dnsop O. Gudmundsson
Internet-Draft CloudFlare
Intended status: Standards Track P. Wouters
Expires: December 12, 2016 Red Hat
June 10, 2016

Managing DS records from parent via CDS/CDNSKEY
draft-ietf-dnsop-maintain-ds-03

Abstract

RFC7344 specifies how DNS trust can be partially 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 multiple unrelated parties. Some of these parties, such as the DNS operator, might not even be known by all the organizations involved. The inability to disable DNSSEC via in-band signalling is seen as a problem or liability that prevents some DNSSEC adoption at large scale. This document adds a method for in-band signalling of these DNSSEC status changes.

Initial trust is considered in general to be a hard technical problem, this document sets forth reasonable policies that clarify and simplify the initial acceptance policy.

Copyright Notice

Table of Contents

1. Introduction

CDS/CDNSKEY [RFC7344] records are used to signal changes in trust anchors. This is one method to maintain delegations that can be used when the DNS operator has no other way to inform the parent that changes are needed.

[RFC7344] is lacking two different options for full automated operation to be possible. Firstly it did not define a method for the

Initial Trust establishment and left it to each parent to come up with an acceptance policy. Secondly it did not provide a "delete" signal for the child to tell the parent that it wants to remove the DNSSEC security for its domain.

1.1. Introducing a DS record

The big issue is how a child domain instructs the parent that it wants to have a DS record added. This problem can be solved using 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 for DNS hosters that do not have automated access to the "registry" of the child zone's parent.

1.2. Removing a DS Record

This document introduces the delete option for both CDS and CDNSKEY, allowing a child to signal to 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 alternative to doing a proper DNSSEC algorithm rollover due to operational limitations such as software limitations.

2 moving from a DNSSEC operator to a non-DNSSEC capable operator.

3 moving to an operator that cannot/does-not-want to do a proper DNSSEC rollover.

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

5 the domain holder no longer wants DNSSEC enabled.

The lack of a "remove my DNSSEC" option is cited as a reason why some operators cannot deploy DNSSEC, as this is seen as an operational risk.

Turning 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.

1.3. Notation

When this document uses the word CDS it implies that the same applies to CDNSKEY and vice verse. The only differences between the two records is how information is represented, and who calculates the DS digiest.

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

When the document uses the word "parent" it implies an entity that is authorized to insert DS records into the parent zone on behalf of the child domain. Which entity this exactly is 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.

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

2 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.

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

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

2.1. The meaning of the CDS RRset

The semantic meaning of publishing a CDS RRset is interpreted to mean:

"Publishing a CDS or CDNSKEY record signals to the parent that the child desires that the corresponding DS records 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 the DS RRset if the CDS and DS Reset's differ. The acceptance policy for CDS in the rollover case is "seeing" according to [RFC7344]. The acceptance policy in the Delete case is seeing a (validly signed) 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 at the parent 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 third 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 can be verified or authenticated.

3.1. Accept policy via authenticated channel

In this case the parent is notified via authenticated channel UI/API that a CDS/CDNSKEY RRset exists. In the case of a CDS RRset the parent retrieves the CDS and inserts the corresponding DS RRset as requested. In the case of CDNSKEY the parent retrieves the CDNSKEY RRset and calculates the DS record(s).

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 requester, whether there have been any changes in registration in the last few days, etc. The parent can also send a notification requesting a confirmation, for example by sending email to the registrant requesting a confirmation. The end result is that the CDS RRset 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 RRset at the child nameservers over period of time to make sure nothing changes. After some time or after a number of checks, preferably from different vantage points in the network, the parent accepts the CDS RRset as a valid signal to update its DS RRset for this child.

3.4. Accept with challenge

In this case the parent instructs the requester 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.

For this reason, using the value 0 in CDS/CDNSKEY delete operations is potentially problematic, but we propose it here anyway as the risk is minimal. The alternative is to reserve a DNSSEC algorithm number for this purpose.

Right now, no DNSSEC validator understands algorithm 0 as a valid signature algorithm. If a validator sees a DNSKEY or DS record with this algorithm value, it MUST treat it as unknown. Accordingly, the zone is treated as unsigned unless there are other algorithms present. In general the value 0 should never be used in the context of DNSKEY and DS records.

In the context of CDS and CDNSKEY records, DNSSEC algorithm 0 is defined to mean that the entire DS RRset MUST be removed. The contents of the CDS or CDNSKEY RRset MUST contain one RR and only contain the exactly the fields as shown below.

1 CDS 0 0 0

2 CDNSKEY 0 3 0

The keying material payload is represented by a single 0. This record is signed in the same way as regular CDS/CDNSKEY RRset's are signed. This is a change in format from strict interpretation of [RFC7344] and may cause problems with some deployed software.

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. The same argument applies to "CDNSKEY 0 3 0", the value 3 in second field is mandated by RFC4034 section 2.1.2.

Once the parent has verified the CDS/CDNSKEY RRset and it has passed other acceptance tests, the parent MUST remove the DS RRset. After waiting a sufficient amount of time - depending on the parental TTL's - the child can start the process of turning off DNSSEC.

5. Security considerations

This document's main goal is to avoid validation failures when a domain moves from one DNS operator to another. Turning 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 hand-off, but that is not always possible. This document addresses the case where unsigned state is needed to complete a rollover.

Users SHOULD keep in mind that re-establishing trust in delegation can be hard and takes a long time. Before deciding to complete the rollover via an unsigned state, 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 RRset that is used to bootstrap the security is visible from a geographically and topologically diverse view. It SHOULD also ensure that the zone validates correctly if the parent publishes the DS record. A parent zone might also consider sending an email to its contact addresses to give the child zone a warning that security will be enabled after a certain amount of wait time - thus allowing a child administrator to cancel the request.

6. IANA considerations

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

Algorithm 0 adds a reference to this document.

6.1. Promoting RFC7344 to standards track

Experience has shown that CDS/CDNSKEY are useful in the deployment of DNSSEC. [RFC7344] was published as Informational, this document elevates RFC7344 to standards track.

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

[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. We thank number of people that have provided feedback and useful comments including Bob Harold, John Levine, Matthijs Mekking, Dan York, Shane Kerr, Jacques Latour.

Authors' Addresses

Olafur Gudmundsson
CloudFlare

Email: olafur+ietf@cloudflare.com

Paul Wouters
Red Hat

Email: pwouters@redhat.com