Domain Name System Operations	W. Mekking	
Internet-Draft	NLnet Labs	
Intended status: Standards Track	December 21, 2010	
Expires: June 24, 2011		

Automated (DNSSEC) Child Parent Synchronization using DNS UPDATE
draft-mekking-dnsop-auto-cpsync-01

Abstract

This document proposes a way to synchronise existing trust anchors automatically between a child zone and its parent. The protocol can be used for other Resource Records that are required to delegate from a parent to a child such as NS and glue records. The synchronization allows for a third party to be involved, thus the protocol is suitable for both cases, whether you have to communicate to the registry or to the registrar.

Requirements Language

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 RFC 2119 (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.) [RFC2119].

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 June 24, 2011.

Copyright Notice

---

1. Introduction

This memo defines a way to synchronise existing trust anchors automatically between a child zone and its parent. The protocol can be used for other Resource Records that are required to delegate from a parent to a child such as NS and glue records. The synchronization allows for a third party to be involved, thus the protocol is suitable for both cases, whether you have to communicate to the registry or to the registrar.
To create a DNSSEC RFC 4035 (Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, “Protocol Modifications for the DNS Security Extensions,” March 2005.) [RFC4035] chain of trust, child zones must submit their DNSKEYs, or hashes of their DNSKEYs, to their parent zone. The parent zone publishes the hashes of the DNSKEYs in the form of a DS record. The DNSKEY RRset at the child may change over time. In order to keep the chain of trust intact, the DS records at the parent zone also needs to be updated. The rolling of the keys with the SEP bit on is one of the few tasks in DNSSEC that yet has to be fully automated.
The DNS UPDATE mechanism RFC 2136 (Vixie, P., Thomson, S., Rekhter, Y., and J. Bound, “Dynamic Updates in the Domain Name System (DNS UPDATE),” April 1997.) [RFC2136] can be used to push zone changes to the parent. To bootstrap the communication channel, information must be exchanged in order to detect service location and granting update privileges. A new or existing child zone is in need of a communication channel with the parent. This can be a direct channel or a channel through a third party:

> If the parent allows for direct communication channel with child zones, the parent can share the required data to set up the channel to the child zone. Once the child has the required credentials, it can use the direct communication channel with the parent to request zone changes related to its delegation.
>
> If a third party is involved, the third party acts on behalf of the parent. In this case, the third party will give out the required credentials to set up the communication channel.

Allowing for a third party in the communication channel ensures flexibility of the service location.
Please note that the document only describes the front end of the synchronization service. The first reason for that is that it is not necessary to write down how the DNS UPDATE is processed after it is accepted. It is merely a goal to provide a way for the child zone to automatically update the records at the zone cut. The second reason is that flexibility is needed in order to fit the protocol into the multifarious policies that exist among the great number of registrars. Thus, it is not required that the DNS UPDATE immediately updates the name server. Thus, it would still be possible to monitor the incoming updates with the tools of your choice. It is not a replacement of your RR provisioning system. The records in the DNS UPDATE can be converted into any kind of format.

---

2. Service Discovery

The service location is handed out during bootstrap. If this information is missing or incorrect, the normal guidelines for sending DNS UPDATE messages SHOULD be followed.

---

3. Access and Update Control

The DNS UPDATE normally is used for granting update permissions to a machine that is within the boundary of the same organization. This document proposes to grant child zones the same permissions. However, it MUST NOT be possible that a child zone updates information in the parent zone that falls outside the administrative domain of the corresponding delegation. For example, it MUST NOT be possible for a child zone to update the data that the parent is authoritative for, or update a delegation that is pointed to a different child zone. It MUST only be able to update records that match one of the following:

> **Or:** The owner name is equal the child zone name and RRtype is delegation specific. Currently those are records with RRtype NS or DS.
>
> **Or:** The owner name exists in the right side of the NS RRset belonging to the child zone and RRtype is is glue specific. Currently those are records with RRtype A or AAAA.

We can make a distinction here between narrow and wide glue records. Narrow glue records are said to be glue specific records with an owner name that is a subdomain of the child zone. Wide glue records are glue

specific records with an owner name that is outside of the delegated child domain.
These updates MAY be refused if it conflicts with the local policy.
This list may be expanded, if there is need for more delegation related zone content.
In case of adding or deleting delegation specific records, the DNSSEC related RRs in the parent zone might need to be updated.

---

4. Update Mechanism

4.1. Update Request

Updating the NS RRset or corresponding glue at the parent, an update can be sent at any time. Updating the DS RRset is part of key rollover, as described in RFC 4641 [RFC4641] (Kolkman, O. and R. Gieben, "DNSSEC Operational Practices," September 2006.). When performing a key rollover that involves updating the RRset at the parent, the child introduces a new DNSKEY in its zone that represents the security entry point for determining the chain of trust. After a while, it will revoke and/or remove the previous security entry point. The timings when to update the DS RRset at the parent are described in draft-dnsop-morris-dnssec-key-timing (Morris, S., Ihren, J., and J. Dickinson, "DNSSEC Key Timing Considerations," March 2010.) [keytiming]. When updating the DS RRset at the parent automatically, these timing specifications SHOULD be followed. To determine the propagation delays described in this document, the child should poll the parent zone for a short time, until the DS is visible at all parent name servers.
[Author's note] To discuss: A child zone might be unable to reach all parent name servers.
The child notifies the parent of the requested changes by sending a DNS UPDATE message. If it receives a NOERROR reply in return, the update is acknowledged by the parent zone. Otherwise, the child MAY retry transmitting the update. In order to prevent duplicate updates, it SHOULD follow the guidelines described in RFC 2136 [RFC2136] (Vixie, P., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)," April 1997.).

4.2. Processing the Update

An incoming DNS UPDATE message is processed as follows:

Step 1:

Check the TSIG/SIG0 credentials. In case of TSIG, the parent should follow the TSIG processing described in section 3.2 of RFC 2845. In case of SIG0, the parent should follow the SIG0 processing described in section 3.2 of RFC 2931.

Step 2: Verify that the updates matches the update policy for child zones.

Step 3: If verified, send back DNS UPDATE OK. Otherwise, send back DNS UPDATE REFUSED.

Step 4: If verified, apply changes. How that is done is a matter of policy.

---

5. Examples

5.1. Example BIND9 Configuration

This is how a parent zone can configure a policy to enable a child zone synchronize delegation specific records. The first rule of the update policy grants children to update their DS and NS records in the parent zone, in this case example.com. The second rule of the update policy grants children to update the corresponding glue records.

```
key cs.example.com. {
algorithm HMAC-MD5;
secret "secretforcs";
}
key math.example.com. {
algorithm HMAC-MD5;
secret "secretformath";
}
...
zone "example.com" {
type master;
file "example.com";
update-policy { grant *.example.com. self *.example.com. DS NS; };
update-policy { grant *.example.com. selfsub *.example.com. A AAAA; };
};
```

---

6. Security Considerations

Automating the synchronization of (DNSSEC) records between the parent and child creates a new communication channel. It is up to the policy of the parent, or the third party acting on behalf of the parent, who is allowed such privileges. A policy would usually include a form of access control. It is recommended that it involves transaction authentication, for example TSIG [RFC2845] (Vixie, P., Gudmundsson, O., Eastlake, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)," May 2000.) or SIG0 [RFC2931] (Eastlake, D., "DNS Request and Transaction Signatures (SIG(0)s)," September 2000.).
In the jungle of the DNS, many stakeholders exist. The registrant of a zone might not be the one that maintains the zone. That can possibly mean that many stakeholders need to possess the security credentials in order to use this synchronization channel. However, this problem exist with any kind of transaction authentication.
The disadvantage of adding a new communication channel is that you create a new attack window onto your DNS and DNSSEC records. When using this synchronization method for your DNSSEC records, a cryptographically equally strong, or stronger private key SHOULD be used, compared to the strength of your DNSSEC keys.
The advantage is that if somehow your DNSSEC keys are compromised, you can still use this channel to perform an emergency key rollover.

---

7. IANA Considerations

None.

---

8. Acknowledgments

Mark Andrews, Rickard Bellgrim, Wolfgang Nagele, Wouter Wijngaards.

---

9. Changelog

01:

> - Make it clear that the solution is flexible and it can fit into many and diverse environments of registrars.
>
> - Short section about service discovery.
>
> - Add text about narrow glue records.

- Add text about transaction authentication concerns with respect to many stakeholders involved.

00:

- Initial document

---

10. References

10.1. Informative References

[RFC2136]	Vixie, P., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)," RFC 2136, April 1997 (TXT, HTML, XML).
[RFC4641]	Kolkman, O. and R. Gieben, "DNSSEC Operational Practices," RFC 4641, September 2006 (TXT).
[keytiming]	Morris, S., Ihren, J., and J. Dickinson, "DNSSEC Key Timing Considerations," March 2010.

10.2. Normative References

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC2845]	Vixie, P., Gudmundsson, O., Eastlake, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)," RFC 2845, May 2000 (TXT).
[RFC2931]	Eastlake, D., "DNS Request and Transaction Signatures (SIG(0)s)," RFC 2931, September 2000 (TXT).
[RFC4035]	Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions," RFC 4035, March 2005 (TXT).

Author's Address

	Matthijs Mekking
	NLnet Labs
	Science Park 140
	Amsterdam 1098 XG

	The Netherlands
EMail:	matthijs@nlnetlabs.nl