Network Working Group   O. Gudmundsson
Internet-Draft   CloudFlare
Intended status: Informational   P. Wouters
Expires: April 17, 2016   Red Hat
                                                       October 15, 2015

Managing DS records from parent via CDS/CDNSKEY
draft-ogud-dnsop-maintain-ds-00

Abstract

RFC7344 specifies how DNS trust can be maintained in-band between parent and child. There are two features missing in that specification: initial trust setup and removal of trust anchor. This document addresses both these omissions.

Changing a domain's DNSSEC status can be a complicated matter involving many parties. Some of these parties, such as the DNS operator, might not even be known by all organisations involved. The inability to enable or disable DNSSEC via in-band signalling is seen as a problem or liability that prevents DNSSEC adoption at large scale. This document adds a method for in-band signalling of DNSSEC status changes.

Initial trust is considered a much harder problem, this document will seek to clarify and simplify the initial acceptance policy.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

CDS/CDNSKEY [RFC7344] records are used to signal changes in trust anchors, this is a great way to maintain delegations when the DNS operator has no other way to inform the parent that changes are needed. RFC7344 contains no "delete" signal for the child to tell the parent that it wants to change the DNSSEC security of its domain.

[RFC7344] punted the Initial Trust establishment question and left it to each parent to come up with an acceptance policy.

1.1. Removing DS

This document introduces the delete option for both CDS and CDNSKEY. to allow a child to signal the parent to turn off DNSSEC. When a domain is moved from one DNS operator to another one, sometimes it is necessary to turn off DNSSEC to facilitate the change of DNS operator. Common scenarios include:

1 moving from a DNSSEC operator to a non-DNSSEC capable one or one that does not support the same algorithms as the old one.

2 moving to one that cannot/does-not-want to do a proper DNSSEC rollover.

3 the domain holder does not want DNSSEC.

4 when moving between two DNS operators that use disjoint sets of algorithms to sign the zone, thus algorithm roll can not be performed.

Whatever the reason, the lack of a "remove my DNSSEC" option is turning into the latest excuse as why DNSSEC cannot be deployed.

Turing off DNSSEC reduces the security of the domain and thus should only be done carefully, and that decision should be fully under the child domain's control.

1.2. Introducing DS

The converse issue is how does a child domain instruct the parent it wants to have a DS record added. This problem is not as hard as many have assumed, given a few simplifying assumptions. This document makes the assumption that there are reasonable policies that can be applied and will allow automation of trust introduction.

Not being able to enable trust via an easily automated mechanism is hindering DNSSEC at scale by anyone that does not have automated access to its parent's "registry".

1.3. Notation

When this document uses the word CDS it implies that the same applies to CDNSKEY and vice versa, the only difference between the two records is how information is represented.

When the document uses the word "parent" it implies an entity that is authorized to insert into parent zone information about this child domain. Which entity this is exactly does not matter. It could be

the Registrar or Reseller that the child domain was purchased from. It could be the Registry that the domain is registered in when allowed. It could be some other entity when the RRR framework is not used.

We use RRR to mean Registry Registrar Reseller in the context of DNS domain markets.

1.4. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. The Three Uses of CDS

In general there are three operations that a domain wants to influence on its parent:

1 Roll over KSK, this means updating the DS records in the parent to reflect the new set of KSK's at the child. This could be an ADD operation, a Delete operation on one or more records while keeping at least one DS RR, or a full Replace operation

2 Turn off DNSSEC validation, i.e. delete all the DS records

3 Enable DNSSEC validation, i.e. place initial DS RRset in the parent.

Operation 1 is covered in [RFC7344], operations 2 and 3 are defined in this document. In many people's minds, those two later operations carry more risk than the first one. This document argues that 2 is identical to 1 and the final one is different (but not that different).

2.1. The meaning of CDS ?

The fundamental question is what is the semantic meaning of publishing a CDS RRset in a zone? We offer the following interpretation:

"Publishing a CDS or CDNSKEY record signifies to the parent that the child is ready for the corresponding DS records to be synchronized. Every parent or parental agent should have an acceptance policy of these records for the three different use cases involved: Initial DS publication, Key rollover, and Returning to Insecure."

In short, the CDS RRset is an instruction to the parent to modify DS RRset if the CDS and DS RRsets differ. The acceptance policy for CDS in the rollover case is "seeing" according to [RFC7344]. The acceptance policy in the Delete case is just seeing a CDS RRset with the delete operation specified in this document.

3. Enabling DNSSEC via CDS/CDNSKEY

There are number of different models for managing initial trust, but in the general case, the child wants to enable global validation for the future. Thus during the period from the time the child publishes the CDS until the corresponding DS is published is the period that DNS answers for the child could be forged. The goal is to keep this period as short as possible.

One important case is how a 3rd party DNS operator can upload its DNSSEC information to the parent, so the parent can publish a DS record for the child. In this case there is a possibility of setting up some kind of authentication mechanism and submission mechanism that is outside the scope of this document.

Below are some policies that parents can use. These policies assume that the notifications are can be authenticated and/or identified.

3.1. Accept policy via authenticated channel

In this case the parent is notified via UI/API that CDS exists, the parent retrieves the CDS and inserts the DS record as requested, if the request comes over an authenticated channel.

3.2. Accept with extra checks

In this case the parent checks that the source of the notification is allowed to request the DS insertion. The checks could include whether this is a trusted entity, whether the nameservers correspond to the requestor, whether there have been any changes in registration in the last few days, etc, or the parent can send a notification requesting an confirmation.

The end result is that the CDS is accepted at the end of the checks or when the out-of-band confirmation is received.

3.3. Accept after delay

In this case, if the parent deems the request valid, it starts monitoring the CDS records at the child nameservers over period of time to make sure nothing changes. After number of checks,

preferably from different vantage points, the parent accepts the CDS records as a valid signal to update.

3.4. Accept with challenge

In this case the parent instructs the requestor to insert some record into the child domain to prove it has the ability to do so (i.e., it is the operator of the zone).

4. DNSSEC Delete Algorithm

The DNSKEY algorithm registry contains two reserved values: 0 and 255[RFC4034]. The CERT record [RFC4398] defines the value 0 to mean the algorithm in the CERT record is not defined in DNSSEC.

[rfc-editor remove before publication] For this reason, using the value 0 in CDS/CDNSKEY delete operations is potentially problematic, but we propose that here anyway as the risk is minimal. The alternative is to reserve one DNSSEC algorithm number for this purpose. [rfc-editor end remove]

Right now, no DNSSEC validator understands algorithm 0 as a valid signature algorithm, thus if the validator sees a DNSKEY or DS record with this value, it will treat it as unknown. Accordingly, the zone is treated as unsigned unless there are other algorithms present.

In the context of CDS and CDNSKEY records, DNSSEC algorithm 0 is defined and means the entire DS set MUST be removed. The contents of the records MUST contain only the fixed fields as show below.

```
1  CDS 0 0 0

2  CDNSKEY 0 3 0
```

There is no keying material payload in the records, just the command to delete all DS records. This record is signed in the same way as CDS/CDNSKEY is signed.

Strictly speaking the CDS record could be "CDS X 0 X" as only the DNSKEY algorithm is what signals the delete operation, but for clarity the "0 0 0" notation is mandated, this is not a definition of DS Digest algorithm 0. Same argument applies to "CDNSKEY 0 3 0".

Once the parent has verified the CDS/CDNSKEY record and it has passed other acceptance tests, the DS record MUST be removed. At this point the child can start the process of turning DNSSEC off.

5. Security considerations

This document is about avoiding validation failures when a domain moves from one DNS operator to another one. Turing off DNSSEC reduces the security of the domain and thus should only be done as a last resort.

In most cases it is preferable that operators collaborate on the rollover by doing a KSK+ZSK rollover as part of the handoff, but that is not always possible. This document addresses the case where unsigned state is needed.

Users SHOULD keep in mind that re-establishing trust in delegation can be hard and take a long time thus before going to unsigned all options SHOULD be considered.

A parent should ensure that when it is allowing a child to become securely delegated, that it has a reasonable assurance that the CDS/CDNSKEY that is used to bootstrap the security on is visible from a geographically and network topology diverse view. It should also ensure the the zone would validate if the parent published the DS record. A parent zone might also consider sending an email to its contact addresses to give the child a warning that security will be enabled after a certain about of wait time - thus allowing a child administrator to cancel the request.

This document does not introduce any new problems, but like Negative Trust Anchor[I-D.ietf-dnsop-negative-trust-anchors], it addresses operational reality.

6. IANA considerations

This document updates the following IANA registries: "DNS Security Algorithm Numbers"

Algorithm 0 adds a reference to this document.

7. References

7.1. Normative References

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <http://www.rfc-editor.org/info/rfc4034>.

[RFC7344] Kumari, W., Gudmundsson, O., and G. Barwood, "Automating DNSSEC Delegation Trust Maintenance", RFC 7344, DOI 10.17487/RFC7344, September 2014, <http://www.rfc-editor.org/info/rfc7344>.

7.2. Informative References

[I-D.ietf-dnsop-negative-trust-anchors] Ebersman, P., Kumari, W., Griffiths, C., Livingood, J., and R. Weber, "Definition and Use of DNSSEC Negative Trust Anchors", draft-ietf-dnsop-negative-trust-anchors-13 (work in progress), August 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4398] Josefsson, S., "Storing Certificates in the Domain Name System (DNS)", RFC 4398, DOI 10.17487/RFC4398, March 2006, <http://www.rfc-editor.org/info/rfc4398>.

Appendix A. Acknowledgements

This document is generated using the mmark tool that Miek Gieben has developed.

Authors' Addresses

Olafur Gudmundsson
CloudFlare

Email: olafur+ietf@cloudflare.com

Paul Wouters
Red Hat

Email: pwouters@redhat.com