning or a confident prediction, but I think I'll opt for the latter!

The DNS has one point of replicated data, where the parent and the child both list the collection of name servers that are authoritative for the child zone. The DNS states that the child's NS records are authoritative, but when performing name resolution, the top-down discovery process necessarily forces a resolver to follow the NS referral records as provided by a parent server. If these two sets of NS records are in exact agreement, then this is not an issue, but when they differ there are a set of consequent issues about resolver behaviours and modes of breakages in the DNS. As draft-ietf-dnsop-ns-revalidation[20] notes, there is significant variation in resolver behaviour over the processing of delegation record in such cases. The document proposes a standard behaviour mode in a hope that this will bring a greater level of predictability to this aspect of DNS name resolution. The delegation records at the apex of the child zone should be cached by a resolver in preference to the records provided by the parent. Secondly, the delegation itself should be revalidated by querying the parent servers upon expiration of the time-to-live (TTL) of these NS records. I would put this behaviour down to common sense about the nature of authority in delegation, but these days we seem to be explicitly documenting every aspect of DNS behaviour in some considerable detail.

The retrofitting of DNSSEC to the DNS has raised some interesting general questions about how to update the DNS. The desire is to introduce updates to the DNS in a backward compatible mode, so that if a server that is aware of a new feature was being queried by a non-aware resolver, then it should still be able to understand the response it will receive. This desire for backward compatibility has had the

[18] https://datatracker.ietf.org/doc/draft-ietf-dnsop-cds-consistency/

[19] https://datatracker.ietf.org/doc/draft-ietf-dnsop-dnssec-automation/

[20] https://datatracker.ietf.org/doc/draft-ietf-dnsop-ns-revalidation/

result of overloading some DNS response codes (RCODEs). RCODE 2, SERVFAIL, is used when the server has been unable to process the query, and when a DNSSEC-validating resolver is unable to successfully validate a DNSSEC-signed response if returns RCODE 2. Strictly speaking, this response is accurate, in that if a DNSSEC-validating resolver cannot successfully validate a response it will withhold the response and provide this error code to indicate that it cannot provide a response. The pragmatic outcome however is less useful. A SERVFAIL response says to the resolver: "Maybe you might want to query another server", while a DNSSEC validation failure is a stronger message about resolution failure where querying a different server will not necessarily provide a different response. There is a problem in this entire approach. RFC8914 was intended to provide extensibility to DNS errors to provide this information, but as far as the DNS is concerned, there is always more to say! The document draft-ietf-dnsop-structured-dns-error[21] proposes to extend these more detailed error responses into responses that explains if the application of DNS Filtering has been applied to the DNS response.

Much in the technology space is still being worked upon, so standard specifications inevitably change. The RFC document series does not do change easily. Yes, you can get errata notes attached to an RFC, but other forms of updates require an updated RFC, which seems like a lot of work for some relatively small-scale updates to the specification. This is the case for the DNS priming query. First described in RFC1034, and subsequently refined by material in RFCs 3226, 4033, 5452, and 6891, the description of the priming query was pulled together in a single RFC, 8109 in 2017. Evidently it's time to revise it, and draft-ietf-dnsop-rfc8109bis[22] proposes changes to this RFC, none of which seem to me to be substantive in terms of the format of the query or the response!

Some years ago, Cloudflare described a technique[23] that made denial of existence responses for DNSSEC-signed zones smaller and faster. At the heart of the technique was to substitute a signed NODATA response for the signed NXDOMAIN response. The practical outcome of the altered response is the same, namely that there is no answer for this query, but in the case of NODATA the DNS response is a lot smaller, and it's a lot faster for an online front-end signer to generate on the fly without needing to reference a zone file. This technique was described in draft-valsorda-dnsop-black-lies[24] in 2016. The draft has been revived as draft-ietf-dnsop-compact-denial-of-existence[25]. There is one aspect of this revived specification, namely the introduction of a new synthetic Resource Record type, proposed to be called NXNAME, to signal a non-existent name, to allow a client to distinguish this case from an empty non-terminal name (ENT). Some implementations have chosen the other option, using a resource record type for ENTs. A possible, even likely, outcome is to define both Resource Record types. Deciding is hard.

Knocking at the Door!

This collection of drafts represents a considerable body of effort on the part of the authors, reviewers and of course the all the folk involved in the process of IETF standard production. But that's not enough DNS activity. There is still a seat of documents queued up for working group consideration. It was reported at this meeting that another five drafts are candidates for adoption.

The DNSSEC framework is rooted with a single key, the trust anchor, so its handling is critical. RFC7958, published in 2016, describes the formats and distribution methods for that critical key. Unfortunately, there was an error in the specification, and while taking the opportunity to address this error the document draft-bash-rfc7958bis[26] also proposes some more minor changes in the publication format. The process to perform the next roll of the trust anchor key is now starting up, so in terms of having the specification ready prior to the commencement of the process it would be good to have the specification

[21] https://datatracker.ietf.org/doc/draft-ietf-dnsop-structured-dns-error/
[22] https://datatracker.ietf.org/doc/draft-ietf-dnsop-rfc8109bis/
[23] https://blog.cloudflare.com/black-lies/
[24] https://datatracker.ietf.org/doc/html/draft-valsorda-dnsop-black-lies
[25] https://datatracker.ietf.org/doc/draft-ietf-dnsop-compact-denial-of-existence/
[26] https://datatracker.ietf.org/doc/html/draft-bash-rfc7958bis

completed in the coming months. That may be asking a lot from an already heavily laden document review process within the DNSOP Working Group.

The early specifications of the DNS protocol left space in the specification for future needs and evolving requirements. One of these was the option to place multiple queries in a single packet. RFC1035 defined a 16-bit field for the count of queries in a DNS request. The rest of the DNS specification assumes a singular query with a singular response, and no requirement has emerged to change this singular mode of operation of the query/response protocol. So why not reclaim these unused 15 bits and allow for either no query or just one query? The document draft-bellis-dnsop-qdcount-is-one[27] proposes exactly that change to the base DNS specification. If this seems a little pre-emptive after a mere thirty years, then there is always the expired draft draft-bellis-dnsext-multi-qtypes-07[28] that looks at how multiple queries might be supported, with the liberal assistance of EDNS to enable it!

Early efforts to improve the scaling and resilience of the DNS relied in explicit replication, with multiple authoritative servers being used to serve a DNS zone. if one server was unresponsive there were always more! There are limits to this form of scaling and iterating through an extensive list of name servers would push any user to the limits of their patience. In response, we turned to anycast, and use a constellation of servers all responding to queries on the same IP service address. The routing system is used to steer the querier to the closest instance of the anycast service constellation. This approach has far more generous scaling properties, but the cost of operating such a large-scale platform is typically defrayed by serving a collection of zones, not just one (except of course the root zone). Operating a multi-zone anycast cluster has its own challenges in signalling. It is preferable if a node is fully operational and able to answer queries before it joins the anycast constellation and is removed when the server has crashed. Other operational signals are also useful, such as load levels, zone status, DNSSEC key status and errors in DNS resolution. So far, each anycast operator has developed their own tools to manage anycast service constellations, but the interaction between the host and the name server process has meant that name server vendors are being requested to implement a variety of signals to interact with the platform management system. The document draft-grubto-dnsop-dns-out-of-protocol-signalling[29] serves as a requirement document, intended to guide the work on a standard signalling mechanism that will best suit the various use cases. Part of that effort is to enumerate a list of conditions with potential actions as well as assembling an inventory of existing signalling mechanisms in use today.

The DNS infrastructure is critical to the operation of Internet services, and the implications of service failure can be widespread and serious. Failures are painful to endure, but at the same time they can be useful learning exercise, and the outage of DNSSEC parts of the **`.se`** DNS zone in February 2022 has been a good example of this. The zone admin's efforts to critically review and redesign their zone production environment to ensure that the published zone is verified before publication is described as a process framework in draft-johani-tld-zone-pipeline[30]. Their design principles are a good guide to the design of robust and maintainable systems:

1) Data transport between steps should use open, standardised protocols (eg. DNS AXFR/IXFR).
2) The critical path for zone data follows must be via well-reviewed standard software (i.e., no custom software).
3) Errors in the input must not be able to negatively affect the ongoing publication of (an earlier version of) the zone.

Their rationale for bring this work to this operationally focussed working group are also useful as a reminder of why such working groups exist in the IETF: "While we've tried to make it as generic as possible it seems likely that we've failed here and there. It would be great with input from others to make this a more collaborative effort resulting in wider applicability. It is in need of much more third-party review. Therefore, we would like to see this document adopted by the DNSOP WG." This makes a whole lot of sense to me!

---

[27] https://datatracker.ietf.org/doc/draft-bellis-dnsop-qdcount-is-one/

[28] https://datatracker.ietf.org/doc/html/draft-bellis-dnsext-multi-qtypes

[29] https://datatracker.ietf.org/doc/draft-grubto-dnsop-dns-out-of-protocol-signalling/

[30] https://datatracker.ietf.org/doc/html/draft-johani-tld-zone-pipeline

There are some lessons in data distribution systems that the IETF appears not to want to learn. Polling, where the clients periodically check with the server to see of the data has changed and a new version should be loaded suffers from scaling pressures. An increasing population of clients generates more pressure on servers, and any client's desire to make the system more responsive to changes implies an increase the polling intensity by that client. In isolated scenarios with a limited set of clients and servers then this is a readily contained issue, but in a potentially unbounded context the decision to use on-demand polling is an unwise design decision. We are encountering this in considering how to automate DNSSEC provisioning of CDS/CDNSKEY records. The current approach, polling, has these potential scaling issues, and while the level of zone signing is small enough today, and while we are tolerant of daily update cycles, polling is a cheap and simple approach. But what if we want the system to respond to signalled changes within the same second? Or if the parent zone is populated with a few hundred million signed child zones? It's challenging to see how polling techniques can scale indefinitely in both time and increased responsiveness. A potential solution is to only poll when the child signals the parent (or its agent) that a change has occurred. And we have already devised such a signalling mechanism for communication between the primary authoritative name server and the set of secondary services, namely DNS Notify (RFC1996[31]). Can we generalise this Notify mechanism for use in other context where clients in the DNS system need to inform servers of a change in status? Well, yes, and draft-thomassen-dnsop-generalized-dns-notify[32] describes one such approach to such direct notifications.

DNSSEC was specified in a manner that left the choice of which crypto algorithm to use with the zone administrator, as a specific choice of algorithm was not locked into the standard DNSSEC specification. This makes a lot of sense as cryptographic algorithms are a moving target. With generally increasing computation capability an existing algorithm or choice of key size can be compromised over time. It appears that we are facing this situation for a whole class of algorithms if quantum computing passes across a threshold from limited use experimental platform to production systems. What happens if you want to change the algorithm used to sign a zone? For example, you might want to move off RSA-2048 (based on prime number factorisation) and shift to ECDSA P-256 (based on elliptic curve point multiplication) to reduce the size of DNSSEC keys and signatures. But our common forms of DNSSEC use does not make this easy, and during an algorithm change it's common to see every signed entry have multiple RRSIG records, each signed by each algorithm. This allows a client to use just one of the keys (and algorithm) to validate the signed entries in the zone. This raises some issues with multi-signing setups where signers use different algorithms as well as different key values, so adding multiple RRSIG entries to each record in the zone requires a further level of coordination between these multiple signers. Another approach, draft-huque-dnsop-multi-alg-rules[33], attempts relax that requirement. This proposal introduces a taxonomy of algorithms into "universal" for secure, widely adopted algorithms, "insecure" for algorithms that are considered to be no longer suitable for general use, and "insufficient" for algorithms that are not widely adopted. Using this taxonomy, it's proposed that a zone signer may elect to publish RRSIGs that use a "universal" algorithm and not to publish signatures using “insecure” or “insufficient” algorithms. The claimed benefit from this approach is that multiple algorithms can be used in a zone without dual signing as long as one of them is a “universal” algorithm, in a manner analogous to single-algorithm ZSK key rolls.

DNS at the IETF

Enumerating the set of active topics in the DNS work at the IETF is a major task, as this attempt to provide any informative snapshot of this work illustrates. The presentations, recording and video of the Working Group meeting can be found at the meeting agenda[34].

[31] https://www.rfc-editor.org/rfc/rfc1996.html

[32] https://datatracker.ietf.org/doc/draft-thomassen-dnsop-generalized-dns-notify/

[33] https://datatracker.ietf.org/doc/draft-huque-dnsop-multi-alg-rules/

[34] https://datatracker.ietf.org/meeting/117/agenda

Even this extensive summary of activity is just one part of the larger picture of current IETF work on the DNS. I've not even tried to cover the current state of the Adaptive DNS Discovery (**ADD**), working on discovery and selection of DNS resolvers by DNS clients in a variety of networking environments, DNS Privacy (**DPRIVE**), which is currently working on recursive-to-authoritative privacy topics, and Extensions for Scalable DNS Service Discovery (**DNSSD**), working on DNS service discovery in multi-link subnets. I think I've already exhausted your patience (and mine) with this rundown of current DNSOP Working Group activities!

Disclaimer

The above views do not necessarily represent the views or positions 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