deleg
Internet-Draft
Updates: 1034, 1035, 6672, 6840 (if approved)
Intended status: Standards Track
Expires: 16 February 2026

T. April
Google, LLC
P. paek
ISC
R. Weber
Akamai Technologies
D. Lawrence
Salesforce

15 August 2025

Extensible Delegation for DNS
draft-ietf-deleg-02

Abstract

A delegation in the Domain Name System (DNS) is a mechanism that enables efficient and distributed management of the DNS namespace. It involves delegating authority over subdomains to specific DNS servers via NS records, allowing for a hierarchical structure and distributing the responsibility for maintaining DNS records.

An NS record contains the hostname of the nameserver for the delegated namespace. Any facilities of that nameserver must be discovered through other mechanisms. This document proposes a new extensible DNS record type, DELEG, for delegation of the authority for a domain. Future documents then can use this mechanism to use additional information about the delegated namespace and the capabilities of authoritative nameservers for the delegated namespace.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://github.com/ietf-wg-deleg/draft-ietf-deleg-base/tree/gh-pages. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-deleg/.

Discussion of this document takes place on the deleg Working Group mailing list (mailto:dd@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/dd/. Subscribe at https://www.ietf.org/mailman/listinfo/dd/.

Source for this draft and an issue tracker can be found at https://github.com/ietf-wg-deleg/draft-ietf-deleg-base/.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

In the Domain Name System, subdomains within the domain name hierarchy are indicated by delegations to servers which are authoritative for their portion of the namespace. The DNS records that do this, called NS records, contain hostnames of nameservers, which resolve to addresses. No other information is available to the resolver. It is limited to connect to the authoritative servers over UDP and TCP port 53. This limitation is a barrier for efficient introduction of new DNS technology.

The proposed DELEG and DELEGI record types remedy this problem by providing extensible parameters to indicate capabilities and additional information, such as addresses that a resolver may use for the delegated authority. The DELEG record is authoritative and thus signed in the parent side of the delegation making it possible to validate all delegation parameters with DNSSEC.

This document only shows how a DELEG record can be used instead of or along side a NS record to create a delegation. Future documents can use the extensible mechanism for more advanced features like connecting to a name server with an encrypted transport.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Terminology regarding the Domain Name System comes from [BCP219], with addition terms defined here:

* legacy name servers: An authoritative server that does not support the DELEG record

* legacy resolvers: A resolver that does not support the DELEG record

* DELEG-aware: An authoritative server or resolver that follows the protocol defined in this document

2. DELEG and DELEGI Record Types

The DELEG record (whose RRtype is TBD) has Rdata is one field, a list of key-value pairs called "delegation information". The delegation information field has wire and display formats that are based on the rules in Appendix A of [RFC9460]. A DELEG record is authoritative for the named zone, and creates a delegation and thus lives in the parent of the named zone.

The DELEGI record has the identical format as the DELEG record. The use of the DELEGI record is different from the use of the DELEG record: it gives information about delegation. DELEGI records are treated like regular authoritative records in their zone.

Some delegation information key-value pairs are actions that a DELEG-aware resolver takes when it gets a DELEG or DELEGI record. The actions defined in this document are described briefly here, and more fully described in Section 3.1.5.

* server-ip4: a set of IPv4 addresses for nameservers of the given zone

* server-ip6: a set of IPv6 addresses for nameservers of the given zone

* server-name: the domain name of a nameserver of the given zone; the addresses must be fetched

* include-name: the domain name of a zone that has more information about the nameservers of the given zone

Future documents might define additional delegation information that are actions, and might also define delegation information key-value pairs that modify actions.

TODO: Add some introduction comparing how resolvers see legacy delegatation (set of NS and A/AAAA records) and DELEG delegation (DELEG and DELEGI records with server-ip4 and server-ip6 keys)

3. Use of DELEG Records

A DELEG RRset MAY be present at a delegation point. The DELEG RRset MAY contain multiple records. DELEG RRsets MUST NOT appear at a zone’s apex.

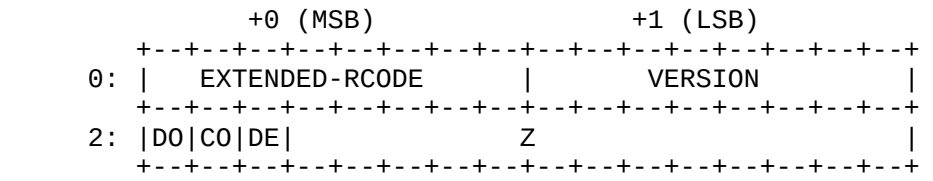
A DELEG RRset MAY be present with or without NS or DS RRsets at the delegation point.

3.1. Resolvers

3.1.1. Signaling DELEG support

A resolver that is DELEG-aware MUST signal its support by sending the DE bit when iterating.

This bit is referred to as the "DELEG" (DE) bit. In the context of the EDNS0 OPT meta-RR, the DE bit is the TBD of the "extended RCODE and flags" portion of the EDNS0 OPT meta-RR, structured as follows (to be updated when assigned):



Setting the DE bit to one in a query indicates the resolver understands new DELEG semantics and does not need NS records to follow a referral. The DE bit cleared (set to zero) indicates the resolver is unprepared to handle DELEG and hence can only be served NS, DS and glue in a delegation response.

Motivation: For a long time there will be both DELEG and NS needed for delegation. As both methods should be configured to get to a proper resolution it is not necessary to send both in a referral response. We therefore purpose an EDNS flag to be use similar to the DO Bit for DNSSEC to be used to signal that the sender understands DELEG and does not need NS or glue information in the referral.

3.1.2. Referral

The DELEG record creates a zone cut similar to the NS record.

If one or more DELEG records exist at a given delegation point, a DELEG-aware resolver MUST treat the name servers from those DELEG records as authoritative for the child zone. In such case, a DELEG-aware resolver MUST NOT use NS records even if they happen to be present in cache, even if resolution using DELEG records have failed for any reason. Such fallback from DELEG to NS would invalidate security guarantees of DELEG protocol.

If no DELEG record exists at a given delegation point, DELEG-aware resolvers MUST use NS records as specified by [RFC1034].

3.1.3. Parent-side types, QTYPE=DELEG

Record types defined as authoritative on the parent side of zone cut (currently DS and DELEG types) retain the same special handling as described in Section 2.6 of [RFC4035].

Legacy resolvers can get different types of answers for QTYPE=DELEG queries based on the configuration of the server, such as whether it is DELEG-aware and whether it also is authoritative for subdomains.

3.1.4. Algorithm for "Finding the Best Servers to Ask"

This document updates instructions for finding the best servers to ask. That information currently is covered in Section 5.3.3 of [RFC1034] and Section 3.4.1 of [RFC6672] with the text "2. Find the best servers to ask." Section 3.1.4.1 of [RFC4035] should have explicitly updated Section 5.3.3 of [RFC1034] for the DS RRtype, but failed to do so; this was remedied by [RFC6672]. This document simply extends this existing behavior to DELEG RRtype as well, and makes this special case explicit.

When a DELEG RRset exists in a zone, DELEG-aware resolvers ignore the NS RRset for that zone. This means that the DELEG-aware resolver ignores the NS RRset in the zone’s parent as well as any cached NS RRset that the resolver might have gotten by looking in the apex of the zone.

DELEG and NS RRtypes can be used differently at each delegation level, and DELEG-aware resolvers MUST be able follow chains of delegations which combines both types in arbitrary ways.

An example of a valid delegation tree:

```
; root zone with NS-only delegations
. SOA ...
test. NS ...

; test. zone with NS+DELEG delegations
test. SOA ...
sld.test. NS ...
sld.test. DELEG ...

; sld.test. zone with NS-only delegation
sld.test. SOA ...
nssub.sld.test. NS ...

; nssub.sld.test. zone with DELEG-only delegation
delegsub.sub.sld.test. DELEG ...
```

TODO: after the text below, refer back to this figure and show the order that a DELEG-aware resolver would take when there is a failure to find any good DELEG addresses at sub.sld.test, then any usable nameservers at sub.sld.test, and then maybe a good DELEG record at test.

The terms SNAME and SLIST used here are defined in Section 5.3.2 of [RFC1034]:

SNAME is the domain name we are searching for.

SLIST is a structure which describes the name servers and the zone which the resolver is currently trying to query. Neither [RFC1034] nor this document define how a resolver uses SLIST; they only define how to populate it.

A DELEG-aware SLIST needs to be able to hold two types of information: delegations defined by NS records and delegations defined by DELEG records. DELEG and NS delegations can create cyclic dependencies and/or lead to duplicate entries which point to the same server. Resolvers need to enforce suitable limits to prevent damage even if someone has incorrectly configured some of the data used to create an SLIST.

This leads to a modified description of find the best servers to ask" from earlier documents for DELEG-aware resolvers. That description becomes:

1. Determine deepest possible zone cut which can potentially hold the answer for given (query name, type, class) combination:

    1. Start with SNAME equal to QNAME.

    2. If QTYPE is a type that is authoritative at the parent side of a zone cut (currently, DS or DELEG), remove the leftmost label from SNAME. For example, if the QNAME is "test.example." and th eQTYPE is DELEG or DS, set SNAME to "example.".

2. Look for locally-available DELEG and NS RRsets, starting at current SNAME.

    1. For given SNAME, check for existence of a DELEG RRset. If it exists, the resolver MUST use its content to populate SLIST. However, if the DELEG RRset is known to exist but is unusable (for example, if it is found in DNSSEC BAD cache), the resolver MUST NOT instead use an NS RRset, even if it is locally available; instead, the resolver MUST treat this case as if no servers were available.

    2. If a given SNAME is proven to not have a DELEG RRset but does have NS RRset, the resolver MUST copy the NS RRset into SLIST.

    3. If SLIST is now populated, stop walking up the DNS tree. However, if SLIST is not populated, remove leftmost label from SNAME and go back to the first step, using the new

(shortened) SNAME. Do not go back to the first step if doing so would exceed the amount of work that the resolver is configured to do when processing names; see Section 4.1.

The rest of the Step 2's description is not affected by this document.

(TODO: Determine what to do about ". DELEG" or ". DS" queries, which by definition do not exist.)

3.1.5. Actions in Delegation Information

The DELEG and DELEGI records have four keys that describe actions the resolver takes. The purpose of these actions is to populate the SLIST with IP addresses of the nameservers for a zone. The actions defined in this document are:

* server-ip4: a set of IPv4 addresses for nameservers of the given zone

* server-ip6: a set of IPv6 addresses for nameservers of the given zone

* server-name: the domain name of a nameserver of the given zone; the addresses must be fetched

* include-name: the domain name of a zone that has more information about the nameservers of the given zone

The presentation values for server-ip4 and server-ip6 are comma-separated list of one or more IP addresses of the appropriate family in standard textual format [RFC5952] [RFC4001]. The wire formats for server-ip4 and server-ip6 are a sequence of IP addresses in network byte order (for the respective address family).

The presentation values for server-name and include-name are as full-qualified domain names. The wire formats are the same as the wire formats for domain names, and MUST NOT be compressed.

If any of these keys are used, it MUST have a value (that is, it cannot be a key with a zero-length value).

A DELEG or DELEGI record SHOULD have only one of the following:

* one server-ip4 key

* one server-ip6 key

* one server-ip4 and one server-ip6 key

* one server-name key

* one include-name key

3.1.6. Populating the SLIST from DELEG and DELEGI Records

Each individual DELEG record inside a DELEG RRset, or each individual DELEGI record in a DELEGI RRset, can cause the addition of zero or more entries to SLIST.

A resolver processes each individual DELEG record within a DELEG RRset, or each individual DELEGI record in a DELEGI RRset, using the following steps:

1. If one or more server-ip4 or server-ip6 actions are present inside the record, copy all the address values from either key into SLIST. Ignore any server-name or include-name keys that are (erroneously) present in the same record. Stop processing this record.

2. If a server-name action is present in the record, resolve it into addresses from the resolver cache or using A and AAAA queries. Copy these addresses into SLIST. Ignore any include-name keys that are (erroneously) present in the same record. Stop processing this record.

3. If a include-name action is present in the record, resolve it into a DELEGI RRset from the resolver cache or by sending queries for the domain name in the value of the include-name pair. Go through these same steps with the result of the DELEGI RRset, after checking that the maximum loop count described in Section 4.1 has not been reached.

4. If none of the above applies, SLIST is not modified by this particular record.

A DELEG-aware resolver MAY implement lazy filling of SLIST, such as by deferring processing remaining records if SLIST already has what the resolver considers a sufficiently large pool of addresses to contact.

3.2. Authoritative Servers

DELEG-aware authoritative servers act differently when handling queries from DELEG-unaware clients (those with DE=0) than from DELEG-aware clients (those with DE=1).

The server MUST copy the value of the DE bit from the query into the response. (TODO: not really necessary protocol-wise, but might be nice for monitoring the deployment?)

3.2.1. DELEG-unaware Clients

DELEG-unaware clients do not use DELEG records for delegation. When a DELEG-aware authoritative server responds to a DELEG-unaware client, any DELEG record in the response does not create zone cut, is not returned in referral responses, and is not considered authoritative on the parent side of a zone cut. Because of this, DELEG-aware authoritative servers MUST answer as if they are DELEG-unaware. Please note this instruction does not affect DNSSEC signing, i.e. no special handling for NSEC type bitmap is necessary and DELEG RRtype is accurately represented even for DELEG-unaware clients.

Two specific cases of DELEG-aware authoritative responding in DELEG-unaware manner are described here.

3.2.1.1. DELEG-unaware Clients Requesting QTYPE=DELEG

In DELEG-unaware clients, records with the DELEG RRtype are not authoritative on the parent side. Thus, queries with DE=0 and QTYPE=DELEG MUST result in a legacy referral response.

3.2.1.2. DELEG-unaware Clients with DELEG RRs Present but No NS RRs

DELEG-unaware clients might ask for a name which belongs to a zone delegated only with DELEG RRs (that is, without any NS RRs). Such zone is, by definition, not resolvable for DELEG-unaware clients. In this case, the DELEG record itself cannot create a zone cut, and the DELEG-aware authoritative server MUST return a legacy response.

The legacy response might be confusing for subdomains of zones which actually exist because DELEG-aware clients would get a different answer, namely a delegation. An example of a legacy response is in Appendix A.3.2.

The authoritative server is RECOMMENDED to supplement these responses to DELEG-unaware resolvers with Extended DNS Error "New Delegation Only".

TODO: debate if WG wants to do explicit SERVFAIL for this case instead of ’just’ EDE.

3.2.2. DELEG-aware Clients

When the client indicates that it is DELEG-aware by setting DE=1 in the query, DELEG-aware authoritative servers treat DELEG records as zone cuts, and the servers are authoritative on parent side of zone cut. This new zone cut has priority over legacy delegation with NS RRset.

3.2.2.1. DELEG-aware Clients Requesting QTYPE=DELEG

An explicit query for the DELEG RRtype at a delegation point behaves much like query for the DS RRtype: the server answers authoritatively from the parent zone. All previous specifications for special handling queries with QTYPE=DS apply equally to QTYPE=DELEG. In summary, server either provides an authoritative DELEG RRset or proves its non-existence.

3.2.2.2. Delegation with DELEG

If the delegation has a DELEG RRset, the authoritative server MUST put the DELEG RRset into the Authority section of the referral. In this case, the server MUST NOT include the NS RRset into the Authority section. Presence of the covering RRSIG follows the normal DNSSEC specification for answers with authoritative zone data.

Similarly, rules for DS RRset inclusion into referrals apply as specified by DNSSEC protocol.

3.2.2.3. DELEG-aware Clients with NS RRs Present but No DELEG RRs

If the delegation does not have a DELEG RRset, the authoritative server MUST put the NS RRset into the authority section of the referral. Absence of DELEG RRset must be proven as specified by DNSSEC protocol for authoritative data.

Similarly, rules for DS RRset inclusion into referrals apply as specified by the DNSSEC protocol. Please note in practice the same process and records are used to prove non-existence of DELEG and DS RRsets.

3.3. DNSSEC Signers

The DELEG record is authoritative on the parent side of a zone cut and needs to be signed as such. Existing rules from DNSSEC specification apply. In summary: for DNSSEC signing, treat the DELEG RRtype the same way as the DS RRtype.

In order to protect validators from downgrade attacks this draft introduces a new DNSKEY flag ADT (Authoritative Delegation Types). In zones which contain a DELEG RRset, this flag MUST be set to 1 in at least one of the DNSKEY records published in the zone.

3.4. DNSSEC Validators

DELEG awareness introduces additional requirements on validators.

3.4.1. Clarifications on Nonexistence Proofs

This document updates Section 4.1 of [RFC6840] to include "NS or DELEG" types in type bitmap as indication of a delegation point, and generalizes applicability of ancestor delegation proof to all RRtypes that are authoritative at the parent (that is, both DS and DELEG). The text in that section is updated as follows:

An "ancestor delegation" NSEC RR (or NSEC3 RR) is one with:

* the NS and/or DELEG bit set,

* the Start of Authority (SOA) bit clear, and

* a signer field that is shorter than the owner name of the NSEC RR, or the original owner name for the NSEC3 RR.

Ancestor delegation NSEC or NSEC3 RRs MUST NOT be used to assume nonexistence of any RRs below that zone cut, which include all RRs at that (original) owner name other than types authoritative at the parent-side of zone cut (DS and DELEG), and all RRs below that owner name regardless of type.

3.4.2. Insecure Delegation Proofs

This document updates Section 4.4 of [RFC6840] to include secure DELEG support, and explicitly states that Opt-Out is not applicable to DELEG. The first paragraph of that section is updated to read:

Section 5.2 of [RFC4035] specifies that a validator, when proving a delegation is not secure, needs to check for the absence of the DS and SOA bits in the NSEC (or NSEC3) type bitmap. The validator also MUST check for the presence of the NS or the DELEG bit in the matching NSEC (or NSEC3) RR (proving that there is, indeed, a delegation). Alternately, the validator must make sure that the delegation with NS record is covered by an NSEC3 RR with the Opt-Out flag set. Opt-Out is not applicable to DELEG RRtype because DELEG records are authoritative at the parent side of a zone cut in the same way that DS RRtypes are.

3.4.3. Referral downgrade protection

When DNSKEY flag ADT is set to 1, a DELEG-aware validator MUST prove the absence of a DELEG RRset in referral responses for a zone.

Without this check, an attacker could strip the DELEG RRset from a referral response and replace it with an unsigned (and potentially malicious) NS RRset. A referral response with an unsigned NS and signed DS RRsets does not require additional proofs of nonexistence according to pre-DELEG DNSSEC specification, and it would have been accepted as a delegation without DELEG RRset.

3.4.4. Chaining

A Validating Stub Resolver that is DELEG-aware has to use a Security-Aware Resolver that is DELEG-aware and, if it is behind a forwarder, that forwarder has to be security-aware and DELEG-aware as well.

4. Security Considerations

TODO: Add more here

4.1. Preventing Over-work Attacks

Resolvers MUST prevent situations where accidental misconfiguration of zones or malicious attacks cause them to perform too much work when resolving. This document describes two sets of actions that, if not controlled, could lead to over-work attacks:

* Names with many subdomains can cause walking up the tree to populate SLIST (Section 3.1.4) to be burdensome. To prevent this, the resolver SHOULD NOT walk up more than %%TODO: come up with a number%% labels in order to contribute to SLIST.

* Long chains of include-name actions (Section 3.1.5), and those with circular chains if include-name actions, can be burdensome. To prevent this, the resolver SHOULD NOT follow more than 3 include-name chains in an RRset when populating SLIST. Note that include-name chains can have CNAME steps in them; in such a case, a CNAME step is counted the same as a DELEGI step when determining when to stop following a chain.

5. IANA Considerations

5.1. Changes to Existing Registries

IANA is requested to allocate the DELEG RR in the Resource Record (RR) TYPEs registry, with the meaning of "enhanced delegation information" and referencing this document.

IANA is requested to assign a new bit in the DNSKEY RR Flags registry ([RFC4034]) for the ADT bit (N), with the description "Authoritative Delegation Types" and referencing this document. For compatibility reasons we request the bit 14 to be used. This value has been proven to work whereas bit 0 was proven to break in practical deployments (because of bugs).

IANA is requested to assign a bit from the EDNS Header Flags registry ([RFC6891]), with the abbreviation DE, the description "DELEG enabled" and referencing this document.

IANA is requested to assign a value from the Extended DNS Error Codes ([RFC8914]), with the Purpose "New Delegation Only" and referencing this document.

5.2. New Registry for Delegation Information

IANA is requested to create the "DELEG Delegation Information" registry. This registry defines the namespace for delegation information keys, including string representations and numeric key values.

5.2.1. Procedure

A registration MUST include the following fields:

Number: Wire-format numeric identifier (range 0-65535) Name: Unique presentation name Meaning: A short description Reference: Location of specification or registration source Change Controller: Person or entity, with contact information if appropriate

The characters in the registered Name field entry MUST be lowercase alphanumeric or "-". The name MUST NOT start with "key".

The registration policy for new entries is Expert Review ([RFC8126]). The designated expert MUST ensure that the reference is stable and publicly available and that it specifies how to convert the delegation information's presentation format to wire format. The reference MAY be any individual's Internet-Draft or a document from any other source with similar assurances of stability and availability. An entry MAY specify a reference of the form "Same as (other key name)" if it uses the same presentation and wire formats as an existing key.

This arrangement supports the development of new parameters while ensuring that zone files can be made interoperable.

5.2.2. Initial Contents

The "DELEG Delegation Information" registry should be populated with the following initial registrations:

Number: 1
Name: server-ip4
Meaning: A set of IPv4 addresses of nameservers
Reference: {{actions}} of this document
Change Controller: IETF

Number: 2
Name: server-ip6
Meaning: A set of IPv6 addresses of nameservers
Reference: {{actions}} of this document
Change Controller: IETF

Number: 3
Name: server-name
Meaning: The fully-qualified domain name of a nameserver
Reference: {{actions}} of this document
Change Controller: IETF

Number: 4
Name: include-name
Meaning: The fully-qualified domain of a DELEGI record
Reference: {{actions}} of this document
Change Controller: IETF

The registration for number 0 is reserved.
The registration for numbers 65280-65535 is reserved for private use.

6. References

6.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/rfc/rfc1034>.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <https://www.rfc-editor.org/rfc/rfc4034>.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005, <https://www.rfc-editor.org/rfc/rfc4035>.

[RFC6672] Rose, S. and W. Wijngaards, "DNAME Redirection in the DNS", RFC 6672, DOI 10.17487/RFC6672, June 2012, <https://www.rfc-editor.org/rfc/rfc6672>.

[RFC6840] Weiler, S., Ed. and D. Blacka, Ed., "Clarifications and Implementation Notes for DNS Security (DNSSEC)", RFC 6840, DOI 10.17487/RFC6840, February 2013, <https://www.rfc-editor.org/rfc/rfc6840>.

[RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/RFC6891, April 2013, <https://www.rfc-editor.org/rfc/rfc6891>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/rfc/rfc8126>.

[RFC8914] Kumari, W., Hunt, E., Arends, R., Hardaker, W., and D. Lawrence, "Extended DNS Errors", RFC 8914, DOI 10.17487/RFC8914, October 2020, <https://www.rfc-editor.org/rfc/rfc8914>.

6.2. Informative References

[BCP219] Best Current Practice 219, <https://www.rfc-editor.org/info/bcp219>. At the time of writing, this BCP comprises the following:

Hoffman, P. and K. Fujiwara, "DNS Terminology", BCP 219, RFC 9499, DOI 10.17487/RFC9499, March 2024, <https://www.rfc-editor.org/info/rfc9499>.

[I-D.tapril-ns2]
April, T., "Parameterized Nameserver Delegation with NS2 and NS2T", Work in Progress, Internet-Draft, draft-tapril-ns2-01, 13 July 2020, <https://datatracker.ietf.org/doc/html/draft-tapril-ns2-01>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, DOI 10.17487/RFC4001, February 2005, <https://www.rfc-editor.org/rfc/rfc4001>.

[RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, DOI 10.17487/RFC5952, August 2010, <https://www.rfc-editor.org/rfc/rfc5952>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC9460] Schwartz, B., Bishop, M., and E. Nygren, "Service Binding and Parameter Specification via the DNS (SVCB and HTTPS Resource Records)", RFC 9460, DOI 10.17487/RFC9460, November 2023, <https://www.rfc-editor.org/rfc/rfc9460>.

Appendix A. Examples

The following example shows an excerpt from a signed root zone. It shows the delegation point for "example." and "test."

The "example." delegation has DELEG and NS records. The "test." delegation has DELEG but no NS records.

TODO: Examples of using server-ip4 and server-ip6. Also, examples that show DELEGI records in ns2.example.net and ns3.example.org.

```
example.   300 IN DELEG server-name=a.example.
example.   300 IN DELEG include-name=ns2.example.net.
example.   300 IN DELEG include-name=ns3.example.org.
example.   300 IN RRSIG DELEG 13 4 300 20250214164848 (
                        20250207134348 21261 . HyDHYVT5KcqWc7J..= )
example.   300 IN NS    a.example.
example.   300 IN NS    b.example.net.
example.   300 IN NS    c.example.org.
example.   300 IN DS    65163 13 2 5F86F2F3AE2B02...
example.   300 IN RRSIG DS 13 4 300 20250214164848 (
                        20250207134348 21261 . O0k558jHhyrC21J..= )
example.   300 IN NSEC  a.example. NS DS RRSIG NSEC DELEG
example.   300 IN RRSIG NSEC 13 4 300 20250214164848 (
                        20250207134348 21261 . 1Kl8vab96gG21Aa..= )
a.example. 300 IN A     192.0.2.1
a.example. 300 IN AAAA  2001:DB8::1
```

The "test." delegation point has a DELEG record and no NS record.

```
test.      300 IN DELEG include-name=ns2.example.net
test.      300 IN RRSIG DELEG 13 4 300 20250214164848 (
                        20250207134348 21261 . 98Aac9f7A1Ac26Q..= )
test.      300 IN NSEC  a.test. RRSIG NSEC DELEG
test.      300 IN RRSIG NSEC 13 4 300  20250214164848 (
                        20250207134348 21261 . kj7YY5tr9h7UqlK..= )
```

A.1. Responses

The following sections show referral examples:

A.2. DO bit clear, DE bit clear

A.2.1. Query for foo.example

```
   ;; Header: QR RCODE=0
   ;;

   ;; Question
   foo.example.  IN MX

   ;; Answer
   ;; (empty)

   ;; Authority
   example.   300 IN NS    a.example.
   example.   300 IN NS    b.example.net.
   example.   300 IN NS    c.example.org.

   ;; Additional
   a.example. 300 IN A     192.0.2.1
   a.example. 300 IN AAAA  2001:DB8::1
```

A.2.2. Query for foo.test

```
   ;; Header: QR AA RCODE=3
   ;;

   ;; Question
   foo.test.   IN MX

   ;; Answer
   ;; (empty)

   ;; Authority
   .   300 IN SOA ...

   ;; Additional
   ;; OPT with Extended DNS Error: New Delegation Only
```

A.3. DO bit set, DE bit clear

A.3.1. Query for foo.example

```
   ;; Header: QR DO RCODE=0
   ;;

   ;; Question
   foo.example.   IN MX

   ;; Answer
   ;; (empty)

   ;; Authority

   example.   300 IN NS    a.example.
   example.   300 IN NS    b.example.net.
   example.   300 IN NS    c.example.org.
   example.   300 IN DS    65163 13 2 5F86F2F3AE2B02...
   example.   300 IN RRSIG DS 13 4 300 20250214164848 (
                           20250207134348 21261 . O0k558jHhyrC21J..= )
   ;; Additional
   a.example. 300 IN A     192.0.2.1
   a.example. 300 IN AAAA  2001:DB8::1
```

A.3.2. Query for foo.test

```
   ;; Header: QR DO AA RCODE=3
   ;;

   ;; Question
   foo.test.      IN MX

   ;; Answer
   ;; (empty)

   ;; Authority
   .          300 IN SOA ...
   .          300 IN RRSIG SOA ...
   .          300 IN NSEC  aaa NS SOA RRSIG NSEC DNSKEY ZONEMD
   .          300 IN RRSIG NSEC 13 4 300
   test.      300 IN NSEC  a.test. RRSIG NSEC DELEG
   test.      300 IN RRSIG NSEC 13 4 300  20250214164848 (
                           20250207134348 21261 . aBFYask;djf7UqlK..= )

   ;; Additional
   ;; OPT with Extended DNS Error: New Delegation Only
```

A.4. DO bit clear, DE bit set

A.4.1. Query for foo.example

```
   ;; Header: QR DE RCODE=0
   ;;

   ;; Question
   foo.example.  IN MX

   ;; Answer
   ;; (empty)

   ;; Authority
   example.   300 IN DELEG server-name=a.example.
   example.   300 IN DELEG include-name=ns2.example.net.
   example.   300 IN DELEG include-name=ns3.example.org.

   ;; Additional
   ;; (empty)
```

A.4.2. Query for foo.test

```
   ;; Header: QR AA RCODE=0
   ;;

   ;; Question
   foo.test.   IN MX

   ;; Answer
   ;; (empty)

   ;; Authority
   test.      300 IN DELEG include-name=ns2.example.net

   ;; Additional
   ;; (empty)
```

A.5. DO bit set, DE bit set

A.5.1. Query for foo.example

```
   ;; Header: QR DO DE RCODE=0
   ;;

   ;; Question
   foo.example.  IN MX

   ;; Answer
   ;; (empty)

   ;; Authority

   example.   300 IN DELEG server-name=a.example.
   example.   300 IN DELEG include-name=ns2.example.net.
   example.   300 IN DELEG include-name=ns3.example.org.
   example.   300 IN RRSIG DELEG 13 4 300 20250214164848 (
                           20250207134348 21261 . HyDHYVT5KcqWc7J..= )
   example.   300 IN DS    65163 13 2 5F86F2F3AE2B02...
   example.   300 IN RRSIG DS 13 4 300 20250214164848 (
                           20250207134348 21261 . O0k558jHhyrC21J..= )

   ;; Additional
   a.example. 300 IN A     192.0.2.1
   a.example. 300 IN AAAA  2001:DB8::1
```

A.5.2. Query for foo.test

```
   ;; Header: QR DO DE AA RCODE=0
   ;;

   ;; Question
   foo.test.      IN MX

   ;; Answer
   ;; (empty)

   ;; Authority
   test.      300 IN DELEG include-name=ns2.example.net.
   test.      300 IN RRSIG DELEG 13 4 300 20250214164848 (
                           20250207134348 21261 . 98Aac9f7A1Ac26Q..= )
   test.      300 IN NSEC  a.test. RRSIG NSEC DELEG
   test.      300 IN RRSIG NSEC 13 4 300  20250214164848 (
                           20250207134348 21261 . kj7YY5tr9h7UqlK..= )

   ;; Additional
   ;; (empty)
```

Appendix B. Acknowledgments {:unnumbered}

This document is heavily based on past work done by Tim April in [I-D.tapril-ns2] and thus extends the thanks to the people helping on this which are: John Levine, Erik Nygren, Jon Reed, Ben Kaduk, Mashooq Muhaimen, Jason Moreau, Jerrod Wiesman, Billy Tiemann, Gordon Marx and Brian Wellington.

Work on DELEG protocol has started at IETF 118 hackaton. Hackaton participants: Christian Elmerot, David Blacka, David Lawrence, Edward Lewis, Erik Nygren, George Michaelson, Jan Velk, Klaus Darilion, Libor Peltan, Manu Bretelle, Peter van Dijk, Petr paek, Philip Homburg, Ralf Weber, Roy Arends, Shane Kerr, Shumon Huque, Vandan Adhvaryu, Vladimr unt.

Other people joined the effort after the initial hackaton: Ben Schwartz, Bob Halley, Paul Hoffman, ...

Appendix C. TODO

RFC EDITOR: PLEASE REMOVE THE THIS SECTION PRIOR TO PUBLICATION.

* Write a security considerations section

* Change the parameters form temporary to permanent once IANA assigned. Temporary use:

  - DELEG QType code is 65432

  - DELEG EDNS Flag Bit is 3

  - DELEG DNSKEY Flag Bit is 0

Authors' Addresses

Tim April
Google, LLC
Email: ietf@tapril.net

Petr paek
ISC
Email: pspacek@isc.org

Ralf Weber
Akamai Technologies
Email: rweber@akamai.com

David C Lawrence
Salesforce
Email: tale@dd.org