The ISP Column
A column on various things Internet

April 2025
Geoff Huston

DNS at IETF 122

One of the more active areas of activity at the IETF falls under the remit of the DNS Operations Working Group. It's an operational Working Group rather than a protocol development working group, but nevertheless the DNS is a very rich topic area with much to talk about. There is a view that this level of attention of unwarranted, as the DNS is just another application layered upon a common Internet substrate. However, I think that this perspective is missing the point that the DNS is an intrinsic component of the Internet's infrastructure and given the level to which the IP address world has been fractured over the years (with Network Address Translators, IPv6, and a panopoly of Dual Stack transition mechanisms), what's left to define the Internet is the DNS. These days the Internet is to all intents and purposes a name-based network. Our framework for supporting rendezvous, authenticity, and encryption is based on this name infrastructure. Little wonder that the DNS has engendered so much attention in the IETF.

1. Document Roundup

In line with this central role of the DNS in the Internet's ecosystem, DNSOP is a very active Working Group and since IETF 121, three Working Group drafts have been published as RFCs. They are:

- **RFC 9609**, on initializing a DNS resolver with Priming Queries. *Priming* is the act of loading the list of root zone servers from a local configuration that lists some or all of the IP addresses of some or all of those root servers. In priming, a recursive resolver starts with no cached information about the root servers, and it finishes with a full list of their names and addresses in its cache.

- **RFC 9715**, an informational document about techniques that can be used in DNS deployments that avoid the occurrence of IP-level packet fragmentation in the DNS.

- **RFC 9718**, the format used to publish the DNSSEC trust anchor (the Root Zone Key-Signing Key).

There is one draft in the RFC Editor Queue:

- **draft-ietf-dnsop-compact-denial-of-existence**, which describes a way to convey that a name in a DNSSEC-signed zone does not exist, but in a manner that uses far smaller responses than conventionally signed NXDOMAIN response. The technique relies on the use of so-called front-end signers in authoritative servers, which can tailor a response to a particular query, rather than relying on the prepared negative spanning NSEC records.

There are a further five drafts with the IESG:

- **draft-ietf-dnsop-generalized-notify**. This is a useful extension to the DNS Notify mechanism to allow triggering other types of actions via the DNS that were previously lacking a trigger mechanism. This work was motivated by the observation that the management of CDS/CDNSKEY records was undertaken by constant polling on the part of the parent to detect when the CDS record changes. The generalisation of the notify mechanism allows the child domain to notify the parent when a change has been made, eliminating the need for the parent to perform constant polling to detect if the information has changed.

- **draft-ietf-dnsop-must-not-ecc-gost**. As the title suggests. These days the adoption (and retirement) of crypto algorithms supported by DNSSEC implementations is governed by standards actions. So this is a retirement document for the ECC GOST algorithm.

- **draft-ietf-dnsop-must-not-sha1**. See above, applied to SHA-1!

- **draft-ietf-dnsop-rfc8624-bis**. This is a housekeeping document to update the processes of administering these DNSSEC registries.

- **draft-ietf-dnsop-structured-dns-error**. The DNS is not exactly a recent protocol and now has an extensive based of deployment. Which means that the process of changing it is exceptionally challenging. Can you add a new DNS response code to the DNS? The act of adding a new number to the IANA-maintained list of defined response codes is easy. Updating all the implementations of DNS to recognise the new code and react correctly is a completely different and far more challenging task. This is the reason why the DNSSEC definition directed resolvers to return the existing error code SERVFAIL rather than return a new code of VALIDATION FAILURE, for example. RFC 8914, published in 2020 allowed an extensible method to return additional information about the cause of DNS errors. Though created primarily to extend SERVFAIL to provide additional information about the cause of DNS and DNSSEC failures, the Extended DNS Errors option defined in this document allows all response types to contain extended error information. But is it enough? This draft This document updates RFC 8914 by signaling client support for structuring the EXTRA-TEXT field of the Extended DNS Error to provide details on the DNS filtering. Such details can be parsed by the client and displayed, logged, or used for other purposes. I am really not sure about the value of this extension. The basic DNS resolution protocol is one persistent querying across all aothoritative nameservers until you get an authoritative response, positive or negative! I can't see the value in attaching a commentary to a negative DNS response, but I might be a traditionalist outlier!

Of the current work items in DNSOP a number of proposals are in their final steps of working group consideration and refinement. These are:

- **draft-ietf-dnsop-dnssec-automation**. The DNS achieves resilience through duplication of roles. It is common for a DNS zone to be served by two or more nameservers, and in the quest to remove various single points of potential failure it is increasingly common to use multiple service providers to operate these various nameservers. DNSSEC adds some complexity to this situation, particularly when each service provider is using its own Zone Signing Key to sign the contents of the zone. This draft outlines the procedures to automate the setup, operations, and decommissioning of Multi-Signer DNSSEC configurations.

- **draft-ietf-dnsop-ns-revalidation**. There is one instance where DNS information is replicated, namely the NS delegation record. This data, containing the set of nameservers that are authoritative for this zone, is held both in the parent zone and the child zone. The child zone is intended to be the point of authority for this information, but it is the data in the parent zone that is used during the top-domain name resolution process. As long as both records are the same, this is not an issue, but when they differ it's the child zone's data that is the "right" data. However, its common to see DNS resolvers locally cache the parent-side data that they collect during name resolution. This proposal recommends that resolvers explicitly query the child zone's nameservers and cache this information in preference to the data obtained from the parent. If the zone is DNSSEC-signed then the child-side NS records are signed, and if the parent-side NS records are attempting to misdirect the resolver, this misdirection can be detected through validation failure (the parent-side DS record, signed by the parent, is the indicator of a signed child zone). On the other hand, if the zone is unsigned, or the resolver is not performing DNSSEC validation, then a parent side misdirection will lead to the "wrong" child zone, and frankly it's no more credible than the parent zone values. I am not convinced of the need to perform this operation for unsigned zones, and for signed zones it's the ultimate response that matters, not how the resolver got there.

2. Current Work

CDS Consistency - draft-ietf-dnsop-cds-consistency

One of the active areas of development in the world of DNSSEC has been in the passing of information from the domain administrator (the "child" domain) to the domain that holds the delegation (the "parent" domain). This is typically performed today with various provisioning steps, using Provisioning Protocols (Such as EPP - RFC 5730). However, the observation is that when a domain is DNSSEC signed, then the authenticity of the information can be validated by performing a validation of the signed resource record.

For the case of DS records, RFC 7344 provides automation by allowing the child to publish CDS and/or CDNSKEY records holding the prospective DS parameters which the parent can ingest. Similarly, RFC 7477 specifies CSYNC records to indicate a desired update of the delegation's NS (and glue) records. Parent-side entities (e.g. Registries, Registrars) can query these records from the child and, after validation, use them to update the parent-side RRsets of the delegation.

The issue is that the DNS is a loosely coupled system and when a domain administrator updates the content of a zone, it may take some time before all the authoritative nameservers for that zone have the updated information. During his period of update a client will receive different responses from different authoritative nameservers. This proposed refinement to the CDS specification, stipulates that when performing such queries, parent-side entities must ensure that updates triggered via CDS/CDNSKEY and CSYNC records are consistent across all the child's reachable authoritative nameservers, before taking any action based on these records.

Domain Verification Techniques - draft-ietf-dnsop-domain-verification-techniques

Many service applications on the Internet need to verify ownership or control of a domain. The general term for this process is "Domain Control Validation", and while it can be done using a variety of methods such as email, or HTTP/HTTPS, this document focuses only on DNS-based methods. This typically involves the Service Provider requesting a DNS record with a specific format and content to be visible in the domain that is to be verified. There is wide variation in the details of these methods today, and some practices can encounter issues. This document provides some best practices for verification to avoid such known problems.

Using DANE with SVCB and QUIC - draft-ietf-dnsop-svcb-dane

One of the issues with DNSSEC has been that it has lacked a compelling user case. It's all well and good to understand that a DNS response is authentic, but to what end? As it turned out the role of authentication of the remote service was undertaken by TLS and the domain name certificates, authenticated within the framework of the Web PKI.

About a decade ago the IETF published RFC 7671, on DNS-Based Authentication of Named Entities (DANE). Given that an X.509 domain name certificate is a third party commentary that a particular entity has a domain name, then surely a better (and at the time far cheaper) way is to place this information directly into the DNS and rely on DNSSEC to ensure that users can authenticate the information? Despite its attractions, DANE did not enjoy widespread deployment. LetsEncrypt came out with free-of-charge domain name certificates and quickly assumed a majority position in the domain name certificate market, while at the same time pushing DNSSEC validation out to the edge for DNS placed a time burden on end users and DNSSEC validation in DANE is still supported by iterative DNS requests.

This draft describes the interaction of DANE with indirection via Service Bindings, i.e. SVCB-compatible records such as SVCB and HTTPS and also explains how to use DANE with new TLS-based transports such as QUIC.

I don't think this work materially changes DANE's prospects. The delays to perform DNSSEC validation at the endpoint are still present, and tolerance for additional delays in the connection process just does not exist in today's Internet. If we could bundle the entire DNSSEC validation chain into some form of extension to the SVCB record and serve this data in some form of bundle of IP packets, then we might be able to make this DNSSEC validation process sufficiently fast to be an interesting alternative to domain name certificates. Until then I'm afraid this this draft is headed to the "potentially good ideas that just won't make it" bucket!

More generally, about the use of SVCB records, it appears that the DNS oscillates between just-in-time and just-in-case provisioning. In the just-in-time model the DNS delivers exactly what the client asked for, and no more. If you ask for an IPv4 address for a name you get precisely that. No IPv6 addresses, as that's a different query. The SVCB model delivers a prepared bundle of service connection information, some of which may be relevant you to, and some not. One school of thought is that queries are cheap, and generating more queries is not a costly imposition on the DNS. Another line of thought is that any form of added delay is to be avoided, and the more you can fit into a single query/response cycle, the better. As well as the SVCB approach there is the perennial question of why can't you place multiple query types in a single DNS query? And a related question as to why can't you provide multiple Query Types in answers (draft-bellis-dnsext-multi-qtypes-07.html)? One potential answer to this type of question can be found in the multi-qtypes draft, namely that "The idea that only a single question is allowed is sufficiently entrenched that many DNS servers will simply return an error (or fail to response at all) if they receive a query with a question count (QDCOUNT) of more than one."

Grease - draft-ietf-dnsop-grease

It was a "moment" in the historical saga of IPv4 address space that when we turned to use the so-called "Class E" space, we found that while the registry had tagged this address block as "reserved for future use" it appeared that many implementations interpreted this as an in instruction of the form: "not to be used – ever!"

The lesson from this case is that long term evolvability of any protocol requires the ability to support change, and reserved or currently unused extension code points in a protocol should not cause implementations to unilaterally reject packets that use such code points. "Greasing" is one technique that exercises the regular use of unallocated protocol extension points to prevent ossification of their current usage patterns by middleboxes or DNS implementations. This draft describes considerations and proposals for applying grease to the DNS protocol.

DNS Integration - draft-sheth-dns-integration

The DNS is not the only naming framework used on the Internet, and various forms of alternative name frameworks continue to be explored. Whether it's as simple as an alternative DNS root, through to an entirely distinct name framework used by the TOR network, all kinds of name systems have been explored over the past few decades.

An enduring question is "how do these name systems co-exist with the DNS?" This space is explored in this draft. Although like its predecessors over the years, I am left with more questions than answers after reading this draft. Should such alternative names attempt to act with the same look and feel as DNS names? Or should they eschew all appearance of seamless integratoion and adopt an entirely distinct form of use and resolution? Should we strive to minimize friction? Or is this friction between different naming frameworks inescapable?

Happy Eyeballs for the DNS - draft-momoka-dnsop-3901bis

RFC 3901 was perhaps a misleading document. Titled "DNS IPv6 Transport Guidelines", the document recommended that every recursive name server SHOULD be either IPv4-only or dual stack, and every DNS zone SHOULD be served by at least one IPv4-reachable authoritative name server. It made a whole lot of sense at the time given that IPv6 is still not accessible to more than one half of the Internet user base, and using IPv6 as the protocol substrate is not as reliable as IPv4 for all forms of DNS queries and responses.

But the advice in RFC3901, namely to ensure that IPv4 is still used by DNS resolvers and servers, has apparently offended some of the IPv6 zealots out there. The advice in this *bis* draft, namely to advocate using IPv6 in all cases for DNS transactions, despite some clearly evident operational issues, strikes me as consistent with much of the strident messaging about IPv6 in recent years, which eschews careful measurement and conservative engineering and is all too willing to throw users under the bus in the name of protocol correctness! Perhaps the energy behind this draft would be better spent firstly ensuring that all DNS resolvers and servers have taken the necessary steps to avoid IP level packet fragmentation, and all DNS infrastructure supports TCP transport to the same level of ubiquity as UDP. Once that has been largely achieved, it would be safe to return to this particular proposal about the use of IPv6 for DNS protocol transactions.

3. New Work

Distributed DNSSSEC Multi-signer

In the quest for resilience a conventional measure is to introduce replication of service delivery platforms. That way if there's a failure in one platform. the other platform would still be available. In DNS terms, all the secondary nameservers do not need to be operated by the same operator.

Does DNSSEC-signing of the zone change this? If the zone is pre-signed, then essentially nothing changes. However, there is an increasing interest in using front-end signers for servers. Here the unsigned zone is distributed to the name servers and the front-end signer signs responses with a local copy of the zone-signing key. This arrangement accommodates larger zones, relieves the need for the zone administrator to perform zone signing and can reduce the size of signed negative answers (see **draft-ietf-dnsop-compact-denial-of-existence**). But it comes at a cost of key synchronisation. The assumption here is that each service provider uses its own keys to sign responses from this zone, and in that case careful coordination of the DNSKEY and DS records is required.

A proposal to automate this coordination is described in delegation-mgmt-via-ddns, that proposes to use Dynamic DNS (DDNS) and a new resource record, the HSYNC record to signal this "horizontal " synchronisation across service providers. I suspect that the complexity of this particular automation solution may be its undoing.

Updating SIG(0)

DNSSEC provides data security to authenticate DNS data or provide an authenticated deniual of existence. But that's not the entirety of DNS's security requirements. DNS Transaction Security provides authentication of DNS requests and responses. Its most commonly used in Dynamic DNS and communication between primary and secondary DNS nameservers. TSIG (Transaction Signature) is a way toensure the authenticity of a request based on a keyed hash algorithm and a shared secret key (RFC 8945). Where there is no shared secret, there is SIG(0) (RFC 2931), which uses public/private key signatures. SIG(0) can authenticate general requests and replies if the public key is associated with requester/server host. It can also authorize an UPDATE request if the public key is associated with zone or other authority, and Public keys may be stored in the DNS with the KEY RR.

However, the SIG(0) specification has some shortcomings, including the lack of an error field, and no "Original ID" field, which prevents forwarded authentication. The document draft-eastlake-dnsop-eastlake-rfc2931bis-sigzero proposes to address these shortcomings to use EDNS(0) to carry the Original ID to support forwarding servers, and an error code, together with stronger language regarding TCP support.

Collision-Free Keytags

There is a DNSSEC-based resource exhaustion attack on DNSSEC-validating resolvers by preparing a "poisoned" signed zone that uses a collection of DNS keys that share a common key tag value (the so-called "Key Trap" attack). If a DNSSEC-validating resolver is tasked with validating signed data from one of these "poisioned" zones it may end up with a significant computational load in attempting to find the "right" key to validate a signature.

One possible response to this potential vulnerability is proposed in draft-huque-dnsop-keytags-01, which directs zone administrators not to use colliding key tags! All well and good, but this direction seems to head to the wrong side of the fence! Surely the most salient advice we can give is for validating resolvers rather than zone publishers, and the advice is to limit the amount of processing and time resources spent on attempting to resolve a DNS name, and be prepared to abandon query resolution with a SERVFAIL response. Some resolver implementations already do this, and maybe this proposal should be directing other implementations to follow a similar path.

DELEG

Over the past 18 months there has been a proposal to re-work delegation in the DNS, consideration of which has been referred to by the name of a new delegation resource record, DELEG. It's a big enough topic that a new Working Group has been spun up to develop this proposal.

The work has come from a number of triggers:

- It is not possible to DNSSEC-sign referral responses, which implies that such referral responses are vulnerable to on-path substitution attacks, even if both parent and child zones are signed and the originator of the request that triggered the referral response requests DNSSEC data, and is capable of validating responses.
- It is not possible to use alias nameserver names (i.e. the target of a NS record cannot itself be a CNAME record). In this environment of outsourced DNS operations, this level of flexibility in having indirection in nameservers would be helpful.
- It is not possible to signal an authoritative nameserver's capability to respond to DNS requests using encrypted transport protocols (DNS over TLS, QUIC or HTTPS). This is unfinished work from the DNS Privacy Working Group, and while it is possible to use probe queries to establish a nameserver's capabilities, direct signalling of capabilities in the DNS would be far more efficient.

The original proposal replaced the NS record, which is held in both the child and parent zones. with a new resource record, namely a DELEG record. The DELEG record value was proposed to follow the SVCB model, allowing multiple attributes, including transport protocol, address hints and relative preference, to be part of the record's value. It was to be authoritative at the parent zone and therefore signed by the parent zone key. It would also permit the alias form of the SVCB record, allowing alias names for nameservers. It was proposed to sit alongside existing NS-style records, and its use in referral responses would be controlled by an EDNS(0) capability flag.

More recently there has been a second proposal, referred to as incremental Delegation (IDELEG), using a reserved label, "_deleg". Such labels use an IDELEG resource record which an SVCB-like record that contains the capabilities of the delegated zone's nameservers, all held in the parent zone.

The Working Group has decided to pick just one approach and work on it. But group decisions are hard and the decision process is not going smoothly.

Into all of these has come a further thought. The original DNSSEC design was not intended to have resolvers validate every DNS delegation. Validation was a tool to authenticate the result of the resolution process not the path taken to get there. This observation calls into question the value of the first requirement. In so far as if the zone if DNSSEC-signed then it's the result that is authenticated, not the path taken to reach that result. If the zone if not signed then the result cannot not be validated, irrespective of the domain resolution path. Interestingly, we could associate a SVCB record for a nameserver, in the same way that we have already started to use the HTTPS record for the resolution of the name of a web server. A nameserver SVCB record could contain supported DNS transport protocol hints and IPv4 and IPv6 address hints. The alias form of the SVCB record would allow alias nameserver names. This SVCB value could be passed to the resolver in the Additional Section of a referral response, in a manner similar to the current handling of a glue record.

Is the objective one of re-defining delegation, or adding signalling to inform resolvers of the capabilities of a zone's nameservers? If it's the latter then its perhaps not necessary to define a new delegation resourcve record, but instead to define a service profile for DNS nameservers.

Conclusions

We continue to place more and more demands on the DNS. We would like it to be faster, we'd like to authenticate DNS data, we'd like to automate the DNS provisioning process, we'd like to open up the DNS and allow role specialization in the various aspects of DNS operations.

All of these objectives can be seen as motivations for the variety of the proposals being considered in the agendas of the IETF's DNS Working Groups. It may be one of the more mature of the Internet's protocols, and certainly one of the more widely used, but it's by no means a static technology. So far, we've been able to make quite fundamental changes to the DNS while still maintaining a cohesive DNS environment that has not isolated users behind obsolete DNS servers. Considering the levels of effort we've expended on the IPv4 to IPv6 transition, the evolution of the DNS has been one that has been surprisingly efficient and relatively smooth!

Disclaimer

Author

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

www.potaroo.net