DNS Delegation
Internet-Draft
Intended status: Standards Track
Expires: 5 October 2025

P. Homburg
NLnet Labs
T. Wicinski
Cox Communications
J. V. Zutphen
University of Amsterdam
W. Toorop
NLnet Labs

3 April 2025

Extensible Delegation for DNS
draft-homburg-deleg-01

Abstract

This document proposes a mechanism for extensible delegations in the DNS. The mechanism realizes delegations with resource record sets placed below a _deleg label in the apex of the delegating zone. This authoritative delegation point can be aliased to other names using CNAME and DNAME. This document proposes a new DNS resource record type, IDELEG, which is based on the SVCB and inherits extensibility from it.

IDELEG RRsets containing delegation information will be returned in the authority section in referral responses from supportive authoritative name servers. Lack of support in the authoritative name servers, forwarders or other components, does not obstruct obtaining the delegation information for resolvers, as it is originally authoritative information that can be queried for directly. None, mixed or full deployment of the mechanism on authoritative name servers are all fully functional, allowing for the mechanism to be incrementally deployed on the authoritative name servers.

About This Document

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

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-homburg-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/NLnetLabs/incremental-deleg.

Status of This Memo

This Internet-Draft will expire on 5 October 2025.

Copyright Notice

Table of Contents

1. Introduction

This document describes a delegation mechanism for the Domain Name System (DNS) [STD13] that addresses several matters that, at the time of writing, are suboptimally supported or not supported at all. These matters are elaborated upon in sections 1.1, 1.3 and 1.4. In addition, the mechanism described in this document aspires to be maximally deployable, which is elaborated upon in Section 1.5.

1.1. Signaling capabilities of the authoritative name servers

A new IDELEG resource record (RR) type is introduced in this document, which is based on and inherits the wire and presentation format from SVCB [RFC9460]. All Service Binding Mappings, as well as the capability signalling, that are specified in [RFC9461] are also applicable to IDELEG, with the exception of the limitations on AliasMode records in Section 6 of [RFC9460]. Capability signalling of DNS over Transport Layer Protocol [RFC7858] (DoT), DNS Queries over HTTPS [RFC8484] and DNS over Dedicated QUIC Connections

[RFC9250], on default or alternative ports, can all be used as specified in [RFC9461]. The IDELEG RR type inherits its extensibility from the SVCB RR type, which is designed to be extensible to support future uses (such as keys for encrypting the TLS ClientHello [I-D.ietf-tls-esni].)

1.2. _Note to the RFC Editor_: please remove this subsection before publication.

The name IDELEG is chosen to avoid confusion with [I-D.wesplaap-deleg].

1.3. Outsourcing operation of the delegation

Delegation information is stored at an authoritative location in the zone with this mechanism. Legacy methods to redirect this information to another location, possible under the control of another operator, such as (CNAME Section 3.6.2 of [RFC1034]) and DNAME [RFC6672] remain functional. One could even outsource all delegation operational practice to another party with a DNAME record on the _deleg label on the apex of the delegating zone.

Additional to the legacy methods, a delegation may be outsourced to a set of third parties by having RRs in AliasMode. Unlike SVCB, IDELEG allows for more than a single IDELEG RR in AliasMode in an IDELEG RRset, enabling outsourcing a delegation to multiple different operators.

1.4. DNSSEC protection of the delegation

With legacy delegations, the NS RRset at the parent side of a delegation as well as glue records for the names in the NS RRset are not authoritative and not DNSSEC signed. An adversary that is able to spoof a referral response, can alter this information and redirect all traffic for the delegation to a rogue name server undetected. The adversary can then perceive all queries for the redirected zone (Privacy concern) and alter all unsigned parts of responses (such as further referrals, which is a Security concern).

DNSSEC protection of delegation information prevents that, and is the only countermeasure that also works against on-path attackers. At the time of writing, the only way to DNSSEC-validate and verify delegations at all levels in the DNS hierarchy is to revalidate delegations [I-D.ietf-dnsop-ns-revalidation], which is done after the fact and has other security concerns (Section 7 of [I-D.ietf-dnsop-ns-revalidation]).

Direct delegation information (provided by IDELEG RRs in ServiceMode) includes the hostnames of the authoritative name servers for the delegation as well as IP addresses for those hostnames. Since the information is stored authoritatively in the delegating zone, it will be DNSSEC-signed if the zone is signed. When the delegation is outsourced, then it's protected when the zones providing the aliasing resource records _and_ the zones serving the targets of the aliases are all DNSSEC-signed.

1.5. Maximize ease of deployment

Delegation information is stored authoritatively within the delegating zone. No semantic changes as to what zones are authoritative for what data are needed. As a consequence, existing DNS software, such as authoritative name servers and DNSSEC signing software, can remain unmodified. Unmodified authoritative name server software will serve the delegation information when queried for. Unmodified signers will sign the delegation information in the delegating zone. Support for IDELEG is only required for functionalities following delegations (like recursive resolvers), but they do not need of the components it queries. None, mixed or full deployment of the mechanism on authoritative name servers are all fully functional, allowing for the mechanism to be incrementally deployed on the authoritative name servers.

1.6. 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.

This document follows terminology as defined in [BCP219].

Throughout this document we will also use terminology with the meaning as defined below:

Deleg:
: The delegation mechanism as specified in this document.

IDELEG delegation:
: A delegation as specified in this document

Legacy delegations:
: The way delegations are done in the DNS traditionally as defined in [STD13].

Delegating zone:
The zone in which the delegation is administered. Sometimes also called the "parent zone" of a delegation.

Subzone:
The zone that is delegated to from the delegating zone.

Delegating name:
The name which realizes the delegation. In legacy delegations, this name is the same as the name of the subzone to which the delegation refers. Delegations described in this document are provided with a different name than the zone that is delegated to. If needed we differentiate by explicitly calling it IDELEG delegation name or legacy delegation name.

Deleg query:
An explicit query for IDELEG RRset at the delegating name as used for IDELEG delegations.

Delegation point:
The location in the delegating zone where the RRs are provided that make up the delegation. In legacy delegations, this is the parent side of the zone cut and has the same name as the subzone. With deleg, this is the location given by the IDELEG delegating name. If needed we differentiate by explicitly calling it IDELEG delegation point or legacy delegation point.

Triggering query:
The query on which resolution a recursive resolver is working.

Target zone:
The zone for which the authoritative servers, that a resolver contacts while iterating, are authoritative.

2. The IDELEG resource record type

The IDELEG RR type is a variant of SVCB [RFC9460] for use with resolvers to perform iterative resolution (Section 5.3.3 of [RFC1034]). The IDELEG type requires registration in the "Resource Record (RR) TYPEs" registry under the "Domain Name System (DNS) Parameters" registry group (see IDELEG RR type (Section 11.1)). The protocol-specific mapping specification for iterative resolutions are the same as those for "DNS Servers" [RFC9461].

Section 2.4.2 of [RFC9460] states that SVCB RRsets SHOULD only have a single RR in AliasMode, and that if multiple AliasMode RRs are present, clients or recursive resolvers SHOULD pick one at random. Different from SVCB (Section 2.4.2 of [RFC9460]), IDELEG allows for

multiple AliasMode RRs to be present in a single IDELEG RRset. Note however that the target of an IDELEG RR in AliasMode is an SVCB RRset for the "dns" service type adhering fully to the Service Binding Mapping for DNS Servers as specified in [RFC9461].

Section 2.4.1 of [RFC9460] states that within an SVCB RRset, all RRs SHOULD have the same mode, and that if an RRset contains a record in AliasMode, the recipient MUST ignore any ServiceMode records in the set. Different from SVCB, mixed ServiceMode and AliasMode RRs are allowed in an IDELEG RRset. When a mixed ServiceMode and AliasMode IDELEG RRset is encountered by a resolver, the resolver first picks one of the AliasMode RRs or all ServiceMode RRs, giving all ServiceMode RRs equal weight as each single AliasMode RR. When the result of that choice is an AliasMode RR, then that RR is followed and the resulting IDELEG RRset is reevaluated. When the result of that choice is all ServiceMode RRs, then within that set the resolver adheres to the ServicePriority value.

At the delegation point (for example customer._deleg.example.), the host names of the authoritative name servers for the subzone, are given in the TargetName RDATA field of IDELEG records in ServiceMode. Port Prefix Naming Section 3 of [RFC9461] is not used at the delegation point, but MUST be used when resolving the aliased-to name servers with IDELEG RRs in AliasMode.

3. Delegation administration

An IDELEG delegation is realized with an IDELEG Resource Record set (RRset) [RFC9460] below a specially for the purpose reserved label with the name _deleg at the apex of the delegating zone. The _deleg label scopes the interpretation of the IDELEG records and requires registration in the "Underscored and Globally Scoped DNS Node Names" registry (see _deleg Node Name (Section 11.2)). The full scoping of delegations includes the labels that are *below* the _deleg label and those, together with the name of the delegating domain, make up the name of the subzone to which the delegation refers. For example, if the delegating zone is example., then a delegation to subzone customer.example. is realized by a IDELEG RRset at the name customer._deleg.example. in the parent zone. A fully scoped delegating name (such as customer._deleg.example.) is referred to further in this document as the "delegation point".

Note that if the delegation is outsourcing to a single operator represented in a single IDELEG RR, it is RECOMMENDED to refer to the name of the operator’s IDELEG RRset with a CNAME on the delegation point instead of a IDELEG RR in AliasMode Section 10.2 of [RFC9460].

For reasons that will be explained in Section 4.2, operators SHOULD include the following in zones that include IDELEG records: *._deleg 86400 IN IDELEG 0 .

3.1. Examples

3.1.1. One name server within the subzone

```
$ORIGIN example.
@                 IN  SOA    ns zonemaster ...
customer1._deleg  IN  IDELEG 1 ( ns.customer1
                                  ipv4hint=198.51.100.1,203.0.113.1
                                  ipv6hint=2001:db8:1::1,2001:db8:2::1
                                )
```

Figure 1: One name server within the subzone

3.1.2. Two name servers within the subzone

```
$ORIGIN example.
@                 IN  SOA    ns zonemaster ...
customer2._deleg  IN  IDELEG 1 ns1.customer2 ( ipv4hint=198.51.100.1
                                                ipv6hint=2001:db8:1::1
                                              )
                  IN  IDELEG 1 ns2.customer2 ( ipv4hint=203.0.113.1
                                                ipv6hint=2001:db8:2::1
                                              )
```

Figure 2: Two name servers within the subzone

3.1.3. Outsourced to an operator

```
$ORIGIN example.
@                 IN  SOA   ns zonemaster ...
customer3._deleg  IN  CNAME _dns.ns.operator1
```

Figure 3: Outsourced with CNAME

Instead of using CNAME, the outsourcing could also have been accomplished with an IDELEG RRset with a single IDELEG RR in AliasMode. The configuration below is operationally equivalent to the CNAME configuration above. It is RECOMMENDED to use a CNAME over an IDELEG RRset with a single IDELEG RR in AliasMode (Section 10.2 of [RFC9460]). Note that an IDELEG RRset refers with TargetName to a DNS service [RFC9461], which will be looked up using Port Prefix Naming Section 3 of [RFC9461], but that an CNAME refers to the domain name of the target SVCB RRset instead (or CNAME) which may have a _dns prefix.

```
$ORIGIN example.
@                 IN  SOA    ns zonemaster ...
customer3._deleg  IN  IDELEG 0 ns.operator1
```

Figure 4: Outsourced with an AliasMode IDELEG RR

The operator SVCB RRset could for example be:

```
$ORIGIN operator1.example.
@                 IN  SOA    ns zonemaster ...
_dns.ns           IN  SVCB   1 ns ( alpn=h2,dot,h3,doq
                                    ipv4hint=192.0.2.1
                                    ipv6hint=2001:db8:3::1
                                    dohpath=/q{?dns}
                                  )
                  IN  SVCB   2 ns ( ipv4hint=192.0.2.2
                                    ipv6hint=2001:db8:3::2
                                  )
ns                IN  AAAA   2001:db8:3::1
                  IN  AAAA   2001:db8:3::2
                  IN  A      192.0.2.1
                  IN  A      192.0.2.2
```

Figure 5: Operator zone

3.1.4. DNSSEC-signed name servers within the subzone

```
  $ORIGIN
  @                 IN  SOA    ns zonemaster ...
                    IN  RRSIG  SOA ...
                    IN  DNSKEY 257 3 15 ...
                    IN  RRSIG  DNSKEY ...
                    IN  NS     ns.example.
                    IN  RRSIG  NS ...
                    IN  NSEC   customer5._deleg NS SOA RRSIG NSEC DNSKEY
                    IN  RRSIG  NSEC ...

  customer5._deleg  IN  IDELEG 1 ns.customer5 alpn=h2,h3 (
                                 ipv4hint=198.51.100.5
                                 ipv6hint=2001:db8:5::1
                                 dohpath=/dns-query{?dns}
                                 )
                    IN  RRSIG  IDELEG ...
                    IN  NSEC   customer7._deleg RRSIG NSEC IDELEG
                    IN  RRSIG  NSEC ...

  customer7._deleg  IN  CNAME  customer5._deleg
                    IN  RRSIG  CNAME ...
                    IN  NSEC   customer5 CNAME RRSIG NSEC
                    IN  RRSIG  NSEC ...

  customer5         IN  NS     ns.customer5
  ns.customer5      IN  A      198.51.100.5
                    IN  AAAA   2001:db8:5::1
  customer5         IN  DS     17405 15 2 ...
                    IN  RRSIG  DS ...
                    IN  NSEC   customer6 NS DS RRSIG NSEC
                    IN  RRSIG  NSEC ...

  customer6         IN  NS     ns.customer6
  ns.customer6      IN  A      203.0.113.1
                    IN  AAAA   2001:db8:6::1
  customer6         IN  DS     ...
                    IN  RRSIG  DS ...
                    IN  NSEC   customer7 NS DS RRSIG NSEC
                    IN  RRSIG  NSEC ...

  customer7         IN  NS     ns.customer5
                    IN  DS     ...
                    IN  RRSIG  DS ...
                    IN  NSEC   example. NS DS RRSIG NSEC
                    IN  RRSIG  NSEC ...
```

Figure 6: DNSSEC-signed deleg zone

customer5.example. is delegated to in an extensible way and customer6.example. is delegated only in a legacy way. customer7.example.’s delegation is outsourced to customer5’s delegation.

The delegation signals that the authoritative name server supports DoH. customer5.example., customer6.example. and example. are all DNSSEC-signed. The DNSSEC authentication chain links from example. to customer5.example. in the for DNSSEC conventional way with the signed customer5.example. DS RRset in the example. zone. Also, customer6.example. is linked to from example. with the signed customer6.example. DS RRset in the example. zone.

Note that both customer5.example. and customer6.example. have legacy delegations in the zone as well. It is important to have those legacy delegations to maintain support for legacy resolvers, that do not support deleg. DNSSEC signers SHOULD construct the NS RRset and glue for the legacy delegation from the IDELEG RRset.

4. Authoritative name server support

Deleg supporting authoritative name servers include the IDELEG delegation information (or the NSEC(3) records showing the non-existence) in the authority section of referral responses to legacy DNS queries. For example, querying the zone from Figure 6 for www.customer5.example. A, will return the following referral response:

```
;; ->>HEADER<<- opcode: QUERY, rcode: NOERROR, id: 54349
;; flags: qr ; QUERY: 1, ANSWER: 0, AUTHORITY: 5, ADDITIONAL: 2
;; QUESTION SECTION:
;; www.customer5.example.       IN      A

;; ANSWER SECTION:

;; AUTHORITY SECTION:
customer5.example.      3600    IN      NS      ns.customer5.example.
customer5.example.      3600    IN      DS      ...
customer5.example.      3600    IN      RRSIG   DS ...
customer5._deleg.example.       3600    IN      IDELEG 1 (
                ns.customer5.example. alpn=h2,h3
                ipv4hint=198.51.100.5 ipv6hint=2001:db8:5::1
                dohpath=/dns-query{?dns}
                )
customer5._deleg.example.       3600    IN      RRSIG   IDELEG ...

;; ADDITIONAL SECTION:
ns.customer5.example.   3600    IN      A       198.51.100.5
ns.customer5.example.   3600    IN      AAAA    2001:db8:5::1

;; Query time: 0 msec
;; EDNS: version 0; flags: do ; udp: 1232
;; SERVER: 192.0.2.53
;; WHEN: Mon Feb 24 20:36:25 2025
;; MSG SIZE  rcvd: 456
```

Figure 7: A deleg referral response

The referral response in Figure 7 includes the signed IDELEG RRset in the authority section.

As another example, querying the zone from Figure 6 for www.customer6.example. A, will return the following referral response:

```
;; ->>HEADER<<- opcode: QUERY, rcode: NOERROR, id: 36574
;; flags: qr ; QUERY: 1, ANSWER: 0, AUTHORITY: 9, ADDITIONAL: 2
;; QUESTION SECTION:
;; www.customer6.example.       IN      A

;; ANSWER SECTION:

;; AUTHORITY SECTION:
customer6.example.      3600    IN      NS      ns.customer6.example.
customer6.example.      3600    IN      DS      ...
customer6.example.      3600    IN      RRSIG   DS ...
customer5._deleg.example.       1234    IN      NSEC    (
                customer7._deleg.example.  RRSIG NSEC IDELEG )
customer5._deleg.example.       1234    IN      RRSIG   NSEC ...
example.        1234    IN      NSEC    (
                customer5._deleg.example.  NS SOA RRSIG NSEC DNSKEY )
example.        1234    IN      RRSIG   NSEC ...

;; ADDITIONAL SECTION:
ns.customer6.example.   3600    IN      A       203.0.113.1
ns.customer6.example.   3600    IN      AAAA    2001:db8:6::1

;; Query time: 0 msec
;; EDNS: version 0; flags: do ; udp: 1232
;; SERVER: 192.0.2.53
;; WHEN: Tue Feb 25 10:23:53 2025
;; MSG SIZE  rcvd: 744
```

Figure 8: Referral response without deleg

Next to the legacy delegation, the deleg supporting authoritative returns the NSEC(3) RRs needed to show that there was no IDELEG delegation in the referral response in Figure 8.

Querying the zone from Figure 6 for www.customer7.example. A, will return the following referral response:

```
;; ->>HEADER<<- opcode: QUERY, rcode: NOERROR, id: 9456
;; flags: qr ; QUERY: 1, ANSWER: 0, AUTHORITY: 7, ADDITIONAL: 2
;; QUESTION SECTION:
;; www.customer7.example.       IN      A

;; ANSWER SECTION:

;; AUTHORITY SECTION:
customer7.example.      3600    IN      NS      ns.customer5.example.
customer7.example.      3600    IN      DS      ...
customer7.example.      3600    IN      RRSIG   DS ...
customer7._deleg.example.       3600    IN      CNAME   (
                customer5._deleg.example. )
customer7._deleg.example.       3600    IN      RRSIG   CNAME ...
customer5._deleg.example.       3600    IN      IDELEG   1 (
                ns.customer5.example. alpn=h2,h3
                ipv4hint=198.51.100.5 ipv6hint=2001:db8:5::1 )
customer5._deleg.example.       3600    IN      RRSIG   IDELEG ...

;; ADDITIONAL SECTION:
ns.customer5.example.   3600    IN      A       198.51.100.5
ns.customer5.example.   3600    IN      AAAA    2001:db8:5::1

;; Query time: 0 msec
;; EDNS: version 0; flags: do ; udp: 1232
;; SERVER: 192.0.2.53
;; WHEN: Tue Feb 25 10:55:07 2025
;; MSG SIZE  rcvd: 593
```

Figure 9: Aliasing referral response

The delegation of customer7.example. is aliased to the one that is also used by customer5.example. Since both delegations are in the same zone, the authoritative name server for example. returns both the CNAME realising the alias, as well as the IDELEG RRset which is the target of the alias in Figure 9. In other cases a returned CNAME or IDELEG RR in AliasMode may need further chasing by the resolver.

With unsigned zones, only if an IDELEG delegation exists, the IDELEG RRset (or CNAME) will be present in the authority section of referral responses. If there is no IDELEG delegation, then the IDELEG RRset (or CNAME) is simply absent from the authority section and the referral response is indistinguishable from an non deleg supportive authoritative.

4.1. Registration of authoritative name server support

When the authority section of a referral response contains an IDELEG RRset or a CNAME record on the IDELEG delegation name, or valid NSEC(3) RRs showing the non-existence of such an IDELEG or CNAME RRset, then IDELEG supporting resolver SHOULD register that the contacted authoritative name server supports deleg for the duration indicated by the TTL for that IDELEG, CNAME or NSEC(3) RRset, adjusted to the resolver's TTL boundaries (Section 4 of [RFC8767]), but only if it is longer than any already registered duration.

For example, in Figure 7, the IDELEG RRset at the IDELEG delegation point has TTL 3600. The resolver should register that the contacted authoritative name server supports deleg for (at least) 3600 seconds (one hour). All subsequent queries to that authoritative name server SHOULD NOT include deleg queries to be send in parallel.

If the authority section contains more than one RRset making up the IDELEG delegation, then the RRset with the longest TTL MUST be taken to determine the duration for which deleg support is registered.

For example, in Figure 9, both a CNAME and an IDELEG RRset for the IDELEG delegation are included in the authority section. The longest TTL must be taken for deleg support registration, though because the TTL of both RRsets is 3600, it does not matter in this case.

If an proof of non-existence of an IDELEG delegation is returned in the authority section of a referral response, IDELEG supporting resolvers SHOULD register that the authoritative name server that returned that response supports IDELEG for the duration of the maximum of the TTL of the NSEC(3) covering the name of the IDELEG delegation and the minimum rdata field of the SOA for the zone, adjusted to the boundaries for TTL values that the resolver has (Section 4 of [RFC8767]), but only if it is longer than any already registered duration.

4.2. Support detection with unsigned zones

An IDELEG RRset on an IDELEG delegation point, with an IDELEG RR in AliasMode, aliasing to the root zone, MUST be interpreted to mean that the legacy delegation information MUST be used to follow the referral. All service parameters for such AliasMode (aliasing to the root) IDELEG RRs on the IDELEG delegation point, MUST be ignored.

For example, such an IDELEG RRset registered on the wildcard below the _deleg label on the apex of a zone, can signal that legacy DNS referrals MUST be used for both signed and _unsigned_ zones:

```
$ORIGIN example.
@                 IN  SOA   ns zonemaster ...
*._deleg   86400  IN  IDELEG 0 .
customer1._deleg  IN  IDELEG 1 ( ns.customer1
                                  ipv4hint=198.51.100.1,203.0.113.1
                                  ipv6hint=2001:db8:1::1,2001:db8:2::1
                                )
customer3._deleg  IN  CNAME _dns.ns.operator1
```

Figure 10: Wildcard IDELEG delegation to control duration of detected support

Resolvers SHOULD register that an authoritative name server supports deleg, if such an IDELEG RRset is returned in the authority section of referral responses, for the duration of the TTL if the IDELEG RRset, adjusted to the resolver's TTL boundaries (Section 4 of [RFC8767]), but only if it is longer than any already registered duration. Note that this will also be included in referral responses for unsigned zones, which would otherwise not have signalling of deleg support by the authoritative name server.

Also, signed zones need fewer RRs to indicate that no IDELEG delegation exists. The wildcard expansion already shows the closest encloser (i.e. _deleg.<apex>), so only one additional NSEC(3) is needed to show non-existence of the queried-for name below the closest encloser.

This method of signalling that the legacy delegation MUST be used, is RECOMMENDED.

4.3. Deregistering authoritative name server support

If the resolver knows that the authoritative name server supports deleg, _and_ a DNSSEC-signed zone is being served, then all referrals SHOULD contain either an IDELEG delegation, or NSEC(3) records showing that the delegation does not exist. If a referral is returned that does not contain an IDELEG delegation nor an indication that it does not exist, then the resolver MAY deregister that the authoritative server supports IDELEG and MUST send an additional deleg query to find the delegation (or denial of its existence) (see also Resolving without authoritative name server support (Section 5)).

5. Resolving without authoritative name server support

In two cases, resolvers do *not* need to do additional processing.

1. As part of the processing a recursive resolver does, it learns where the zone boundaries are in the DNS name tree. If the triggering query name is already known to be the apex of a zone, then no further delegation point probing will need to be done for this name (subject to the TTL of this information).

2. If a resolver encounters a non-referral response, then no delegations are involved and no further delegation probing is needed for this name (subject to the TTL of this information).

If a resolver encounters an referral that does not contain an IDELEG delegation nor an indication that it does not exist, then the resolver MUST find out whether the queried zone contains IDELEG delegations at all.

1. For DNSSEC signed zones, a direct query for the IDELEG delegation information suffices (see The deleg query (Section 5.1)).

2. For unsigned zones, if unknown, the presence of the _deleg label at the zone's apex needs to be detected in addition to a direct query for the IDELEG delegation information (see _deleg label presence (Section 5.2)).

Sub section Summary (Section 5.3) contains an overview summarizing sub sections Section 5.1 and Section 5.2

5.1. The deleg query

The deleg query name is constructed by concatenating the first label below the part that the triggering query name has in common with the target zone, a _deleg label and the name of the target zone. For example if the triggering query is www.customer.example. and the target zone example., then the deleg query name is customer._deleg.example. For another example, if the triggering query is www.faculty.university.example. and the target zone faculty.university.example. then the deleg query name is www._deleg.faculty.university.example.

DNAME, CNAME and IDELEG in AliasMode processing happens as before, though note that when following an IDELEG RR in AliasMode the target RR type is SVCB (see Section 2). The eventual deleg query response, after following all redirections caused by DNAME, CNAME and AliasMode IDELEG RRs, has three possible outcomes:

1. An IDELEG RRset in ServiceMode is returned in the response's answer section containing the delegation for the subzone.

The IDELEG RRs in the RRset MUST be used to follow the referral. The TargetName data field in the IDELEG RRs in the RRset MUST be used as the names for the name servers to contact for the subzone, and the ipv4hint and ipv6hint parameters MUST be used as the IP addresses for the TargetName in the same IDELEG RR.

The NS RRset and glue, from the referral response, MUST NOT be used, but the signed DS record (or NSEC(3) records indicating that there was no DS) MUST be used in linking the DNSSEC authentication chain as which would conventionally be done with DNSSEC as well.

2. The deleg query name does not exist (NXDOMAIN).

   There is no IDELEG delegation for the subzone, and the referral response for the legacy delegation MUST be processed as would be done with legacy DNS and DNSSEC processing.

   When the query was for an DNSSEC signed zone, the NXDOMAIN response will expose whether or not the _deleg label exists at the zone’s apex. If it does not exist, this should be registered for the duration of the maximum of the TTL of the NSEC(3) covering the _deleg label and the minimum rdata field of the SOA for the zone, adjusted to the boundaries for TTL values that the resolver has (Section 4 of [RFC8767]). For this period, deleg queries to this zone MUST NOT be made (see also Section 5.2).

3. The deleg query name does exist, but resulted in a NOERROR no answer response (also known as a NODATA response).

   If the legacy query, did result in a referral for the same number of labels as the subdomain that the deleg query was for, then there was no IDELEG delegation for the subzone, and the referral response for the legacy delegation MUST be processed as would be done with legacy DNS and DNSSEC processing.

   Otherwise, the subzone may be more than one label below the delegating zone.

   A new deleg query MUST be spawned, matching the zone cut of the initial referral response. For example if the triggering query is www.university.ac.example. and the target zone example., and the legacy response contained an NS RRset for university.ac.example., then the deleg query name is university.ac._deleg.example. The response to the new deleg query MUST be processed as described above, as if it was the initial deleg query.

If the legacy query was sent minimised and needs a followup query, then parallel to that followup query a new deleg query will be sent, adding a single label to the previous deleg query name. For example if the triggering query is www.university.ac.example. and the target zone is example. and the minimised legacy query name is ac.example. (which also resulted in a NOERROR no answer response), then the deleg query to be sent along in parallel with the followup legacy query will become university.ac._deleg.example. Processing of the responses of those two new queries will be done as described above.

5.2. _deleg label presence

Absence of the _deleg label in a zone is a clear signal that the zone does not contain any IDELEG delegations. Recursive resolvers SHOULD NOT send any additional deleg queries for zones for which it is known that it does not contain the _deleg label at the apex. The state regarding the presence of the _deleg label within a resolver can be "unknown", "known not to be present", or "known to be present".

The state regarding the presence of the _deleg label can be deduced from the response of the deleg query, if the target zone is signed with DNSSEC. *No* additional queries are needed. If the target zone is unsigned, the procedure as described in the remainder of this section SHOULD be followed.

5.2.1. Test _deleg label presence (unsigned zones only)

When the presence of a _deleg label is "unknown", a _deleg presence test query SHOULD be sent in parallel to the next query for the unsigned target zone (though not when the target name server is known to support _deleg, which is discussed in Authoritative name server support (Section 4)). The query name for the test query is the _deleg label prepended to the apex of zone for which to test presence, with query type NS.

The testing query can have three possible outcomes:

1. The _deleg label does not exist within the zone, and an NXDOMAIN response is returned.

   The non-existence of the _deleg label MUST be registered for the duration indicated by the "minimum" RDATA field of the SOA resource record in the authority section, adjusted to the boundaries for TTL values that the resolver has (Section 4 of [RFC8767]). For the period the non-existence of the _deleg label is cached, the label is "known not to be present" and the resolver SHOULD NOT send any (additional) deleg queries.

2. The _deleg label does exist within the zone but contains no data. A NOERROR response is returned with no RRs in the answer section.

   The existence of the _deleg name MUST be cached for the duration indicated by the "minimum" RDATA field of the SOA resource record in the authority section, adjusted to the resolver's TTL boundaries (Section 4 of [RFC8767]). For the period the existence of the empty non-terminal at the _deleg label is cached, the label is "known to be present".

3. The _deleg label does exist within the zone, but is a delegation. A NOERROR legacy referral response is returned with an NS RRset in the authority section.

   The resolver MUST record that the zone does not have valid IDELEG delegations deployed for the duration indicated by the NS RRset's TTL value, adjusted to the resolver's TTL boundaries (Section 4 of [RFC8767]). For the period indicated by the NS RRset's TTL value, the zone is considered to *not* to have valid IDELEG delegations, and MUST NOT send any (additional) deleg queries.

5.3. Summary

Table Table 1 provides an overview of when additional queries, following a non-IDELEG referral response, are sent.

	auth support	_deleg presence		<sub>._deleg.<apex> IDELEG	_deleg.<apex> A
1	Yes	*			
2	No	No			
3	No	Yes		X	
4	No	Unknown		X	only for unsigned zones

Table 1: Additional queries to an non-IDELEG referral response

The two headers on the left side of the table mean the following:

auth support:

Whether or not the target authoritative server supports incremental deleg. "Yes" means support is registered and "No" means support is not registered.

_deleg presence:
Whether or not the _deleg label is present in the target zone (and thus incremental delegations) "Yes" means it is present, "No" means it is not present, "Unknown" means this is not yet known and "*" means it doesn't matter.

On the right side of the table are the additional follow-up queries, to be sent in response to non-IDELEG referrals. The _deleg presence test query (most right column) only needs to be sent to unsigned target zones, because its non-existence will be shown in the NSEC(3) records showing the non-existence of the incremental delegation (second from right column).

If the query name is the same as the apex of the target zone, or the response is a non-referral response, no additional queries need to be sent. Otherwise:

1. If the authoritative name server supports IDELEG (Row 1), no additional queries need to be sent.

2. If the _deleg label is known not to exist in the target zone (Row 2), no additional queries need to be sent.

3. If the _deleg label is known to exist in the target zone, but there was no authoritative name server support in the referral response (Row 3), an additional deleg query (Section 5.1) needs to be sent.

4. If it is not known whether or not the _delewg label exists, and there was no authoritative name server support in the referral response (Row 4), an additional deleg query (Section 5.1) needs to be sent. If the target zone is unsigned, presence of the _deleg label needs to be tested explicitly as well (Section 5.2).

5.3.1. Reduced latency

Latency can be reduced with one round trip when the deleg query is sent in parallel to the legacy query. This will induce more additional queries since authoritative name server support (eliminating the need to send deleg queries), is detected from the legacy query.

6. Priming queries

Some zones, such as the root zone, are targeted directly from hints files. Information about which authoritative name servers and with capabilities, MAY be provided in an IDELEG RRset directly at the _deleg label under the apex of the zone. Priming queries from an deleg supporting resolver, MUST send an _deleg.<apex> IDELEG query in parallel to the legacy <apex> NS query and process the content as if it was found through an referral response.

7. Limitations

The presence of the _deleg label in the delegation information reduces that maximum length of a domain name for a zone to 248 bytes. Note that names within zones are not limited and can still be 255 bytes.

8. How does the protocol described in this draft meet the requirements

This section will discuss how deleg meets the requirements for a new delegation mechanism as described in [I-D.ietf-deleg-requirements-02]

* H1. DELEG must not disrupt the existing registration model of domains.

  The existing zone structure including the concept of delegations from a parent zone to a child zone is left unchanged.

* H2. DELEG must be backwards compatible with the existing ecosystem.

  The new delegations do not interfere with legacy software.

  The behavior of deleg supporting resolvers includes a fallback to NS records if no IDELEG delegation is present (See Section 5.1).

* H3. DELEG must not negatively impact most DNS software.

  Deleg introduces a new RR type. Software that parses zone file format needs to be changed to support the new type. Though unknown type notation [RFC3597] can be used in some cases if no support for the new RR type is present. Existing authoritatives can serve IDELEG zones (though less efficiently), existing signers can sign IDELEG zones, existing diagnostic tools can query IDELEG zones. Non-recursive DNSSEC validators can operate independently from (possibly legacy) recursive resolvers.

* H4. DELEG must be able to secure delegations with DNSSEC.

IDELEG delegations as described in this document are automatically secured with DNSSEC (if the parent zone is signed). A replacement for DS records is described in [I-D.homburg-deleg-incremental-dnssec].

* H5. DELEG must support updates to delegation information with the same relative ease as currently exists with NS records.

  IDELEG delegations are affected by TTL like any other DNS record.

* H6. DELEG must be incrementally deployable and not require any sort of flag day of universal change.

  IDELEG zones can be added without upgrading authoritatives. IDELEG zones still work with old resolvers and validators. Basically any combination of old and new should work, though with reduced efficiency for some combinations.

* H7. DELEG must allow multiple independent operators to simultaneously serve a zone.

  Deleg allows for multiple IDELEG records. This allows multiple operators to serve the zone.

* S1. DELEG should facilitate the use of new DNS transport mechanisms

  New transports are already defined for the DNS mode of SVCB ([RFC9461]). The same parameters are used for IDELEG.

* S2. DELEG should make clear all of the necessary details for contacting a service

  Most of the needed SVCB parameters are already defined in existing standards. The exception is a replacement for the DS records, which is described in [I-D.homburg-deleg-incremental-dnssec].

* S3. DELEG should minimize transaction cost in its usage.

  Assuming Qname-minimisation, there are no extra queries needed in most cases if the authoritative name server has deleg support. The exception is when the parent zone is not signed and has no IDELEG records. In that case, one extra query is needed when the parent zone is first contacted (and every TTL).

  Additional queries may be needed to resolve aliases.

* S4. DELEG should simplify management of a zone’s DNS service.

Zone management can be simplified using the alias mode of IDELEG. This allows the zone operator to change the details of the delegation without involving the parent zone.

Draft [I-D.homburg-deleg-incremental-dnssec] defines the dnskeyref parameter which offers the same simplification for DNSSEC delegations.

* S5. DELEG should allow for backward compatibility to the conventional NS-based delegation mechanism.

  NS records and glue can be extracted from the IDELEG record assuming no aliasing is used.

  The ds parameter in [I-D.homburg-deleg-incremental-dnssec] has the same value as the rdata of a DS record.

* S6. DELEG should be extensible and allow for the easy incremental addition of new delegation features after initial deployment.

  SVCB-style records are extensible by design.

* S7. DELEG should be able to convey a security model for delegations stronger than currently exists with DNSSEC.

  IDELEG delegations are protected by DNSSEC, unlike NS records at the parent zone.

9. Implementation Status

Note to the RFC Editor: please remove this entire section before publication.

We are using RR type code 65280 for experiments.

Jesse van Zutphen has built a proof of concept implementation supporting IDELEG delegations as specified in a previous version of this document [I-D.homburg-deleg-incremental-deleg-00] for the Unbound recursive resolver as part of his master thesis for the Security and Network Engineering master program of the University of Amsterdam [JZUTPHEN]. Jesse’s implementation has been adapted to query for the IDELEG RR types (with code point 65280). This version is available in the ideleg branch of the NLnetLabs/unbound github repository [IDELEG4UNBOUND]. Note that this implementation does not yet respond to support in the authoritative Authoritative name server support (Section 4), and also does not yet follow AliasMode IDELEG RRs.

The ldns DNS library and tools software has been extended with support for IDELEG, which is available in the features/ideleg branch of the NLnetLabs/ldns github repository [IDELEG4LDNS]. This includes support for IDELEG in the DNS lookup utility drill, as well as in the DNSSEC zone signer ldns-signzone and all other tools and examples included with the ldns software.

Wouter Petri has built a proof of concept support for IDELEG in the NSD authoritative name server software as part of a research project for the Security and Network Engineering master program of the University of Amsterdam [WPETRI]. The source code of his implementation is available on github [IDELEG4NSD].

Wouter’s implementation is serving the ideleg.net. domain, containing a variety of different IDELEG delegations, for evaluation purposes. We are planning to provide information about the deployment, including what software to evaluate these delegations, at http://ideleg.net/ (http://ideleg.net/), hopefully before the IETF 122 in Bangkok (https://datatracker.ietf.org/meeting/122/proceedings).

10. Security Considerations

Deleg moves the location of referral information to a unique location that currently exists. However, as this is a new approach, thought must be given to usage. There must be some checks to ensure that the registering a _deleg subdomain happens at the time the domain is provisioned. The same care needs to be addressed when a domain is de-provisioned that the _deleg is removed. This is similar to what happens to A/AAAA glue records for NS records deployed in parent zones.

While the recommendation is to deploy DNSSEC with deleg, it is not mandatory. However, using deleg with unsigned zones can create possibilities of domain hijackings. This could be hard to detect when not speaking directly to the authoritative name server. This risk of domain hijacking is not expected to increase significantly compared to the situation without deleg.

There are bound to be other considerations.

11. IANA Considerations

11.1. IDELEG RR type

IANA is requested to update the "Resource Record (RR) TYPEs" registry under the "Domain Name System (DNS) Parameters" registry group as follows:

TYPE	Value	Meaning	Reference
IDELEG	TBD	Delegation	[this document]

Table 2

11.2. _deleg Node Name

Per [RFC8552], IANA is requested to add the following entry to the DNS "Underscored and Globally Scoped DNS Node Names" registry:

RR Type	_NODE NAME	Reference
IDELEG	_deleg	[this document]

Table 3: Entry in the Underscored and Globally Scoped DNS Node Names registry

12. References

12.1. Normative References

[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>.

[RFC3597] Gustafsson, A., "Handling of Unknown DNS Resource Record (RR) Types", RFC 3597, DOI 10.17487/RFC3597, September 2003, <https://www.rfc-editor.org/rfc/rfc3597>.

[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>.

[RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <https://www.rfc-editor.org/rfc/rfc7858>.

[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>.

[RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <https://www.rfc-editor.org/rfc/rfc8484>.

[RFC8767] Lawrence, D., Kumari, W., and P. Sood, "Serving Stale Data to Improve DNS Resiliency", RFC 8767, DOI 10.17487/RFC8767, March 2020, <https://www.rfc-editor.org/rfc/rfc8767>.

[RFC9250] Huitema, C., Dickinson, S., and A. Mankin, "DNS over Dedicated QUIC Connections", RFC 9250, DOI 10.17487/RFC9250, May 2022, <https://www.rfc-editor.org/rfc/rfc9250>.

[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>.

[RFC9461] Schwartz, B., "Service Binding Mapping for DNS Servers", RFC 9461, DOI 10.17487/RFC9461, November 2023, <https://www.rfc-editor.org/rfc/rfc9461>.

[STD13] Internet Standard 13, <https://www.rfc-editor.org/info/std13>. At the time of writing, this STD comprises the following:

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

Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

12.2. 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.homburg-deleg-incremental-deleg-00] Homburg, P., van Zutphen, J., and W. Toorop, "Incrementally Deployable Extensible Delegation for DNS",

Work in Progress, Internet-Draft, draft-homburg-deleg-incremental-deleg-00, 8 July 2024, <https://datatracker.ietf.org/doc/html/draft-homburg-deleg-incremental-deleg-00>.

[I-D.homburg-deleg-incremental-dnssec]
Homburg, P. and W. Toorop, "Incrementally Deployable DNSSEC Delegation", Work in Progress, Internet-Draft, draft-homburg-deleg-incremental-dnssec-00, 16 January 2025, <https://datatracker.ietf.org/doc/html/draft-homburg-deleg-incremental-dnssec-00>.

[I-D.ietf-deleg-requirements-02]
Lawrence, Lewis, E., Reid, J., and T. Wicinski, "Problem Statement and Requirements for an Improved DNS Delegation Mechanism abbrev: DNS DELEG Requirements", Work in Progress, Internet-Draft, draft-ietf-deleg-requirements-02, 12 October 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-deleg-requirements-02>.

[I-D.ietf-dnsop-ns-revalidation]
Huque, S., Vixie, P. A., and W. Toorop, "Delegation Revalidation by DNS Resolvers", Work in Progress, Internet-Draft, draft-ietf-dnsop-ns-revalidation-09, 27 February 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-dnsop-ns-revalidation-09>.

[I-D.ietf-tls-esni]
Rescorla, E., Oku, K., Sullivan, N., and C. A. Wood, "TLS Encrypted Client Hello", Work in Progress, Internet-Draft, draft-ietf-tls-esni-24, 20 March 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-esni-24>.

[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>.

[I-D.wesplaap-deleg]
April, T., paek, P., Weber, R., and Lawrence, "Extensible Delegation for DNS", Work in Progress, Internet-Draft, draft-wesplaap-deleg-02, 18 February 2025, <https://datatracker.ietf.org/doc/html/draft-wesplaap-deleg-02>.

[IDELEG4LDNS]
Toorop, W., "A proof of concept support for IDELEG in the ldns DNS library and tools", n.d., <https://github.com/NLnetLabs/ldns/tree/features/ideleg>.

[IDELEG4NSD]
Petri, W., "A proof of concept support for IDELEG in the NSD authoritative name server software", n.d., <https://github.com/WP-Official/nsd>.

[IDELEG4UNBOUND]
van Zutphen, J. and P. Homburg, "A proof of concept implementation of incremental deleg", n.d., <https://github.com/NLnetLabs/unbound/tree/ideleg>.

[JZUTPHEN] van Zutphen, J., "Extensible delegations in DNS Recursive resolvers", n.d., <https://nlnetlabs.nl/downloads/publications/extensible-deleg-in-resolvers_2024-07-08.pdf>.

[RFC8552] Crocker, D., "Scoped Interpretation of DNS Resource Records through "Underscored" Naming of Attribute Leaves", BCP 222, RFC 8552, DOI 10.17487/RFC8552, March 2019, <https://www.rfc-editor.org/rfc/rfc8552>.

[WPETRI] Petri, W., "Extensible delegations in authoritative nameservers", n.d., <https://nlnetlabs.nl/downloads/publications/extensible-delegations-in-authoritative-nameservers_2025-02-09.pdf>.

Acknowledgments

The idea to utilize SVCB based RRs to signal capabilities was first proposed by Tim April in [I-D.tapril-ns2].

The idea to utilize SVCB for IDELEG delegations (and also the idea described in this document) emerged from the DNS Hackathon at the IETF 118. The following participants contributed to this brainstorm session: Vandan Adhvaryu, Roy Arends, David Blacka, Manu Bretelle, Vladimr unt, Klaus Darilion, Peter van Dijk, Christian Elmerot, Bob Halley, Philip Homburg, Shumon Huque, Shane Kerr, David C Lawrence, Edward Lewis, George Michaelson, Erik Nygren, Libor Peltan, Ben Schwartz, Petr paek, Jan Velk and Ralf Weber

Authors’ Addresses

Philip Homburg
NLnet Labs

Email: philip@nlnetlabs.nl

Tim Wicinski
Cox Communications
Email: tjw.ietf@gmail.com

Jesse van Zutphen
University of Amsterdam
Email: Jesse.vanZutphen@os3.nl

Willem Toorop
NLnet Labs
Email: willem@nlnetlabs.nl