template
Internet-Draft
Intended status: Informational
Expires: August 27, 2013

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

February 25, 2013

Automating DNSSEC delegation trust maintenance
draft-kumari-ogud-dnsop-cds-01

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 August 27, 2013.

Copyright Notice

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. In many cases this is a fairly annoying and manual process. Unfortunately, every time the child rolls their KSK (Key Signing Key) key they have to repeat the process, possibly multiple times. As this is a manual process DNS operators often avoid rolling their keys, as they don't want to have to do go through the annoyance of publishing the new DS records 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

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].

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.

The parent publishes a NS set that is authoritative for the existence of the delegation but is hint to the contents of the NS record, as well as DS record that expresses what DNSKEY records are to be trusted to sign the DNSKEY RRset in the child. The NS in parent is unsigned as it is hint, the NS bit in the NSEC/NSEC3 record is the proof that the delegation exists. The DS record on the other hand is signed.

The child publishes a signed NS record that as it is authoritative for the contents of the NS set. The child on the other hand can not via current DNS mechanisms express all its desires in which DS records to publish.

This document is aimed at the case where there is an organizational separation of the child and parent. In this case there are many different operating situations. A common case is the Registrant/ Registrar/Registry relationship. In this case the parent consists of Registrar and Registry, with different rules on what each can do or not do. To remain operating model neutral we will use the neutral word "Parental Agent" as the entity that uses results of DNS queries to inject delegation changes into the parent zone. The entity that

inserts the changes in the the DNS is called "DNS Publisher"

In many R/R/R cases the Registrar and Registry communicate via EPP[RFC5730] and use the EPP DNSSEC extension [RFC5910].

The "ICANN TLD case" is a common case and we will expand on that here. The registrant registers a domain though a registrar, who then enters information into a database(s), the DNS information (NS, DS and address records) are placed in a database at the registry, and published in the TLD DNS servers. Frequently registrations and subsequent updates take place via web interfaces. When the registrant wants to change NS or DS information it needs to go access the web interface which may take few minutes and many pages to enter the new information. In the ICANN TLD case the Registry operator is by contract not allowed to change the delegation information without the registrar consent, what this means is all changes MUST flow though the registrar. In the context of ICANN TLD's the "Parental Agent" can be assumed to be an registrar, but in other context the "Parental Agent" can be function of the registry.

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 does the DNS operation. In the case of a third party DNS operator the Registrant either needs to relay changes in DNS delegation changes or give the operator access to its registration account. If the Registrar is the DNS operators, 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" to describe the party that makes changes in the child zone.

In some cases Registries want to receive DNSKEY records instead of DS records from as the registry calculates the DS records itself. That operating model constrains what the child can do to automate maintenance of DS records, as the child can not publish a DS record for a key that is not in its DNSKEY RRset. Similarly the Child can not control what digest algorithms are used.

2.2. DNSSEC key change process

After an DNS operator first signs its zone, there is a need to interact with the parent via the registration interface to "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.

Eventually the Child may want to publish a new DS record in the parent, either because they are rolling their keys, or because they want to publish a stand-by key DS record. This involves performing the same process -- logging into the registration interfaces, selecting the domain, finding the link to change DNSSEC information, pasting (or typing) their DS record (often in a non-standard format) and clicking submit. In a real world test, on web interface this took 12 steps and approximately 3 minutes). As humans (especially DNS operators :-)) dislike tedious, repetitive steps they avoid rolling their DNSSEC keys to avoid having to perform this. Furthermore as this is manual process with cut and paste operations mistakes will happen.

3. CDS Record

As the DS record can only be present at the parent some other method is needed to automate the expression of what the parental zone DS records contents ought to be. One possibly is to use flags in 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 has 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 full control of what is published for it in the parental zone.

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 resolvers, when serving or resolving. CDS unlike a DS resides in the child zone.

The CDS record MUST be at the zone apex, and MUST be signed with a key that is represented in the current DNSKEY and DS RRset's. If these conditions are not met the CDS record MUST be ignored.

3.2. CDS Behavior

The CDS RRset MAY be used by the Parental Agent to update the DS RRset in the parent zone

Transfer of the contents of the CDS record can be accomplished in a number of ways. A Parental Agent MAY periodically check the child zone to see if the CDS RRset has changed. The child MAY request that the parent check the CDS set via registration interface, or via some other automated mechanism.

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.

The Child MUST make sure that the CDS RRset is at all times can be validated using a DNSKEY that is referenced from the current DS set in the parent. This can be accomplished by making sure that at all times during a KEY rollover there are either two DS records or two DNSKEY records with SEP bit published in the DNS.

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.

3.2.1. Periodic check by Parental Agent

In this case the Parental Agent will query each child zone that has a DS RRset, looking for CDS RRset

If present the Parental Agent MUST validate [[RFC4035]] the CDS RRset with a DNSKEY that is represented in the current DS RRset in parent. The Parental Agent should submit a request to the DNS Publisher to publish the contents of the CDS RR(s) as the new a DS record(s) for that zone. The Parental Agent SHOULD log the date and time when of this action including the signature initiation time on the CDS record. The DNS Publisher should log if possible the source of the update, user interface/CDS etc.

The Parental Agent SHOULD NOT check more often than . * TTL on the CDS records.

3.3. Usage

The Parental Agent SHOULD ensure that old versions of the CDS RRset do not overwrite newer versions, which can occur the parent performs the checks too frequently. In that case when there is a delay updating 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.

3.3.1. Going unsigned

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

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

This is an suggestion and its security implications have not been fully examined but an RFC11 like process 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 has timed out from caches.

Note: maybe it is better to register a special DS digest algrithm 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. 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).

5. Security Considerations

[This needs a more work, suggestions welcome.]

In the event of a compromise of the server generating signatures for a zone, 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. This can be dealt with by contacting the parent (possibly via a registrar web interface) and removing 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

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 is used, strict rules for inclusion of keys like hold down times, challenge data inclusion etc., ought to be used.

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.

6. 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.

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

7. References

7.1. 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.

7.2. Informative References

[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 - 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 terminolgy 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: warren@kumari.net