dnsop
Internet-Draft
Intended status: Informational
Expires: December 19, 2013

W. Kumari
Google
O. Gudmundsson
Shinkuro Inc.
G. Barwood

June 17, 2013

Automating DNSSEC delegation trust maintenance
draft-kumari-ogud-dnsop-cds-02

Abstract

This document describes a method to allow DNS operators to more easily update DNSSEC Key Signing Keys using DNS as communication channel. This document does not address the initial configuration of trust anchors for a domain.

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 http://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 December 19, 2013.

Table of Contents

1. Introduction

When a DNS operator first signs their zone, they need to communicate their DS record(s) (or DNSKEY(s)) to their parent through some out-of-band method to complete the chain of trust.

Each time the child changes/rolls the key that is represented in the parent, the new and/or deleted key information has to be communicated to the parent and published there. How this information is sent to the parent depends on the relationship the child has with the parent. In many cases this is a manual process, and not an easy one. For each key roll, there may be two interactions with the parent. Any manual process is susceptible to mistakes and/or errors. In addition, due to the annoyance factor of the process, operators may avoid performing key rollovers or skip needed steps to publish the new DS at the parent.

DNSSEC provides data integrity to information published in DNS; thus DNS publication can be used to automate maintenance of delegation information. This document describes a method to automate publication of subsequent DS records, after the initial one has been published.

Readers are expected to be familiar with DNSSEC, including [RFC4033], [RFC4034], [RFC4035], [RFC5011] and [RFC6781].

This document is a compilation of two earlier drafts, draft-barwood-dnsop-ds-publish and draft-wkumari-dnsop-ezkeyroll

This document outlines a technique in which the parent (frequently registrar / registry) periodically (or upon request) polls its signed children and automatically publish new DS records. To a large extent, the procedures this document follows are in [RFC6781] section 4.1.2

This technique is in some ways similar to RFC 5011 style rollovers, but for sub-domains DS records, instead of trust anchors

This technique is designed to be friendly to automated tools, that the tools can perform all the actions needed w/o human intervention, and monitor when it is save to move to next step.

1.1. Requirements notation

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].

1.2. Terminology

In this section we define terms used in the document. There are also definitions we use in section Section 2.2.1.

RRR: This is our shorthand for the common registration model used in many delegation mainly zones, this stands for Registry/ Registrar/Registrant. Registry = the domain the registration takes place in, and also refers to the operator of said registry. Registrar on the other hand deals with "customers", "sells'' registrations and provides support. Registrant is the party "buying'' the registration. In some cases there is the 4th R in this -- Reseller which is under contract with a Registrar allowing it to sell registrations and record them via the Registrar's systems.

2. Background

2.1. DNS delegations

DNS operation consists of delegations of authority. For each delegation there are (most of the time) two parties the parent and child.

In DNS, the parent publishes information about the delegations to the child; for the name-servers it publishes an NS RRset that lists a hint for name-servers that are authoritative for the child. The child also publishes a NS RRset and this set is the authoritative list of name-servers to the child zone.

The second RRset the parent sometime publishes is the DS set. The DS RRset provides information about the key(s) that the child has told the parent it will use to sign its DNSKEY RRset. In DNSSEC trust relationship between zones is provided by the following chain:

parent DNSKEY --> DS --> child DNSKEY.

2.2. Relationship between Parent and Child DNS operator

In the real world, there are many different relationships between the parent and child DNS operators. The type of relationship affects how the child operator communicates with the parent. This section will highlight some of the different situations, but is by no means a complete list.

A domain name holder (child) may operate its own DNS servers or out source the operation. While we use the word parent as a singular, parent can consist of single entity or a composite of many discrete parts that have rules and roles. For example in many of the TLD cases there is the RRR model (Registry, Registrar and Registrant). The Registry operates DNS for the TLD, the Registrars accept registrations and place information into the Registries' database. The Registrant only communicates with the Registrar; frequently the Registry is not allowed to communicate with the Registrant. In that case as far as the registrant is concerned the Registrar == Parent.

Another common case is the enterprise case where an organization may delegate parts of its namespace to be operated by a group that is not the same as operates the enterprises DNS servers. In this case the flow of information is frequently handled in either an ad hoc manner or via some corporate mechanism, this can range from email to fully-automated operation. The word enterprise above is supposed to cover all organization where the domains are not sold on the open market and there is some relationship between the entities.

2.2.1. Roles

Highlighted roles

- Child: "The entity on record that has the delegation of the domain from the parent"
- Parent: "The domain in which the child is registered"
- Child DNS operator: "The entity that maintains and publishes the zone information for the child DNS"
- Parent DNS operator: "The entity that maintains and publishes the zone information for the parent DNS"
- Parental Agent: "The entity that the child has relationship with, to change its delegation information."

Different communication paths:

- Direct/API: The child can change the delegation information via automated/scripted means EPP[RFC5730] used by many TLDs is an example of this. Another example is the web services' programmatic interfaces that Registrars make available to their Resellers.
- User Interface: The Child uses a (web) site set up by the Parental Agent for updating delegation information.
- Indirect: The communication has to be transmitted via out-of-band between two parties, such as email, telephone etc.. This is common when the Child's DNS operator is neither the child itself nor the Registrar for the domain but a third party.
- Multi-step Combinations: The information flows through an intermediary. It is possible, but unlikely, that all the steps are automated via APIs and there are no humans are involved.

In the RRR world, the different parties are frequently from different organizations. In the single enterprise world there are also organizational/geographical/clutural seperations that affect how information flows from a child delegation to the parent.

Due to the complexity of the different roles and interconnections, automation of delegation information has been punted in the past. There have been some proposals to automate this, in order to improve the reliability of the DNS. These proposals have not gained enough traction to become standards.

A prior proposal [cite] suggested that the child send an "update" to the parent via a mechanism similar to Dynamic Update [I-D.auto-cpsync]. . The main issue became: How does the child find the actual parental agent/server to send the update to? While that could have been solved via technical means, the proposal died.

2.3. Solution Space

This document is aimed at the cases in which there is an organizational separation of the child and parent.

In many RRR cases the Registrar and Registry communicate via EPP[RFC5730] and use the EPP DNSSEC extension [RFC5910]. In number of ccTLDs there are other mechanisms in use as well as EPP, but in general there seems to be a movement towards EPP usage when DNSSEC is enabled in the TLD.

A further complication is when the DNS Operation is separate from the Registrant. There are two common cases of this, registrar handles the DNS operation and a third party takes care of the DNS operation. In the case of a third party DNS operator, the Registrant either needs to relay changes in DNS delegation or give the operator access to its registration account. If the Registrar is the DNS operator, life is much easier, as it can inject any delegation changes directly into the Registry data bases. The techniques described below are not needed in the case when Registrar is the DNS operator. To reflect that the Registrant is not always the DNS Operator we will use the word "Child Operator" to describe the party that makes changes in the child zone.

Some parents want the child to express the changes in trust anchors via DS records, while others want to receive DNSKEY records and calculate the DS records themselves. There is no consensus on which method is better; both have good reasons to exist. The proposal below can operate with both models, but the child needs to be aware of the parental policies.

2.4. DNSSEC key change process

After a DNS operator first signs its zone, there is a need to interact with the parent via the registration interface to "upload/ paste-in the zone's DS information". The action of logging in through the registration interface authenticates that the user is authorized to change delegation information published in the parent zone. In the case where "Child Operator" does not have access to the registration account, the Registrant needs to perform the action.

At a later date, the Child Operator may want to publish a new DS record in the parent, either because they are rolling keys, or because they want to publish a stand-by key. This involves performing the same process as before. Furthermore this is a manual process with cut and paste; operational mistakes will happen.

3. CDS (Child DS) record definition

The DS record can only be present at the parent RFC4034 [RFC4034] some other method is needed to automate the expression of what the parental zone DS records contents ought to be. One possibility is to use flags in the DNSKEY record. The SEP bit is an optional bit to indicate that the key is allowed to sign the DNSKEY RRset, and the Parental Agent can calculate DS records based on that. But this fails to meet some operating needs, including the child having no influence what DS digest algorithms are used and DS records can only be published for keys that are in the DNSKEY RRset.

The CDS record can be published in the child zone and gives the child more control of what is published for it in the parental zone. The CDS RRset expresses what the DS RRset SHOULD look like after the change thus it is a "replace" operation, it is up to the consumer of the records to translate that into the appropriate add/delete operations in the registration systems.

3.1. CDS Resource Record Format

The wire and presentation format of the CDS ("Child DS") record is identical to the DS record. IANA has allocated RR code 59 for the CDS record.

No special processing is performed by authoritative servers or by revolvers, when serving or resolving. For all practical purposes CDS is a regular RR type.

3.1.1. Going unsigned

In theory the child can use the CDS to reflect to the parent that it wants DS records removed. This can be accomplished by publishing CDS record with the following contents:

```
@ IN CDS 0 0 0
```

This is a suggestion and its security implications have not been fully examined but if like process like [RFC5011] should be used before this is accepted. It is important that the Child remain signed until the DS record has been removed from the parent and the DS has timed out from all caches.

Note: maybe it is better to register a special DS digest algorithm number for this ?

If the child zone does go unsigned, the Parental Agent should not treat that as intent to go unsigned since that could be an attack. An attacker could spoof unsigned responses to queries from the Parental Agent in an attempt to force a break in the DNSSEC chain.

4. Automating DS maintainance with CDS records

4.1. CDS Publication

CDS records are intended to be "consumed" by delegation trust maintainers, to enable this constraints are placed on how the CDS record as follows:

- Location: "the CDS record MUST be at the child zone apex"

- Signer: "MUST be signed with a key that is represented in both the current DNSKEY and DS RR-set's."

- Continuity: "MUST not break the current delegation if applied"

If any these conditions fail the CDS record MUST be ignored, similarly the absence of CDS record signals "No change" in the current DS set. The use of CDS is optional.

4.2. CDS Consumption

The CDS RRset MAY be used by the Parental Agent to update the DS RRset in the parent zone. The Parental Agent for this uses a tool that understands the CDS signing rules from Section 4.1 thus it is not be able to use a standard validator.

How the Parental Agent gets the CDS record may differ, below are two examples as how this can take place.

Polling The Parental Agent operates a tool that periodically checks each of the children that has a DS record to see if there is a CDS record. If one exits it applies the checks from section X and if the CDS and DS ``differ'' it applies the changes.

Pushing The Parental Agent in its user interface has a button {Fetch DS} when pushed preforms the CDS processing.

In the "Polling" case the Parental Agent may apply additional rules that defer the acceptance of CDS information, these rules include CDS remain in place for some time. For example RFC 5011 [RFC5011] uses

hold down timers that require new keying information to be published for a month before acceptance as new trust anchor. It is up to each "Parent" and "Parental Agent" to publish minimal rules they apply for child to follow in these cases. The rules SHOULD also include the list of understood digest algorithms.

If at least one DS and one CDS records exist, the Parental Agent validates and then copies the contents of the CDS RRset and replaces the entire existing DS set with the new one.

When using CDS to publish its key rollover information it is the child's responsibility to monitor the parent for changes to the DS RRset before performing the next action in the key rollover sequence. What this implies is that the child MUST NOT follow a strict time-line but rather strict sequence of steps with time checks.

4.3. Usage

The Parental Agent SHOULD ensure that old versions of the CDS RRset do not overwrite newer versions, which could occur if the parent performs the checks too frequently. In that case when there is a delay updating the secondary name servers for the child zone. This MAY be accomplished by checking that the signature inception in the RRSIG for CDS is newer and/or the serial number on the child's SOA is greater.

If the CDS RRset does not exist, the parent MUST take no action. Specifically it MUST NOT delete the existing DS RRset.

If the child zone loses the secret key(s) for the zone, and needs to reset the parent DS RRset, this can only be accomplished by an out-of-band mechanism not defined here.

To mitigate situations where a key signing key has been compromised, the Parental Agent MAY take extra security measures, for example informing (by email or other methods) the child zone administrator of the change, or by delaying the acceptance of the new DS RRset for some period of time. However the precise out-of-band measures that a parent zone SHOULD take are outside the scope of this document.

4.4. Parent calculates DS

There are cases where the Parent wants to calculate the DS record for them self due to policy reasons. In this case the Child can still publish a CDS records instructing the parent which DNSKEY's to represent in the DS RRset. This requires publication of future keys in the DNSKEY RRset for the parent to be able to calculate the DS record. The DNS Parent needs to publish guidelines for the children as to what digest algorithms are acceptable in the CDS record.

When the Parent operates in "calculate DS" mode it can operate in one of two modes "full" i.e. it only publishes DS records it calculates from DNSKEY records, and "augment" i.e. it will make sure there are DS records for the digest algorithm(s) it requires(s).

Implications on Parental Agent are that the CDS and DS are not exactly the same after update thus it needs to take that into consideration when checking CDS records. In the RRR case this calculation can take place either at the Registry or the Registrar (as Parental Agent). If the Registry performs the calculation Parental Agent needs to submit DNSKEY records and possibly (C)DS records as well.

5. IANA Considerations

IANA has assigned RR Type code 59 for CDS. This was done for an earlier version of this document (draft-barwood-dnsop-ds-publish).

6. Security Considerations

[This needs more work, suggestions welcome.]

This work is for the normal case, when things go wrong there is only so much that automation can fix.

If child breaks DNSSEC validation by removing all the DNSKEYS that are represented in the DS set its only repair actions are to contact the parent or restore the DNSKEY's in the DS set.

In the event of a compromise of the server or system generating signatures for a zone, an attacker might be able to generate and publish new CDS records. The modified CDS records will be picked up by this technique and so may allow the attacker to extend the effective time of his attack. If there a delay in accepting changes to DS, as in RFC5011, then the attacker needs to hope his activity is not detected before the DS in parent is changed. If this type of change takes place, the child need to contact the parent (possibly via a registrar web interface) and remove any compromised DS keys.

A compromise of the registrar will not be mitigated by this technique, as the "new registrant" can delete/modify the DS records at will.

While it may be tempting, this SHOULD NOT be used for initial enrollment of keys since there is no way to ensure that the initial key is the correct one. If is used, strict rules for inclusion of keys like hold down times, challenge data inclusion etc., ought to be used, along with some kind of challenge mechanism.

The CDS RR type should allow for enhanced security by simplifying process. Since rollover is automated, updating a DS RRset by other means may be regarded as unusual and subject to extra security checks.

If there is a failure in applying changes in child zone to all DNS servers listed in either parent or child NS set it is possible that the Parental agent may get confused either not perform action because it gets different answers on different checks or CDS validation fails. In the worst case Parental Agent performs an action reversing a prior action but after the child signing system decides to take the next step in rollover, resulting in a broken delegation.

7. Acknowledgements

This is by no means the invention of the authors. This idea has been floating around for a long time. This simply documents it for discussion.

We would like to thank: Joe Abley, Roy Arends, Jim Galvin, Cricket Liu, Stephan Lagerholm, Matt Larson, Olaf Kolkman, Suzanne Woolf, Paul Wouters, Wes Hardaker, Doug Barton, Brian Dickinson, Marco Sanz, Tony Finch, Antoin Verschuren, Edward Lewis.

There were a large number of other folk with whom we discussed this, apologies for not remembering everyone.

8. References

8.1. Normative References

[IANA.AS_Numbers]
IANA, "Autonomous System (AS) Numbers", , <http://www.iana.org/assignments/as-numbers>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

8.2. Informative References

[I-D.auto-cpsync] Mekking, W., "Automated (DNSSEC) Child Parent Synchronization using DNS UPDATE", draft-mekking-dnsop-auto-cpsync-01 (work in progress), December 2010.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.

[RFC5011] StJohns, M., "Automated Updates of DNS Security (DNSSEC) Trust Anchors", STD 74, RFC 5011, September 2007.

[RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, August 2009.

[RFC5734] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Transport over TCP", STD 69, RFC 5734, August 2009.

[RFC5910] Gould, J. and S. Hollenbeck, "Domain Name System (DNS) Security Extensions Mapping for the Extensible Provisioning Protocol (EPP)", RFC 5910, May 2010.

[RFC6781] Kolkman, O., Mekking, W., and R. Gieben, "DNSSEC Operational Practices, Version 2", RFC 6781, December 2012.

Appendix A. Changes / Author Notes.

[RFC Editor: Please remove this section before publication]

From 01 to 02

- Major restructuring to facilitate easier discussion
- Lots of comments from DNSOP mailing list incorporated, including making draft DNSKEY/DS neutral, explain different relationships that exists,

- added more people to acks.
- added description of enterprise situations
- Unified on usign Parental Agent over Parental Representative
- Removed redundant text when possible
- Added text to explain what can go wrong if not all child DNS servers are in sync.
- Reference prior work by Matthijs Mekking
- Added text when parent calculates DS from DNSKEY

From - to -1.

- Removed from section .1: "If a child zone has gone unsigned, i.e. no DNSKEY and no RRSIG in the zone, the parental representative MAY treat that as intent to go unsigned. (NEEDS DISCUSSION)." Added new text at end. -- suggestion by Scott Rose 20/Feb/13.
- Added some background on the different DNS Delegation operating situations and how they affect interaction of parties. This moved some blocks of text from later sections into here.
- Number of textual improvements from Stephan Lagerholm
- Added motivation why CDS is needed in CDS definition section
- Unified terminology in the document.
- Much more background

Authors' Addresses

Warren Kumari
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
US

Email: warren@kumari.net

Olafur Gudmundsson
Shinkuro Inc.
4922 Fairmont Av, Suite 250
Bethesda, MD 20814
USA

Email: ogud@ogud.com

George Barwood
33 Sandpiper Close
Gloucester GL2 4LZ
United Kingdom

Email: george.barwood@blueyonder.co.uk