DNSOP
Internet-Draft
Intended status: Informational
Expires: January 10, 2013

P. Wouters
Red Hat

July 9, 2012

Secure parent-child DNS update use cases
draft-wouters-dnsop-secure-update-use-cases-00.txt

Abstract

DNS zone administrators occasionally need to update data published by a parent zone, such as NS and DS records. Traditionally these updates have happened out-of-band: through DNS registrar interfaces, EPP, websites, or manually by operators. Some updates could also be done using DNS Dynamic Update [RFC2136].

The IETF's DNSOP working group is considering proposing additional mechanisms for such updates, possibly leveraging DNSSEC for authentication.

This document presents some use cases to drive this design work.

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 January 10, 2013.

Table of Contents

1. Introduction

Existing mechanisms for child-to-parent communication in DNS have some constraints that limit their utility. In particular, they require an authentication, which typically requires an extra credential to be exchanged between parent and child. With the advent of DNSSEC, it might be possible to use DNSSEC to authenticate these updates.

Furthermore, current mechanisms such as dynamic update also require that the child zone be able to reach the master server for the parent zone. In environments with hidden masters, offline DNSSEC signers or other network architecture constraints, this is not always be feasible.

This document identifies the main targets and use cases for automated updates and the concerns related to such automation.

[Note: While the document describes the use-cases with the zone, not the name server, as actor, this should not be taken to mean the signaling must be within the zone]

2. 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 RFC 2119 [RFC2119].

3. DNS records with use cases for automated updates

This document limits the scope of use cases to those DNS records that relate to the parent-child relationship itself. Policies for the TTL could be dictated by the parent or the child, depending on the relationship.

3.1. The DS RRset

The DS record needs to be updated when the child zone performs a Key Signing Key rollover. The parent name server cannot necessarily confirm the updated information by looking into the child zone, for example when the child zone has a spare, unpublished, DNSKEY record. Some parents want to receive DNSKEYs and create the DS record based on the received record. Other parents do not want to be responsible for creating any data for the child, and want to receive ready-made DS records, optionally restrained by the parent’s choices of valid algorithms.

3.2. The NS RRset

Both the child and the parent have a copy of the NS RRset. These RRsets are supposed to be identical. If they differ, it is referred as a "Lame Delegation". Keeping these sets synchronized would result in fewer lame delegations. Modifying the NS RRset is more complicated, as it could involve talking to name servers who do not yet know about the zone.

3.3. Glue records

Glue records are A or AAAA records that are needed to resolve an NS record that has a recursive relationship. For example, if the NS record for example.com points to ns.example.com, then a glue record is added to the parent zone (.com) for ns.example.com. Note that ns.example.com could be used in NS records for other zones as well.

4. Use cases

There are different kind of parent-child relationships. A very common relationship is the TLD registry using a Registry-Registrar-Registrant model. In this model, the child dictates the content to the parent. Another common parent-child relationship is the corporate relationship where the head office dictates some parent zone content to the child.

4.1. DNSSEC use cases in the Registant, Registrar, Registry model

4.1.1. Registrar has not adopted DNSSEC

Registrant running the child zone needs to convey their DS record to the Registry running the parent zone. Registrant can only communicate to the Registry using a Registrar. This Registrar does not support the EPP option to convey the DS record from Registrant to Registry. By sending an update via DNS to the Registry, Registrant bypasses the limitations of the Registrar. This use case would require some kind of boot-strap.

4.1.2. Registrar supports DNSSEC tediously

Registrar supports sending a DS record to the Registry via EPP. Registrant needs to use a human-oriented website interface of Registrar, which is very hard to automate and would break every time Registrar modifies their website for Registrants. By sending an update via DNS to the Registry, Registrant bypasses the limitations of the Registrar.

4.1.3. sub-Registrar supports DNSSEC but Registrar does not

Registrant can send DNSSEC updates to their (sub)Registrar, but the Registrar does not support receiving updates from sub-Registrar and sub-Registrar cannot communicate to Registry directly. The Registrant or sub-Registrar could bypass the limitations of the Registrar by sending DNSSEC updates directly to the Registry.

4.1.4. Registrant not setup to talk EPP to Registrar

Registrant is a lightweight entity using an off-the-shelve DNSSEC management solution. They have no technical expertise to communicate using EPP to the Registrar or Registry. Their DNS software could automate sending DNSSEC updates to the Registrar or Registry.

4.2. DNSSEC use cases with direct parent-child DNS server communication

4.2.1. DNS management solution of different vendors cannot communicate

Two different vendors have implemented non-standard, vendor-specific methods for non-DNS parent-child interaction. The DNS administrator(s) have different devices that cannot communicate with each other. If a generic DNS method was standardized, devices could implement this method and inter-operate with each other.

4.2.2. DNS management solution requires non-DNS traffic and new Authentication method

A non-DNS method for updating DS records between parent and child has been implemented. This method requires a lot of overhead to deploy. A new authentication method between parent and child is needed, for which there is no standard, causing potential interoperability issues. Firewall zones for DNS servers need to be updated to allow non-DNS traffic. If a generic DNS method was standardized, devices could implement this method and inter-operate with each other.

4.2.3. DNS Management GUI tools are lacking DNSSEC support

The DNS administrator is both administrating parent and child zone using one or more DNS management solutions. These solutions are running known up to date name server software but the vendor has not yet adopted DNSSEC in their GUI. A standardized solution not requiring additional GUI components could support updates more readily.

4.2.4. DNS management solution does not handle when being both child and parent

The DNS administrator uses a vendor product that does not automate adding the DS in the parent zone, despite the child DNSKEY being available to it. The DNS administrator needs to manually calculate the DS record and add it to the parent. They can no longer run automated rollovers due to this required action that can only be performed manually. If a generic DNS method was standardized, the device could send updates irrespective of whether it also manages the parent zone without additional effort.

4.3. Non-DNSSEC related DNS record updates

4.3.1. NS record and glue updates for the parent

Registrant has a difficult time keeping parent glue and NS RRsets up to date due to using a manual process. After establishing an authenticated relationship between parent and child using the DNSKEY/DS records, the parent could update its glue records based on the child zone content, either by regular polling, or by receiving a notification of the child to update. The parent could distribute such a notification to its siblings.

4.3.2. Parent changes its infrastructure

Parent name servers are pulling zones from different hidden primaries run by different departments with hundreds of zones. The parent name server infrastructure changes, and it wants to all its hidden primaries to use a different NS RRset. The parent sends an update to the hidden primaries to update the NS RRset for their zones. This category would also cover dyndns solutions where clients send individual host record updates to a parent that might change its location.

5. Relationships of zones and name servers

While the relationship between child zone and parent zone are well defined, in practice the chain of DNS servers involved is more complicated. Often the authoritative servers for the child zone do not communicate directly with the authoritative servers of the parent zone. Any methods for signaling between the child and parent zone should attempt to accommodate the listed infrastructure.

5.1. Hidden primary servers

Zones could be updated with IXFR/AXFR using hidden primary servers. DNSSEC signers often work this way. These primary name servers are usually restricted via dedicated VPN links or firewalls, and may not be able to determine or communicate with the required parent server for sending or receiving updates.

5.2. Offline private keys

Some DNSSEC signing solutions keep the private key inside an HSM or otherwise keep the private keys offline. Updates would need to be able to be generated offline, transported to an internet connected machine, and then transmitted to the parent zone.

5.3. Parent infrastructure

Some parent zones will require receiving updates for child zones directly from the child name servers, facilitating their current use of firewalls to restrict communication within the network. Other parent zones, such as TLDs, will want to leave their current name server structure unchanged and prefer updates for the child to a special name server dedicated to receive these updates.

5.4. Update capability indicator

Servers or zones that do not support or allow secure updates should not be sent repetitive update requests.

5.5. Legalities

Some deployments need to take legal restrictions into account. One such example is the Registry, Registrar, Registrant model, where the Registrant and Registry have no formal relationship with each other or are prohibited from communicating directly with each other. In such situations, secure automated updates should not be attempted.

6. The in-band update process

Depending on the appropriate process and relationship between parent and child zone, there could be different requirements for the update process.

6.1. No automatic updates

Records must be added or modified by the administrator of the zone using an out-of-band method.

6.2. Automatic child to parent updates for the DS record only

The child can send updates of its DS record to the parent, but cannot request updates to the NS RRset or glue records. The parent must be able to reject DS records that do not comply to its allowed selection of valid DNSKEY algorithms.

6.3. Fully-automatic child to parent updates

The child can send updates of all its records hosted at the parent, including DS records, NS records and glue records. The parent must be able to reject certain updates based on local policy

6.4. Automatic parent to child updates

The parent can send updates to the child for the NS records and glue records.

6.5. Fully-automatic child and parent synchronization

Parent and child automatically synchronize with no interaction on the part of the operators. This could be uni-directional of bi-directional.

6.6. Semi-automatic update

Parent and child synchronize, but only on the request of the parent or child administrator.

7. Applicability of automated updates to DNS infrastructure records

Automation and direct communication might not be appropriate in all scenarios. Implementations should take note of the considerations in this section.

7.1. Administrative Criteria

There are many situations where automated updates would not be allowed, or in practice could not be deployed in certain jurisdictions or corporate structures. Automatic update solutions should allow for disabling any such updates to support these restricted deployments.

7.1.1. Contractual obligations

Some DNS deployments have contractual restrictions that prevents certain parties from directly communicating with each other. For

example, some TLDs using the RRR model do not allow the Registry to talk to Registrants directly.

7.1.2. Company policy

Corporations often separate duties to different individuals or departments, sometimes across different jurisdictions . For example, a DNS officer in one country might not have the authorization of the company to update a DNS zone run by a subsidiary in other country. However, the reverse policy could also be true, where a DNS officer in one country running the parent zone must be able to update any child zone record of a subsidiary in another country.

7.1.3. Separation of roles

A Registrant (or "owner") of a zone might use a subcontractor to run the infrastructure of its zone. It might not be appropriate for the subcontractor to make any changes in the infrastructure of the zone, despite being in possession of required private keys to send changes to the parent. Similarly, a DNS administrator might be using a DNSSEC signing service, but would not want to allow this signing service to make any changes to the zone content other then signing the zone.

7.2. Content criteria

[Note: With NS records, are there any cases where the NS and glue records in the parent zone should not be identical to those in the child zone? What if the child name servers report different NS RRsets?]

When a DNS update is requested by the child zone, the parent zone could check and see if such an update would cause (significant?) harm to the child zone, and potentially refuse such an update.

7.2.1. DS update changing a secure zone to become insecure

If a DS record deletion request would cause the last DS record in the parent for that zone to be deleted, DNSSEC validation for the child zone would change from secure to insecure. A parent zone might wish to refuse such an update or require an additional confirmation.

7.2.2. DS update changing a zone to become bogus

The parent zone has two DS records for a child zone. Only one of these matches a DNSKEY record in the child zone. If a DS record deletion request would cause the valid DS record in the parent zone to be deleted, DNSSEC validation for the child zone would change from

secure to bogus. Similarly, if a child zone is is currently not signed, and the parent zone receives a DS record addition request, DNSSEC validation for the child zone would switch from insecure to bogus. A parent zone might wish to refuse such an update or require an additional confirmation.

7.2.3. DS update changing a zone to become secure

If a child zone becomes signed and automatically sends a DS addition request to the parent zone, the child zone would change from insecure to secure. This requires a sustained commitment by the child to maintain its DNSSEC status by regularly resigning its RRSIG records. The operators of the child zone might not be ready for such commitment, resulting in the zone becoming bogus at a later state. A parent zone might wish to refuse such an update or require an additional confirmation.

7.2.4. NS update causing an outage

If a child zone sends an NS update to the parent, the parent zone could check if the new NS records are properly configured to serve the child zone, guaranteeing that no service interruption would be caused by this update. A parent zone might wish to ignore such an update without an explicit override flag. This might be especially important to DNS operators that are unaware of these new DNS update mechanism and believe that changing zone content on the child would never cause any impacts to the parents.

8. Security Considerations

[Note: This currently overlaps with the section above]

An update of a DS record could change the authentication state of the parent-child relationship and should be handled with care. [Note: or require out-of-band signaling?]

9. IANA Considerations

This Internet Draft includes no request to IANA.

10. Acknowledgements

[Note: none yet]

11. References

11.1. Normative References

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

11.2. 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.

[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.

Author’s Address

Paul Wouters
Red Hat

Email: pwouters@redhat.com