
Stream: Internet Engineering Task Force (IETF)
RFC: [9615](#)
Updates: [7344](#), [8078](#)
Category: Standards Track
Published: July 2024
ISSN: 2070-1721
Authors: P. Thomassen N. Wisiol
deSEC, Secure Systems Engineering deSEC, Technische Universität Berlin

RFC 9615

Automatic DNSSEC Bootstrapping Using Authenticated Signals from the Zone's Operator

Abstract

This document introduces an in-band method for DNS operators to publish arbitrary information about the zones for which they are authoritative, in an authenticated fashion and on a per-zone basis. The mechanism allows managed DNS operators to securely announce DNSSEC key parameters for zones under their management, including for zones that are not currently securely delegated.

Whenever DS records are absent for a zone's delegation, this signal enables the parent's registry or registrar to cryptographically validate the CDS/CDNSKEY records found at the child's apex. The parent can then provision DS records for the delegation without resorting to out-of-band validation or weaker types of cross-checks such as "Accept after Delay".

This document establishes the DS enrollment method described in Section 4 of this document as the preferred method over those from Section 3 of RFC 8078. It also updates RFC 7344.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc9615>.

Copyright Notice

Copyright (c) 2024 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
1.1. Terminology	4
1.2. Requirements Notation	4
2. Updates to RFCs	5
3. Signaling	5
3.1. Chain of Trust	5
3.2. Signaling Names	5
4. Bootstrapping a DNSSEC Delegation	5
4.1. Signaling Consent to Act as the Child's Signer	6
4.1.1. Example	6
4.2. Validating CDS/CDNSKEY Records for DNSSEC Bootstrapping	6
4.2.1. Example	7
4.3. Triggers	8
4.4. Limitations	8
5. Operational Recommendations	9
5.1. Child DNS Operator	9
5.2. Parental Agent	9
6. Security Considerations	10
7. IANA Considerations	10

8. References	10
8.1. Normative References	10
8.2. Informative References	11
Acknowledgements	12
Authors' Addresses	12

1. Introduction

Securing a DNS delegation for the first time requires that the child's DNSSEC parameters be conveyed to the parent through some trusted channel. While the communication conceptually has to occur between the parent registry and the DNSSEC key holder, what that means exactly and how communication is coordinated traditionally depends on the relationship the child has with the parent.

A typical situation is that the key is held by the child DNS operator; thus, the communication often involves this entity. In addition, depending on the circumstances, it may also involve the registrar, possibly via the registrant (for details, see [Appendix A](#) of [\[RFC7344\]](#)).

As observed in [\[RFC7344\]](#), these dependencies often result in a manual process that is susceptible to mistakes and/or errors. In addition, due to the annoyance factor of the process, involved parties may avoid the process of getting a DS record set (RRset) published in the first place.

To alleviate these problems, automated provisioning of DS records has been specified in [\[RFC8078\]](#). It is based on the parental agent (registry or registrar) fetching DNSSEC key parameters from the CDS and CDNSKEY records ([\[RFC7344\]](#)) located at the child zone's apex, and validating them somehow. This validation can be done using the child's existing DNSSEC chain of trust if the objective is to update an existing DS RRset (such as during key rollover). However, when bootstrapping a DNSSEC delegation, the child zone has no existing DNSSEC validation path, so other means to ensure the CDS/CDNSKEY records' legitimacy must be found.

Due to the lack of a comprehensive DNS-innate solution, either out-of-band methods have been used so far to complete the chain of trust, or cryptographic validation has been entirely dispensed with, in exchange for weaker types of cross-checks such as "Accept after Delay" ([Section 3.3](#) of [\[RFC8078\]](#)). [\[RFC8078\]](#) does not define an in-band validation method for enabling DNSSEC.

This document aims to close this gap by introducing an in-band method for DNS operators to publish arbitrary information about the zones for which they are authoritative, in an authenticated manner and on a per-zone basis. The mechanism allows managed DNS operators to securely announce DNSSEC key parameters for zones under their management. The parent can then use this signal to cryptographically validate the CDS/CDNSKEY RRsets found at an insecure child zone's apex and, upon success, secure the delegation.

While applicable to the vast majority of domains, the protocol does not support certain edge cases, such as excessively long child zone names, or DNSSEC bootstrapping for domains with in-domain nameservers only (see [Section 4.4](#)).

DNSSEC bootstrapping is just one application of the generic signaling mechanism specified in this document. Other applications might arise in the future, such as publishing operational metadata or auxiliary information that the DNS operator likes to make known (e.g., API endpoints for third-party interaction).

Readers are expected to be familiar with DNSSEC [\[BCP237\]](#).

1.1. Terminology

This section defines the terminology used in this document.

CDS/CDNSKEY: This notation refers to CDS and/or CDNSKEY, i.e., one or both.

Child: See [Section 7](#) of [\[RFC9499\]](#).

Child DNS operator: The entity that maintains and publishes the zone information for the child DNS.

Parent: See [Section 7](#) of [\[RFC9499\]](#).

Parental agent: The entity that has the authority to insert DS records into the parent zone on behalf of the child. (It could be the registry, registrar, a reseller, or some other authorized entity.)

Signaling domain: A domain name constructed by prepending the label `_signal` to a hostname taken from a delegation's NS RRset. There are as many signaling domains as there are distinct NS targets.

Signaling name: The labels that are prefixed to a signaling domain in order to identify a signaling type and a child zone's name (see [Section 3.2](#)).

Signaling record: A DNS record located at a signaling name under a signaling domain. Signaling records are used by the child DNS operator to publish information about the child.

Signaling type: A signal type identifier, such as `_dsboot` for DNSSEC bootstrapping.

Signaling zone: The zone that is authoritative for a given signaling record.

1.2. Requirements Notation

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.

2. Updates to RFCs

The DS enrollment methods described in [Section 3](#) of [RFC8078] are less secure than the method described in [Section 4](#) of this document. Therefore, child DNS operators and parental agents wishing to use CDS/CDNSKEY records for initial DS enrollment **SHOULD** support the authentication protocol described here.

In order to facilitate publication of signaling records for the purpose of DNSSEC bootstrapping (see [Section 4.1](#)), the first bullet ("Location") of [Section 4.1](#) of [RFC7344] is removed.

3. Signaling

This section describes the general mechanism by which a child DNS operator can publish an authenticated signal about a child zone. Parental agents (or any other party) can then discover and process the signal. Authenticity is ensured through standard DNSSEC validation.

3.1. Chain of Trust

If a child DNS operator implements this specification, each signaling zone **MUST** be signed and be validatable by the parental agent (i.e., have a valid publicly resolvable DNSSEC chain of trust). This is typically achieved by securely delegating each signaling zone.

For example, when publishing a signal that relates to a child zone with NS records `ns1.example.net` and `ns2.example.org`, the child DNS operator needs to ensure that the parental agent has a valid DNSSEC chain of trust for the zone(s) that are authoritative for the signaling domains `_signal.ns1.example.net` and `_signal.ns2.example.org`.

3.2. Signaling Names

To publish information about the child zone in an authenticated fashion, the child DNS operator **MUST** publish one or more signaling records at a signaling name under each signaling domain.

Signaling records **MUST** be accompanied by RRSIG records created with the corresponding signaling zone's key(s). The type and contents of these signaling records depend on the type of signal.

The signaling name identifies the child and the signaling type. It is identical to the child name (with the final root label removed), prefixed with a label containing the signaling type.

4. Bootstrapping a DNSSEC Delegation

When the child zone's CDS/CDNSKEY RRsets are used for setting up initial trust, they need to be authenticated. This is achieved by copublishing the child's CDS/CDNSKEY RRsets as an authenticated signal as described in [Section 3](#). The parent can discover and validate it, thus transferring trust from the child DNS operator nameservers' chain of trust to the child zone.

This protocol is not intended for updating an existing DS RRset. For this purpose, the parental agent can validate the child's CDS/CDNSKEY RRsets directly, using the chain of trust established by the existing DS RRset (Section 4 of [RFC7344]).

4.1. Signaling Consent to Act as the Child's Signer

To confirm its willingness to act as the child's delegated signer and authenticate the child's CDS/CDNSKEY RRsets, the child DNS operator **MUST** copublish them at the corresponding signaling name under each signaling domain, excluding those that would fall within the child domain (Section 3.2). For simplicity, the child DNS operator **MAY** also copublish the child's CDS/CDNSKEY RRsets under signaling domains within the child domain, although those signaling domains are not used for validation (Section 4.2).

Unlike the CDS/CDNSKEY RRsets at the child's apex, a signaling RRset **MUST** be signed with the corresponding signaling zone's key(s). Its contents **MUST** be identical to the corresponding RRset published at the child's apex.

Existing use of CDS/CDNSKEY records was specified at the child apex only (Section 4.1 of [RFC7344]). This protocol extends the use of these record types to non-apex owner names for the purpose of DNSSEC bootstrapping. To exclude the possibility of semantic collision, there **MUST NOT** be a zone cut at a signaling name.

4.1.1. Example

For the purposes of bootstrapping the child zone `example.co.uk` with NS records `ns1.example.net`, `ns2.example.org`, and `ns3.example.co.uk`, the required signaling domains are `_signal.ns1.example.net` and `_signal.ns2.example.org`.

In the zones containing these domains, the child DNS operator authenticates the CDS/CDNSKEY RRsets found at the child's apex by copublishing them as CDS/CDNSKEY RRsets at the names:

```
_dsboot.example.co.uk._signal.ns1.example.net
_dsboot.example.co.uk._signal.ns2.example.org
```

These RRsets are signed with DNSSEC just like any other zone data.

Publication of signaling records under the in-domain name `_signal.ns3.example.co.uk` is not required.

4.2. Validating CDS/CDNSKEY Records for DNSSEC Bootstrapping

To validate a child's CDS/CDNSKEY RRset for DNSSEC bootstrapping, the parental agent, knowing both the child zone name and its NS hostnames, **MUST** execute the following steps:

Step 1: verify that the child has no DS records published at the parent and that at least one of its nameservers is outside the child domain;

- Step 2: query the CDS/CDNSKEY RRset at the child zone apex directly from each of the authoritative servers as determined by the delegation's (parent-side) NS RRset, without caching;
- Step 3: query the CDS/CDNSKEY RRset located at the signaling name under each signaling domain (except those falling within the child domain) using a trusted DNS resolver and enforce DNSSEC validation;
- Step 4: check (separately by record type) that all RRsets retrieved in Steps 2 and 3 have equal contents;

If the above steps succeed without error, the CDS/CDNSKEY RRsets are successfully verified, and the parental agent can proceed with the publication of the DS RRset under the precautions described in [Section 5](#) of [RFC8078].

The parental agent **MUST** abort the procedure if an error condition occurs, in particular:

- in [Step 1](#): the child is already securely delegated or has in-domain nameservers only;
- in [Step 2](#): any failure during the retrieval of the CDS/CDNSKEY RRset located at the child apex from any of the authoritative nameservers;
- in [Step 3](#): any failure to retrieve the CDS/CDNSKEY RRsets located at the signaling name under any signaling domain, including failure of DNSSEC validation, or unauthenticated data (AD bit not set);
- in [Step 4](#): inconsistent responses (for at least one of the types), including an RRset that is empty in one of Steps 2 or 3, but non-empty in the other.

4.2.1. Example

To verify the CDS/CDNSKEY RRsets for the child `example.co.uk`, the parental agent (assuming that the child delegation's NS records are `ns1.example.net`, `ns2.example.org`, and `ns3.example.co.uk`)

1. checks that the child domain is not yet securely delegated;
2. queries the CDS/CDNSKEY RRsets for `example.co.uk` directly from `ns1.example.net`, `ns2.example.org`, and `ns3.example.co.uk` (without caching);
3. queries and validates the CDS/CDNSKEY RRsets located at (see [Section 3.2](#); `ns3.example.co.uk` is ignored because it is in-domain)

```
_dsboot.example.co.uk._signal.ns1.example.net
_dsboot.example.co.uk._signal.ns2.example.org
```

4. checks that the CDS/CDNSKEY RRsets retrieved in Steps 2 and 3 agree across responses.

If all of these steps succeed, the parental agent can proceed to publish a DS RRset as indicated by the validated CDS/CDNSKEY RRset.

As in-domain signaling names do not have a chain of trust at bootstrapping time, the parental agent does not consider them during validation. Consequently, if all NS hostnames are in-domain, validation cannot be completed and DS records are not published.

4.3. Triggers

Parental agents **SHOULD** trigger the procedure described in [Section 4.2](#) once one of the following conditions is fulfilled:

- The parental agent receives a new or updated NS RRset for a child;
- The parental agent receives a notification indicating that the child wishes to have its CDS/CDNSKEY RRset processed;
- The parental agent encounters a signaling record during a proactive, opportunistic scan (e.g., daily queries of signaling records for some or all of its delegations);
- The parental agent encounters a signaling record during an NSEC walk or when parsing a signaling zone (e.g., when made available via AXFR by the child DNS operator);
- Any other condition deemed appropriate by local policy.

Timer-based trigger mechanisms (such as scans) exhibit undesirable properties with respect to processing delay and scaling; on-demand triggers (like notifications) are preferable. Whenever possible, child DNS operators and parental agents are thus encouraged to use them, reducing both delays and the amount of scanning traffic.

Most types of discovery (such as daily scans of delegations) are based directly on the delegation's NS RRset. In this case, these NS names can be used as is by the bootstrapping algorithm ([Section 4.2](#)) for querying signaling records.

Some discovery methods, however, do not imply reliable knowledge of the delegation's NS RRset. For example, when discovering signaling names by performing an NSEC walk or zone transfer of a signaling zone, the parental agent **MUST NOT** assume that a nameserver under whose signaling domain a signaling record appears is actually authoritative for the corresponding child.

Instead, whenever a list of "bootstrappable domains" is obtained by means other than directly from the parent, the parental agent **MUST** ascertain that the delegation actually contains the nameserver hostname seen during discovery, and ensure that signaling-record queries are only made against the proper set of nameservers as listed in the child's delegation from the parent.

4.4. Limitations

As a consequence of [Step 3](#) in [Section 4.2](#), DS bootstrapping does not work for fully in-domain delegations, as no preexisting chain of trust to the child domain is available during bootstrapping. (As a workaround, one can add an out-of-domain nameserver to the initial NS RRset and remove it once bootstrapping is completed. Automation for this is available via CSYNC records, see [\[RFC7477\]](#).)

Fully qualified signaling names must be valid DNS names. Label count and length requirements for DNS names ([Section 3.1](#) of [\[RFC1035\]](#)) imply that the protocol does not work for unusually long child domain names or NS hostnames.

5. Operational Recommendations

5.1. Child DNS Operator

It is possible to add CDS/CDNSKEY records and corresponding signaling records to a zone without the domain owner's explicit knowledge. To spare domain owners from being caught off guard by the ensuing DS changes, child DNS operators following this practice are advised to make that transparent, such as by informing the domain owner during zone creation (e.g., in a GUI) or by notifying them via email.

When transferring a zone to another DNS operator, the old and new child DNS operators need to cooperate to achieve a smooth transition, e.g., by using the multi-signer protocols described in [\[RFC8901\]](#). If all else fails, the domain owner might have to request the removal of all DS records and have the transfer performed insecurely (see [\[INSEC\]](#)).

Signaling domains **SHOULD** be delegated as standalone zones, so that the signaling zone's apex coincides with the signaling domain (such as `_signal.ns1.example.net`). While it is permissible for the signaling domain to be contained in a signaling zone of fewer labels (such as `example.net`), a zone cut ensures that bootstrapping activities do not require modifications of the zone containing the nameserver hostname.

Once a child DNS operator determines that specific signaling record sets have been processed (e.g., by seeing the result in the parent zone), they are advised to remove them. This will reduce the size of the signaling zone and facilitate more efficient bulk processing (such as via zone transfers).

5.2. Parental Agent

In order to ensure timely DNSSEC bootstrapping of insecure domains, stalemate situations due to mismatch of stale cached records ([Step 4](#) of [Section 4.2](#)) need to be avoided. It is thus **RECOMMENDED** that queries into signaling domains with an (initially) empty resolver cache be performed, or that some other method for retrieving fresh data from authoritative servers be used.

It is also **RECOMMENDED** that QNAME minimization [\[RFC9156\]](#) be used when resolving queries for signaling records to guard against certain attacks (see [Section 6](#)).

6. Security Considerations

The DNSSEC bootstrapping method introduced in this document is based on the approaches described in [Section 3](#) of [\[RFC8078\]](#), but adds authentication to the CDS/CDNSKEY concept. Its security level is therefore strictly higher than that of existing approaches described in that document (e.g., "Accept after Delay"). Apart from this general improvement, the same Security Considerations apply as in [\[RFC8078\]](#).

The level of rigor in [Section 4.2](#) is needed to prevent publication of an ill-conceived DS RRset (authorized only under a subset of NS hostnames). This ensures, for example, that an operator in a multi-homed setup cannot enable DNSSEC unless all other operators agree.

In any case, as the child DNS operator has authoritative knowledge of the child's CDS/CDNSKEY records, it can readily detect fraudulent provisioning of DS records.

In order to prevent the parents of nameserver hostnames from becoming a single point of failure for a delegation (both in terms of resolution availability and for the trust model of this protocol), diversifying the path from the root to the child's nameserver hostnames is advisable. For example, different and independently operated TLDs may be used for each one.

If QNAME minimization [\[RFC9156\]](#) is not used when querying for signaling records, an upstream parent of a signaling domain will see those CDS/CDNSKEY queries and could respond with an authoritative answer signed with its own key, instead of sending the referral. Enabling QNAME minimization reduces the attack surface for such forgery.

7. IANA Considerations

IANA has added the following entries to the "Underscored and Globally Scoped DNS Node Names" registry [\[RFC8552\]](#):

RR Type	_NODE NAME	Reference
CDS	_signal	RFC 9615
CDNSKEY	_signal	RFC 9615

Table 1

8. References

8.1. Normative References

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

- [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/info/rfc2119>>.
- [RFC7344] Kumari, W., Gudmundsson, O., and G. Barwood, "Automating DNSSEC Delegation Trust Maintenance", RFC 7344, DOI 10.17487/RFC7344, September 2014, <<https://www.rfc-editor.org/info/rfc7344>>.
- [RFC7477] Hardaker, W., "Child-to-Parent Synchronization in DNS", RFC 7477, DOI 10.17487/RFC7477, March 2015, <<https://www.rfc-editor.org/info/rfc7477>>.
- [RFC8078] Gudmundsson, O. and P. Wouters, "Managing DS Records from the Parent via CDS/CDNSKEY", RFC 8078, DOI 10.17487/RFC8078, March 2017, <<https://www.rfc-editor.org/info/rfc8078>>.
- [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/info/rfc8174>>.
- [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/info/rfc8552>>.
- [RFC9156] Bortzmeyer, S., Dolmans, R., and P. Hoffman, "DNS Query Name Minimisation to Improve Privacy", RFC 9156, DOI 10.17487/RFC9156, November 2021, <<https://www.rfc-editor.org/info/rfc9156>>.
- [RFC9499] Hoffman, P. and K. Fujiwara, "DNS Terminology", BCP 219, RFC 9499, DOI 10.17487/RFC9499, March 2024, <<https://www.rfc-editor.org/info/rfc9499>>.

8.2. Informative References

- [BCP237] Best Current Practice 237, <<https://www.rfc-editor.org/info/bcp237>>. At the time of writing, this BCP comprises the following:
 - Hoffman, P., "DNS Security Extensions (DNSSEC)", BCP 237, RFC 9364, DOI 10.17487/RFC9364, February 2023, <<https://www.rfc-editor.org/info/rfc9364>>.
- [INSEC] Hardaker, W., "Intentionally Temporarily Degraded or Insecure", Work in Progress, Internet-Draft, draft-hardaker-dnsop-intentionally-temporary-insec-01, 21 October 2021, <<https://datatracker.ietf.org/doc/html/draft-hardaker-dnsop-intentionally-temporary-insec-01>>.
- [RFC8901] Huque, S., Aras, P., Dickinson, J., Vcelak, J., and D. Blacka, "Multi-Signer DNSSEC Models", RFC 8901, DOI 10.17487/RFC8901, September 2020, <<https://www.rfc-editor.org/info/rfc8901>>.

Acknowledgements

Thanks to Brian Dickson, Ondřej Caletka, John R. Levine, Christian Elmerot, Oli Schacher, Donald Eastlake, Libor Peltan, Warren Kumari, Scott Rose, Linda Dunbar, Tim Wicinski, Paul Wouters, Paul Hoffman, Peter Yee, Benson Muite, Roman Danyliw, Éric Vyncke, and Joe Abley for reviewing draft proposals and offering comments and suggestions.

Thanks also to Steve Crocker, Hugo Salgado, and Ulrich Wisser for early-stage brainstorming.

Authors' Addresses

Peter Thomassen

deSEC, Secure Systems Engineering
Berlin
Germany
Email: peter@desec.io

Nils Wisiol

deSEC, Technische Universität Berlin
Berlin
Germany
Email: nils@desec.io