Workgroup: DNSOP Working Group
Internet-Draft:
draft-yorgos-dnsop-dry-run-dnssec-01
Published: 11 July 2022
Intended Status: Standards Track
Expires: 12 January 2023
Authors: Y. Thessalonikefs   W. Toorop   R. Arends
         NLnet Labs   NLnet Labs   ICANN

dry-run DNSSEC

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 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 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

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 publish a newly signed 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 [DNS-ERROR-REPORTING]. 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 gives confidence to operators to adopt DNSSEC by introducing a new DS Type Digest Algorithm. The zone operator signs the zone and makes sure that the DS record published on the parent side uses the specific DS Type Digest Algorithm. Validating resolvers that don't support the DS Type Digest algorithm ignore it as per [RFC6840], Section 5.2. Validating resolvers that do support dry-run DNSSEC make use of [RFC8914] and [DNS-ERROR-REPORTING] to report any DNSSEC errors to the zone operator. If a DNSSEC validation error was due to dry-run DNSSEC, validation restarts by ignoring the dry-run DS in order to give the real DNS response 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.

real DS The actual DS record for the delegation.

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

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 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 [DNS-ERROR-REPORTING] and have a reporting agent in place to receive the error reports.

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

Validating resolvers that don't support the DS Type Digest algorithm ignore it as per [RFC6840], Section 5.2. Validating resolvers that support dry-run DNSSEC are signaled to treat the zone as a dry-run zone. Validating resolvers that support dry-run DNSSEC MUST support [DNS-ERROR-REPORTING].

Valid answers as a result of dry-run validation yield authentic data (AD) responses and 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 not proper data integrity but the delegation should not be considered DNSSEC signed at this point. For zones that had other non-dry-run DS RRs in the parent, validation MUST restart by using those RRs instead.

[DNS-ERROR-REPORTING] 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 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 singed zone is to replace the dry-run DS with the real DS record on the parent zone.

3.1. Use cases

Dry-run DNSSEC can be used to test different DNSSEC scenarios. From adopting DNSSEC for a zone, which is the main goal of this document, to testing experimental DNSSEC configurations and key rollovers. Dry-run resolvers generate error reports in case of validation errors in dry-run zones and they fallback to the non-dry-run part of the zones to complete validation.

3.1.1. DNSSEC adoption

This use case tests DNSSEC adoption for an insecure zone. The zone is signed and a single dry-run DS record is published on the parent. Validation errors yield error reports but invalid answers do not result in SERVFAIL responses to the clients. In the absence of real DS records, resolvers fallback to no DS records for the zone. The zone is treated as insecure, yielding insecure responses of the DNS data.

3.1.2. Experimental DNSSEC configuration

This use case can test a completely different DNSSEC configuration for an already signed zone. The zone is doubly signed and there are at least two DS RRs in the parent zone. Dry-run resolvers try to use the dry-run part of the zone. In case of validation errors they fallback to the real DS and restart validation which may or may not lead to further validation errors depending on the real DNSSEC status of the zone.

3.1.3. Key rollover

As with the experimental case above, but for the benefit of testing a key rollover before actually committing to it. The rollover can be tested by introducing the real DS also as a dry-run DS record as the first step. Normal key rollover procedures can continue by introducing the new key as another dry-run DS record. In case of validation errors, dry-run resolvers fallback to the real DS and restart validation. When testing was successful, the same exact procedure can be followed by replacing the dry-run DS steps with real DSes.

A special key rollover case could be for the root. This can be made possible by specifying the dry-run DS Digest Type in the <DigestType> element in http://data.iana.org/root-anchors/root-anchors.xml or a different way of indicating in the xml file.

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

For further end-to-end DNS testing, a new EDNS0 option code TBD (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 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.

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

4. Signaling

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

The current version of the document describes two different timelines: one where a single DS Digest Type Algorithm is introduced and one where multiple DS Digest Type Algorithms are introduced.

For the single timeline, the algorithm specifies the use of dry-run DNSSEC for the zone. The actual DS Digest Type Algorithm is encoded in the first byte of the RDATA. This results in variable length DS RDATA in relation to the DS Digest Type Algorithm and is discussed further in Section 5. For more information about the record see Section 4.2.

For the multiple timeline, each algorithm has a potential dry-run equivalent. This can be realised by either burning a bit in the DS Digest Type Algorithm, so that all current and future algorithms have a dry-run DNSSEC equivalent, or by explicitly specifying algorithms for each current and future algorithm. The current convention for this document is to specify a new one for SHA-256 only at the moment.

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

4.1. Feedback from IETF 113

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

This is addressed feedback as a result of IETF 113. We keep it here for future reference while the document is advancing.

We currently have settled for augmenting the DS with information regarding dry-run DNSSEC by specifying additional DS Digest Type(s) which does not affect current logic in the name servers and resolvers alike (except from the added support by dry-run resolvers).

4.1.1. Hash is created from DNSKEY (or CDNSKEY)

- Feedback from Gavin Brown from CentralNIC.
- DNSKEYs do have space for flags which are ignored. There was a suggestion to use the flags in the DNSKEY to signal Dry-run, but we do not like it, because it makes the move to actual DNSSEC impossible without also changing the DNSKEY RRset.

4.1.2. Idea: Have a general purpose DS Digest Type for signaling

- From Ben Schwartz
- To avoid polluting the digest type space with all the different ideas.
- Although there are a lot of DS hacks that need to convey information about the delegation, we find the dry-run DS logic to be tightly coupled with the actual DS record. So it is not a hack per say but an addition to the DS information.
- Sure, it will be another draft dependency then. Personally we'd prefer Petr's idea (see below).

4.1.3. Idea from Petr: Normalize the different DS hacks

- There are now several drafts on hold because they want to "misuse" DS for signalling. Petr's proposal: Why not have a range of RR types for which the parent is authoritative (like DS, and what NS should have been).
- This could work for Dry-run, we could have a DDS RR type which would have the same rdata as DS, but then signals Dry-run.
- We like it, but it creates another dependency for all these drafts (including ours) to progress.

4.2. The dry-run DS structure

This is only relevant for the single timeline as described in Section 4.

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:

```
                     1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Key Tag             |  Algorithm    |    DRY-RUN    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Digest Type   |                                               /
+-+-+-+-+-+-+-+-+            Digest                             /
/                                                               /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Dry-run 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.

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 record 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 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 DNSKEYs for use of the DRY-RUN DS Type Digest Algorithm. The case of CDS/CDNSKEY is discussed below.

One issue in the single timeline is that the DS record will have variable length RDATA for the single defined DS Digest Type Algorithm and that could trigger parsing errors on the registrars. This is the main reason for the multiple timeline existence. This is something that could be addressed by the registrars and allow for the single timeline.

5.1. Feedback from IETF 113

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

This is addressed feedback as a result of IETF 113. We keep it here for future reference while the document is advancing.

5.1.1. Registry supports only fixed sized hashes per hash algorithm

- *Feedback from Gavin from CentralNic.
- *We could also have a dry-run hash algorithm per DS algorithm; this is presented in Section 4.
- *Disadvantage burn hash algorithms twice as fast.
- *Registries could also just change this rule for dry-run.

5.2. 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 publication.

5.2.1. 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 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

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.

7. IANA Considerations

7.1. DRY-RUN DS Type Digest Algorithm

The changes needed for either the single timeline or multiple timeline as described in Section 4.

7.1.1. Single timeline

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

7.1.2. Multiple timeline

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

Value	Digest Type	Status	Reference
TBD	SHA-256 DRY-RUN	OPTIONAL	[this document]

Table 2

7.2. 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 3

8. 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.

9. Normative References

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

[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>.

[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>.

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.

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