DNSOP Working Group
Internet-Draft
Intended status: Standards Track
Expires: 5 September 2022

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

4 March 2022

# dry-run DNSSEC
## draft-yorgos-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 a new DS Type Digest Algorithm that signals to validating resolvers that dry-run DNSSEC is used for the zone. DNSSEC errors are then reported with DNS Error Reporting, but the bogus response is withheld. Instead resolvers fallback from dry-run DNSSEC and provide the response that would have been answered without the presence of a dry-run DS. A further 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 5 September 2022.

## Copyright Notice

Table of Contents



## 1. Introduction

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

This document describes a method called "dry-run DNSSEC" that gives

confidence to operators to adopt DNSSEC by introducing a new DS Type Digest Algorithm. Resolvers that don't support the algorithm continue to treat the delegation as insecure [RFC6840], Section 5.2. Validating resolvers are signaled to treat the delegation as being in an intermediate test step for DNSSEC. Valid answers yield authentic





data (AD) responses. Therefore, clients that expect the AD flag can already profit from the transition. Invalid answers instead of SERVFAIL yield the response that would have been answered when no dry-run DS would have been present in the parent. For zones that had only dry-run DS RRs in the parent, an invalid answer yields an insecure response. This is of course not proper data integrity but the delegation should not be considered DNSSEC signed at this point.

Based on DNS Error Reporting [DNS-ERROR-REPORTING], invalid answers for dry-run DNSSEC errors generate reports in order to monitor potential DNS breakage when changing the DNSSEC configuration for a zone. This is also the main purpose of dry-run DNSSEC.

The signed zone is publicly deployed but DNSSEC configuration errors cannot break DNS resolution yet. DNSSEC health feedback can pinpoint potential issues back to the operator. When the operator is confident that the DNSSEC adoption does not introduce DNS breakage, the real DS record can be published on the parent zone and that concludes the actual DNSSEC deployment.

Dry-run DNSSEC can further be used on already singed zones to test key rollovers. In this case a dry-run DS record for the future key is used next to the current DS record which itself needs to be also presented in the dry-run format. Validating resolvers that understand dry-run DNSSEC first try to validate with a dry-run DS before falling back to real DSes.

For further end-to-end DNS testing, a new EDNS0 option code is introduced that a client can send along with a query to a validating resolver. This signals validating resolvers that the client has opted-in to DNSSEC errors for dry-run delegations. The resolver still uses DNS Error Reporting [DNS-ERROR-REPORTING] for dry-run errors but instead of the insecure answer it provides the client with the SERVFAIL answer, same as with actual DNSSEC. These clients are called "wet-run clients".

## 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 record with the special DS type digest algorithm that signals dry-run DNSSEC for the delegation.

real DS  The actual DS record for the delegation.  Replaces the dry-run DS to complete DNSSEC deployment.





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.

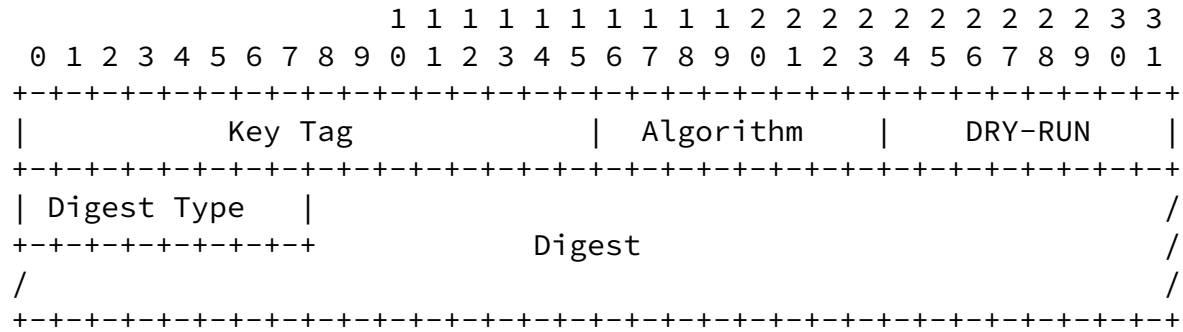
wet-run client  A client that has opted-in to receive the actual DNSSEC errors from the upstream validating resolver instead of the insecure answers.

## 3. Description

TODO

### 3.1. The dry-run DS structure

The dry-run DS record is a normal DS record with updated semantics to allow for dry-run signaling to a validating resolver.  The DS Type Digest Algorithm value MUST be TBD (DRY-RUN).  The first octet of the DS Digest field contains the actual Type Digest Algorithm, followed by the actual Digest:




Validating resolvers encountering such a DS record will know to mark

this delegation as dry-run DNSSEC and extract the actual Type Digest Algorithm and Digest from the dry-run DS Digest field.

Validating resolvers that have no knowledge for the DRY-RUN DS Type Digest Algorithm MUST disregard the DS record as per [RFC6840], Section 5.2.

## 3.2. DNSSEC Error Reporting

The main purpose of the dry-run DNSSEC proposal is to be able to monitor potential DNS breakage when adopting DNSSEC for a zone. The main tool to do that is DNS Error Reporting [DNS-ERROR-REPORTING].

Operators that want to use dry-run DNSSEC SHOULD support DNSSEC Error Reporting and have a reporting agent in place to receive the error reports.





Implementations that support dry-run DNSSEC MUST also support DNSSEC Error Reporting and report any DNSSEC errors for the dry-run zone to the designated report agent.

## 3.3. 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 record. If the parent accepts DS records from the child, the child needs to provide the dry-run DS record. 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 record.

### 3.3.1. 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 record.

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

### 3.4. dry-run DS and real DS coexistence

TODO tldr: for example testing key rollover.

* For ease of implementation and DoS prevention validators SHOULD pick a DS and DNSKEY pair they understand from both the dry-run and real pool of available DSes.
* If dry-run DSes are present, the validator MUST first consider those.
* If real DS is picked by validator, carry on.
* If dry-run DS is picked,
  - If everything OK, secure.
  - If something not OK, should report and fallback to real DS. No insecure answers for this one. It guarantees that the DNSSEC of the zone is not altered.
  - If going back to real DS, the real DS is now cached and no EDER reports for the same dry-run DS should be generated.





### 3.5. wet-run clients

Wet-run clients are clients that send the EDNS0 option code TBD (Wet-Run DNSSEC) when querying a validating resolver. These clients opt-in to receive error responses in case of DNSSEC errors in a dry-run zone. They allow for end-to-end DNSSEC testing in a controlled environment.

Validating resolvers that recognise the option MUST respond with the error that they would normally respond for a DNSSEC zone and MUST attach the same EDNS0 option code TBD in the response to mark the error response as coming from a dry-run zone.

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

## 4. Implementation Notes

TODO tldr; validating resolvers need to keep an additional DNSSEC status for cached records that notes the DNSSEC status for the dry-run part. Responses can then be provided based on the Wet-Run DNSSEC EDNS0 option.

## 5. Security Considerations

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 still 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.

Parent zones that provide signed delegations to child zones should be aware that by using dry-run DNSSEC (e.g., testing a key roll to a stronger algorithm key) they risk the DNSSEC status of the child zones. If the trust chain becomes invalid between parent and child because of dry-run DNSSEC the child zone will be treated as insecure.

## 6. IANA Considerations

### 6.1. DRY-RUN DS Type Digest Algorithm

This document defines a new entry in the "Delegation Signer (DS) Resource Record (RR) Type Digest Algorithms" registry:





| Value | Digest Type | Status | Reference |
|---|---|---|---|
| TBD | DRY-RUN | OPTIONAL | [this document] |

Table 1

## 6.2. Wet-Run EDNS0 Option

This document defines a new entry in the "DNS EDNS0 Option Codes (OPT)" registry on the "Domain Name System (DNS) Parameters" page:

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

Table 2

## 7. Acknowledgements

Martin Hoffmann contributed the idea of using the DS record of an already signed zone also as a dry-run DS in order to facilitate testing key rollovers.

## 8. 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>.

[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>.

[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>.

## 9. Informative References





[DNS-ERROR-REPORTING]

Arends, R. and M. Larson, "DNS Error Reporting", <https://datatracker.ietf.org/doc/html/draft-ietf-dnsop-dns-error-reporting>.

[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>.

## 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.

* TODO

## Appendix B. Change History (to be removed before final publication)

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

| Initial public draft.

## Authors' Addresses

Yorgos Thessalonikefs
NLnet Labs
Science Park 400
1098 XH Amsterdam
Netherlands
Email: george@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