DNSOP Working Group
Internet-Draft
Intended status: Standards Track
Expires: 22 June 2026

Y. Thessalonikefs
W. Toorop
NLnet Labs
R. Arends
ICANN

19 December 2025

dry-run DNSSEC
draft-ietf-dnsop-dry-run-dnssec-00

Abstract

This document describes a method called "dry-run DNSSEC" that allows for testing DNSSEC deployments without affecting the DNS service in case of DNSSEC errors. It accomplishes that by introducing new DS Type Digest Algorithms that when used in every record of a DS RRset, referred to as dry-run DS, signal to validating resolvers that dry-run DNSSEC is used for the zone. DNSSEC errors are then reported with DNS Error Reporting, but any bogus responses to clients are withheld. Instead, validating resolvers fallback from dry-run DNSSEC and provide the response that would have been answered without the presence of the dry-run DS. A further EDNS option is presented for clients to opt-in for dry-run DNSSEC errors and allow for end-to-end DNSSEC testing.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 22 June 2026.

Copyright Notice

Table of Contents

1. Introduction

DNSSEC was introduced to provide DNS with data origin authentication and data integrity. This brought quite an amount of complexity and fragility to the DNS which in turn still hinders general adoption. When an operator decides to adopt DNSSEC on an existing insecure zone there is no way to realistically check that DNS resolution will not break for the zone.

Recent efforts that improve troubleshooting DNS and DNSSEC include Extended DNS Errors [RFC8914] and DNS Error Reporting [RFC9567]. The former defines error codes that can be attached to a response as EDNS options. The latter introduces a way for resolvers to report those error codes to the zone operators.

This document describes a method called "dry-run DNSSEC" that builds upon the two aforementioned efforts and provides measurable feedback about DNSSEC resolution health to operators by enabling production testing of a DNSSEC zone. This is accomplished by introducing new DS Type Digest Algorithms. The zone operator signs the zone and makes sure that every DS record in the published DS RRset on the parent side use dry-run DS Type Digest Algorithm(s).

Validating resolvers that don’t support the DS Type Digest algorithms ignore it as per [RFC6840], Section 5.2. Validating resolvers that do support dry-run DNSSEC make use of [RFC8914] and [RFC9567] to report any DNSSEC errors to the zone operator. If a DNSSEC validation error was due to dry-run DNSSEC, validation falls back to insecure as the reply to the client.

This allows real world testing with resolvers that support dry-run DNSSEC by reporting DNSSEC feedback, without breaking DNS resolution for the domain under test.

2. 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.

dry-run DS The DS RRset with dry-run DS Type Digest Algorithm(s) that signals dry-run DNSSEC for the delegation. All DS records in the RRset MUST use a dry-run DS Type Digest Algorithm.

dry-run zone A zone that is DNSSEC signed but uses a dry-run DS to signal the use of the dry-run DNSSEC method.

dry-run parent zone A zone that supports dry-run DNSSEC for its

delegation; that is support for publishing the dry-run DS.

dry-run resolver A validating resolver that supports dry-run DNSSEC.

wet-run client A client that has opted-in to receive the actual DNSSEC errors from the upstream validating resolver instead of the insecure answers.

3. Overview

Dry-run DNSSEC builds upon three previous experiences namely DMARC [RFC7489], Root Key Trust Anchor Sentinel [RFC8509] and Signaling Trust Anchor Knowledge [RFC8145]. The former enabled email operators to verify their configuration with real email servers by getting DMARC reports and understanding the impact on email delivery their configuration would have before committing to enable DMARC. Experience with the latter two showed that with only a small, up to date resolver population, the signaling is already quite substantial.

Dry-run DNSSEC offers zone operators the means to test newly signed zones and a turn-key action to conclude testing and commit to the tested DNSSEC records. Operators that want to use dry-run DNSSEC SHOULD support [RFC9567] and have a reporting agent in place to receive the error reports.

The only change from normal operations when signing a zone with dry-run DNSSEC is to not publish the real DS RRset on the parent but publish the dry-run DS instead. See Section 4 for more information on the dry-run DS itself, and Section 5 on the parent-child communication for the dry-run DS.

[RFC9567] is used for invalid answers and it can generate reports for errors in dry-run DNSSEC zones. This helps with monitoring potential DNS breakage when testing a DNSSEC configuration for a zone. This is also the main purpose of dry-run DNSSEC.

The newly signed zone is publicly deployed but DNSSEC configuration errors cannot break DNS resolution yet. DNS Error Reports can pinpoint potential issues back to the operator. When the operator is confident that the DNSSEC configuration under test does not introduce DNS breakage, the turn-key action to conclude testing and commit to the signed zone is to replace the dry-run DS with the real DS RRset on the parent zone.

3.1. DNSSEC validation of a dry-run zone

Validating resolvers that don’t support the DS Type Digest algorithm ignore it as per [RFC6840], Section 5.2. Dry-run resolvers are signaled to treat the zone as a dry-run zone. Dry-run resolvers SHOULD support [RFC9567] in order to report possible errors back to the operators.

Valid answers as a result of dry-run validation yield authentic data (AD) responses and clients that expect the AD flag can already benefit from the transition.

Invalid answers yield the insecure response that would have been answered when no dry-run DS would have been present in the parent, instead of SERVFAIL. This is not proper data integrity but the delegation SHOULD NOT be considered DNSSEC signed at this point.

Dry-run resolvers MAY store the dry-run validation status if they want to support end-to-end testing as discussed in Section 3.4.

3.1.1. Inconsistencies in the dry-run DS

If a dry-run DS consists of multiple DS records and not all of them use a dry-run DS Type Digest algorithm, the DS records with a dry-run DS Type Digest algorithm MUST be ignored by dry-run resolvers. This means that in this case, DNSSEC validation continues only with the non dry-run DS records.

3.1.2. Use of aggressive negative caching

Aggressive negative caching [RFC8198] needs an explicit mention since DNSSEC faults there can lead to valid answers that could potentially mask underlying NSEC(3) issues.

Dry-run resolvers that support aggressive negative caching, upon synthesizing an answer in a dry-run zone, SHOULD hold off using the synthesized answer and instead issue an explicit query for the record in question. If the reply that comes back is different, i.e., the synthesized answer would prove that the record does not exist whereas the explicit query comes back with the record itself, this means that there lies an issue with negative records. A report SHOULD be generated using the Extended DNS Error code TBD_nsec and the answer to the explicit upstream query SHOULD be used instead of the synthesized one.

3.2. Fallback behavior

In case of validation errors with the dry-run DS, dry-run resolvers fallback to the insecure state of the zone. Dry-run resolvers MAY store the dry-run validation status if they want to support end-to-end testing as discussed in Section 3.4.

3.3. NOERROR report

Dry-run DNSSEC relies on DNS Error Reporting [RFC9567] to report resolution errors back to the zone operators. DNS Error Reporting solely addresses the reporting of DNS errors but it does not give any guarantees that DNS Error Reporting aware resolvers are resolving the zone. This raises a concern especially for dry-run DNSSEC where absence of error reports needs to translate to a positive signal that no DNSSEC errors were encountered.

To solve this, dry-run DNSSEC introduces the NOERROR report. The NOERROR report is sent from the resolver when no error was encountered during dry-run DNSSEC validation and notifies the reporting agent of the resolver’s presence.

As with [RFC9567], Section 4 the resolver will cache the reporting agent reply and dampen the number of NOERROR report queries.

The NOERROR report is using the Extended DNS Error code TBD_no.

3.3.1. Constructing the NOERROR Query

The QNAME for the NOERROR report query follows the same semantics as with [RFC9567], Section 6.1.1 and is constructed by concatenating the following elements:

* A label containing the string "_er".
* The decimal value "0" in a single DNS label as the QTYPE is not relevant for the NOERROR report.
* The list of non-null labels representing the apex of the query name that triggered this report.
* The decimal value of TBD_no in a single DNS label as the Extended DNS Error.
* A label containing the string "_er".
* The agent domain. The agent domain as received in the EDNS0 Report-Channel option set by the authoritative server.

As with [RFC9567], Section 6.1.1 if the QNAME of the report query exceeds 255 octets, it MUST NOT be sent.

The apex is specifically used as the query name for resolvers to only send one NOERROR report (if applicable) per zone and for the monitoring agents to differentiate between different zones they are configured with.

3.4. Opt-in end-to-end DNSSEC testing

For further end-to-end DNS testing, a new EDNS0 option code TBD_w (Wet-Run DNSSEC) is introduced that a client can send along with a query to a validating resolver. This signals dry-run resolvers that the client has opted-in to DNSSEC errors for dry-run zones. Dry-run resolvers that support opt-in MUST respond with the dry-run DNSSEC error, if any, and MUST attach the same EDNS0 option code TBD_w in the response to mark the error response as coming from a dry-run zone.

Dry-run resolvers that support opt-in MUST cache the DNSSEC status of the dry-run validation next to the actual DNSSEC status. This enables cached answers to both regular and opt-in clients, similar to cached answers to clients with and without the CD flag set.

Additional Extended DNS Errors can still be attached in the error response by the validating resolver as per [RFC8914].

Dry-run resolvers that do not support opt-in MUST ignore the TBD_w EDNS0 option and MUST NOT attach the TBD_w EDNS0 option code in their replies.

4. Signaling

Signaling to dry-run resolvers that a delegation uses dry-run DNSSEC happens naturally with the DS RRset returned from the parent zone by specifying new DS Digest Type Algorithm(s).

Each real algorithm has a potential dry-run equivalent. Since this is an attribute for all available DS Digest Type Algorithms, the most significant bit of the DS Digest Type Algorithm is used to signal dry-run when that bit is set.

Resolvers that do not support dry-run DNSSEC and have no knowledge of the introduced DS Digest Type Algorithms ignore them as per [RFC6840], Section 5.2.

4.1. Discussion from IETF 114

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

This is addressed feedback as a result of IETF 114. We keep it here for future reference while the document is advancing.

4.1.1. Burn a bit for dry-run DS Digest Type Algorithms

* Viktor Dukhovni:
  - Saner than variable variant.
  - Hash algorithms are introduced exceedingly rarely, symmetric hashes are very stable.
  - No evidence that SHA2 will be compromised in the next 100 years; we may have SHA3 at some point but little demand.
* Peter Thomassen:
  - Better to sacrifice a bit than variable length. Also for post quantum crypto, in response to Paul Hoffman below, even if keys are large the hash value will have a constant length.
* Libor Peltan: (mailing list)
  - Only a few code points in use now, it seems viable.

4.1.2. Use a single DS Digest Type Algorithm for dry-run

* Need to encode the actual algorithm and data in the DS record; results in variable length DS record for a single algorithm.
* May hinder adoption due to EPP checks/requirements (known record length for each algorithm).
* Mark Andrews:
  - Variable length will be needed for private algorithm types so we may as well support it here.
* Paul Hoffman:
  - Recommends going to variable length to pave the way for post quantum crypto and the surprising length it may need.

5. Provisioning

This section discusses the communication between a dry-run DNSSEC zone and the parent domain and the procedures that need to be in place in order for the parent to publish a dry-run DS for the delegation. Most of the burden falls with the parent zone since they have to understand the delegation's intent for use of dry-run DNSSEC. If the parent does not accept DS records, they need to provide a means so that the child can mark the provided DNSKEY(s) as dry-run DNSSEC. This can be achieved either by a flag on the parent's interface, or their willingness to accept and inspect DS records, that accompany DNSKEY records, for use of the DRY-RUN DS Type Digest Algorithm. The case of CDS/CDNSKEY is discussed below.

5.1. Parent zone records

The only change that needs to happen for dry-run DNSSEC is for the parent to be able to publish the dry-run DS. If the parent accepts DS records from the child, the child needs to provide the dry-run DS. If the parent does not accept DS records and generates the DS records from the DNSKEY, support for generating the dry-run DS record, when needed, should be added to the parent if dry-run DNSSEC is a desirable feature.

When the child zone operator wants to complete the DNSSEC deployment, the parent needs to be notified for the real DS RRset publication.

5.1.1. CDS and CDNSKEY Consideration

CDS works as expected by providing the dry-run DS content for the CDS record. CDNSKEY cannot work by itself; it needs to be accompanied by the aforementioned CDS to signal dry-run DNSSEC for the delegation. Thus, parents that rely only on CDNSKEY need to add support for checking the accompanying CDS record for the DRY-RUN DS Type Digest Algorithm and generating a dry-run DS if such a record is encountered.

Operators of a dry-run child zone are advised to publish both CDS and CDNSKEY so that both cases above are covered.

6. Security Considerations

6.1. DNSSEC status

For the use case of DNSSEC adoption, dry-run DNSSEC disables one of the fundamental guarantees of DNSSEC, data integrity. Bogus answers for expired/invalid data will become insecure answers providing the potentially wrong information back to the requester. This is a feature of this proposal but it also allows forged answers by third parties to affect the zone.

This should be treated as a warning that dry-run DNSSEC is not an end solution but rather a temporarily intermediate test step of a zone going secure.

Thus, a dry-run zone (only dry-run DS on the parent) SHOULD NOT be considered as DNSSEC signed since it does not offer all the DNSSEC guarantees.

6.2. Error reporting

Since dry-run DNSSEC relies heavily on DNS Error Reporting [RFC9567], the same security considerations about the generated error reports apply here as well. Especially the use of TCP or DNS Cookies for the reports, which can be enforced by the monitoring agent to make it harder to falsify the source address of error reports.

7. IANA Considerations

7.1. DRY-RUN DS Type Digest Algorithm

This document defines a new entry in the "Digest Algorithms" registry in the "Delegation Signer (DS) Resource Record (RR) Type Digest Algorithms" registry at https://www.iana.org/assignments/ds-rr-types (https://www.iana.org/assignments/ds-rr-types) :

Value	Description	Status	Reference
128-255	Dry-run DNSSEC	OPTIONAL	[this document]

Table 1

7.2. NOERROR Extended DNS Error

This document defines a new entry in the "Extended DNS Error Codes" registry in the "Domain Name System (DNS) Parameters" registry group at https://www.iana.org/assignments/dns-parameters (https://www.iana.org/assignments/dns-parameters) :

INFO-CODE	Purspose	Reference
TBD_no	NOERROR reporting	[this document]
TBD_nsec	Broken negative cache	[this document]

Table 2

7.3. Wet-Run EDNS0 Option

This document defines a new entry in the "DNS EDNS0 Option Codes (OPT)" registry in the "Domain Name System (DNS) Parameters" registry group at https://www.iana.org/assignments/dns-parameters (https://www.iana.org/assignments/dns-parameters) :

Value	Name	Status	Reference
TBD_wet	Wet-Run DNSSEC	Optional	[this document]

Table 3

8. Acknowledgements

The authors would like to thank the following people, in no particular order, who contributed into shaping this document with their feedback: Libor Peltan, Dave Lawrence, Paul Wouters, Tedius Schrijven, Mats Dufberg, Petr paek, Marco Davids, Mark Andrews, Ben Schwartz, Peter Thomassen, Gavin Brown, Nils Wisiol, Viktor Dukhovni, Paul Hoffman and Lars-Johan Liman.

9. 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/info/rfc2119>.

[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/info/rfc6840>.

[RFC7489] Kucherawy, M., Ed. and E. Zwicky, Ed., "Domain-based Message Authentication, Reporting, and Conformance (DMARC)", RFC 7489, DOI 10.17487/RFC7489, March 2015, <https://www.rfc-editor.org/info/rfc7489>.

[RFC8145] Wessels, D., Kumari, W., and P. Hoffman, "Signaling Trust Anchor Knowledge in DNS Security Extensions (DNSSEC)", RFC 8145, DOI 10.17487/RFC8145, April 2017, <https://www.rfc-editor.org/info/rfc8145>.

[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>.

[RFC8198] Fujiwara, K., Kato, A., and W. Kumari, "Aggressive Use of DNSSEC-Validated Cache", RFC 8198, DOI 10.17487/RFC8198, July 2017, <https://www.rfc-editor.org/info/rfc8198>.

[RFC8509] Huston, G., Damas, J., and W. Kumari, "A Root Key Trust Anchor Sentinel for DNSSEC", RFC 8509, DOI 10.17487/RFC8509, December 2018, <https://www.rfc-editor.org/info/rfc8509>.

[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/info/rfc8914>.

[RFC9567] Arends, R. and M. Larson, "DNS Error Reporting", RFC 9567, DOI 10.17487/RFC9567, April 2024, <https://www.rfc-editor.org/info/rfc9567>.

Appendix A. Implementation Status

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

In the following implementation status descriptions, "dry-run DNSSEC" refers to dry-run DNSSEC as described in this document.

None yet.

Appendix B. Change History

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

* draft-yorgos-dnsop-dry-run-dnssec-00

| Initial public draft.

* draft-yorgos-dnsop-dry-run-dnssec-01

| Document restructure and feedback incorporation from IETF 113.

* draft-yorgos-dnsop-dry-run-dnssec-02

| Document restructure and feedback incorporation from IETF 114;
| mainly:
|
| Use explicit dry-run algorithm types for DS.
|
| Introduce NOERROR reporting.

* draft-yorgos-dnsop-dry-run-dnssec-03

| Shape up NOERROR reporting.
|
| No need for exclusive NOERROR signal from upstream; existence of
| dry-run suffices.
|
| Ask for NOERROR Extended DNS Error.
|
| Remove most IETF 114 feedback sections for better flow of the
| document; kept the discussion about signaling.
|
| Add security considerations for increased validation workload.
|
| Add an explicit fallback behavior section.

* draft-yorgos-dnsop-dry-run-dnssec-04

| Dry-run only specified for insecure zones.
|
| Add complete acknowledgement section covering all feedback thus
| far.
|
| Add explicit section about negative caching.
|
| Burn a bit in the DS Digest Type Algorithm for dry-run.
|
| Specify that all DSes in the DS set must be dry-run.

* draft-ietf-dnsop-dry-run-dnssec-00

| Same as draft-yorgos-dnsop-dry-run-dnssec-04 but resubmitted after
| WG adoption.

Authors' Addresses

Yorgos Thessalonikefs
NLnet Labs
Science Park 400
1098 XH Amsterdam
Netherlands
Email: yorgos@nlnetlabs.nl

Willem Toorop
NLnet Labs
Science Park 400
1098 XH Amsterdam
Netherlands
Email: willem@nlnetlabs.nl

Roy Arends
ICANN
Email: roy.arends@icann.org