Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 22, 2016

J. Latour
CIRA
O. Gudmundsson
Cloudflare, Inc.
P. Wouters
Red Hat
M. Pounsett
Rightside Group, Ltd.

March 21, 2016

Third Party DNS operator to Registrars/Registries Protocol
draft-latour-dnsoperator-to-rrr-protocol-03.txt

Abstract

There are several problems that arise in the standard Registrant/Registrar/Registry model when the operator of a zone is neither the Registrant nor the Registrar for the delegation. Historically the issues have been minor, and limited to difficulty guiding the Registrant through the initial changes to the NS records for the delegation. As this is usually a one time activity when the operator first takes charge of the zone it has not been treated as a serious issue.

When the domain on the other hand uses DNSSEC it necessary for the Registrant in this situation to make regular (sometimes annual) changes to the delegation in order to track KSK rollover, by updating the delegation's DS record(s). Under the current model this is prone to Registrant error and significant delays. Even when the Registrant has outsourced the operation of DNS to a third party the registrant still has to be in the loop to update the DS record.

There is a need for a simple protocol that allows a third party DNS operator to update DS and NS records in a trusted manner for a delegation without involving the registrant for each operation.

The protocol described in this draft is REST based, and when used through an authenticated channel can be used to establish the DNSSEC Initial Trust (to turn on DNSSEC or bootstrap DNSSEC). Once DNSSEC trust is established this channel can be used to trigger maintenance of delegation records such as DS, NS, and glue records. The protocol is kept as simple as possible.

Status of This Memo

This Internet-Draft will expire on September 22, 2016.

Copyright Notice

Table of Contents

1. Introduction

Why is this needed? DNS registration systems today are designed around making registrations easy and fast. After the domain has been registered the there are really three options on who maintains the DNS zone that is loaded on the "primary" DNS servers for the domain this can be the Registrant, Registrar, or a third party DNS Operator.

Unfortunately the ease to make changes differs for each one of these options. The Registrant needs to use the interface that the registrar provides to update NS and DS records. The Registrar on the other hand can make changes directly into the registration system. The third party DNS Operator on the hand needs to go through the Registrant to update any delegation information.

Current system does not work well, there are many examples of failures including the inability to upload DS records due to non-support by Registrar interface, the registrant forgets/does-not perform action but tools proceed with key roll-over without checking that the new DS is in place. Another common failure is the DS record is not removed when the DNS Operator changes from one that supports DNSSEC signing to one that does not.

The failures result either inability to use DNSSEC or in validation failures that case the domain to become invalid and all users that are behind validating resolvers will not be able to to access the domain.

2. Notional Conventions

2.1. Definitions

For the purposes of this draft, a third-party DNS Operator is any DNS Operator responsible for a zone where the operator is neither the Registrant nor the Registrar of record for the delegation.

Uses of the word 'Registrar' in this document may also be applied to resellers: an entity that sells delegations through a registrar with whom the entity has a reseller agreement.

2.2. RFC2119 Keywords

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].

3. What is the goal?

The primary goal is to use the DNS protocol to provide information from child zone to the parent zone, to maintain the delegation information. The precondition for this to be practical is that the domain is DNSSEC signed.

In the general case there should be a way to find the right Registrar/Registry entity to talk to but that does not exist. Whois[] is the natural protocol to carry such information but that protocol is unreliable and hard to parse. Its proposed successor RDAP [RFC7480] has yet be deployed on most TLD's.

The preferred communication mechanism is to use is to use a REST [RFC6690] call to start processing of the requested delegation information.

3.1. Why DNSSEC?

DNSSEC [RFC4035] provides data authentication for DNS answers, having DNSSEC enabled makes it possible to trust the answers. The biggest stumbling block is deploying DNSSEC is the initial configuration of the DNSSEC domain trust anchor in the parent, DS record.

3.2. How does a child signal its parent it wants DNSSEC Trust Anchor? The child

needs first to sign the domain, then the child can "upload" the DS record to its parent. The "normal" way to upload is to go through registration interface, but that fails frequently. The DNS Operator

may not have access to the interface thus the registrant needs to relay the information. For large operations this does not scale, as evident in lack of Trust Anchors for signed deployments that are operated by third parties.

The child can signal its desire to have DNSSEC validation enabled by publishing one of the special DNS records CDS and/or CDNSKEY[RFC7344] and its proposed extension [I-D.ietf-dnsop-maintain-ds].

Once the "parent" "sees" these records it SHOULD start acceptance processing. This document will cover below how to make the CDS records visible to the right parental agent.

We and [I-D.ogud-dnsop-maintain-ds] argue that the publication of CDS/CDNSKEY record is sufficient for the parent to start the acceptance processing. The main point is to provide authentication thus if the child is in "good" state then the DS upload should be simple to accept and publish. If there is a problem the parent has ability to not add the DS.

3.3. What checks are needed by parent?

The parent upon receiving a signal that it check the child for desire for DS record publication. The basic tests include,

1. The zone is signed
2. The zone has a CDS signed by a KSK referenced in the current DS, referring to a at least one key in the current DNSKEY RRset
3. All the name-servers for the zone agree on the CDS RRset contents

Parents can have additional tests, defined delays, queries over TCP, and even ask the DNS Operator to prove they can add data to the zone, or provide a code that is tied to the affected zone. The protocol is partially-synchronous, i.e. the server can elect to hold connection open until the operation has concluded or it can return that it received the request. It is up to the child to monitor the parent for completion of the operation and issue possible follow-up calls.

4. OP-3-DNS-RR RESTful API

The specification of this API is minimalist, but a realistic one. Question: How to respond if the party contacted is not ALLOWED to make the requested change?

4.1. Authentication

The API does not impose any unique server authentication requirements. The server authentication provided by TLS fully addresses the needs. In general, for the API SHOULD be provided over TLS-protected transport (e.g., HTTPS) or VPN.

4.2. Authorization

Authorization is out of scope of this document. The CDS records present in the zone file are indications of intention to sign/unsign/update the DS records of the domain in the parent zone. This means the proceeding of the action is not determined by who issued the request. Therefore, authorization is out of the scope. Registries and registrars who plan to provide this service can, however, implement their own policy such as IP white listing, API key, etc.

4.3. Base URL Locator

The base URL for registries or registrars who want to provide this service to DNS Operators can be made auto-discoverable as an RDAP extension.

4.4. CDS resource

Path: /domains/{domain}/cds {domain}: is the domain name to be operated on

4.4.1. Initial Trust Establishment (Enable DNSSEC validation)

4.4.1.1. Request

Syntax: POST /domains/{domain}/cds

A DS record based on the CDS record in the child zone file will be inserted into the registry and the parent zone file upon the successful completion of such request. If there are multiple CDS records in the CDS RRset, multiple DS records will be added.

Either the CDS/CDNSKEY or the DNSKEY can be used to create the DS record. Note: entity expecting CDNSKEY is still expected accept the /cds command.

4.4.1.2. Response

- HTTP Status code 201 indicates a success.

- HTTP Status code 400 indicates a failure due to validation.

- HTTP Status code 403 indicates a failure due to an invalid challenge token.

- HTTP Status code 404 indicates the domain does not exist.

- HTTP Status code 500 indicates a failure due to unforeseeable reasons.

4.4.2. Removing a DS (turn off DNSSEC)

4.4.2.1. Request

Syntax: DELETE /domains/{domain}/cds

4.4.2.2. Response

- HTTP Status code 200 indicates a success.

- HTTP Status code 400 indicates a failure due to validation.

- HTTP Status code 404 indicates the domain does not exist.

- HTTP Status code 500 indicates a failure due to unforeseeable reasons.

4.4.3. DS Maintenance (Key roll over)

4.4.3.1. Request

Syntax: PUT /domains/{domain}/cds

4.4.3.2. Response

- HTTP Status code 200 indicates a success.

- HTTP Status code 400 indicates a failure due to validation.

- HTTP Status code 404 indicates the domain does not exist.

- HTTP Status code 500 indicates a failure due to unforeseeable reasons.

4.5. Tokens resource

Path: /domains/{domain}/tokens {domain}: is the domain name to be operated on

4.5.1. Setup Initial Trust Establishment with Challenge

4.5.1.1. Request

Syntax: POST /domains/{domain}/tokens

A random token to be included as a _delegate TXT record prior establishing the DNSSEC initial trust.

4.5.1.2. Response

- HTTP Status code 200 indicates a success. Token included in the body of the response, as a valid TXT record

- HTTP Status code 404 indicates the domain does not exist.

- HTTP Status code 500 indicates a failure due to unforeseeable reasons.

4.6. Customized Error Messages

Service providers can provide a customized error message in the response body in addition to the HTTP status code defined in the previous section.

This can include an Identifiying number/string that can be used to track the requests.

#Using the definitions This section at the moment contains comments from early implementers

4.7. How to react to 403 on POST cds

The basic reaction to a 403 on POST /domains/{domain}/cds is to issue POST /domains/{domain}/tokens command to fetch the challenge to insert into the zone.

5. Security considerations

TBD This will hopefully get more zones to become validated thus overall the security gain out weights the possible drawbacks.

risk of takeover ? risk of validation errors < declines transfer issues

6. IANA Actions

URI ??? TBD

7. Internationalization Considerations

This protocol is designed for machine to machine communications

8. References

8.1. Normative References

[I-D.ietf-dnsop-maintain-ds]
Gu[eth]mundsson, O. and P. Wouters, "Managing DS records from parent via CDS/CDNSKEY", draft-ietf-dnsop-maintain-ds-00 (work in progress), December 2015.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005, <http://www.rfc-editor.org/info/rfc4035>.

[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>.

8.2. Informative References

[I-D.ogud-dnsop-maintain-ds]
Gu[eth]mundsson, O. and P. Wouters, "Managing DS records from parent via CDS/CDNSKEY", draft-ogud-dnsop-maintain-ds-00 (work in progress), October 2015.

[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>.

[RFC6690] Shelby, Z., "Constrained RESTful Environments (CoRE) Link Format", RFC 6690, DOI 10.17487/RFC6690, August 2012, <http://www.rfc-editor.org/info/rfc6690>.

[RFC7480] Newton, A., Ellacott, B., and N. Kong, "HTTP Usage in the Registration Data Access Protocol (RDAP)", RFC 7480, DOI 10.17487/RFC7480, March 2015, <http://www.rfc-editor.org/info/rfc7480>.

Appendix A. Document History

A.1. Version 03

Clarified based on comments and questions from early implementors

A.2. Version 02

Reflected comments on mailing lists

A.3. Version 01

This version adds a full REST definition this is based on suggestions from Jakob Schlyter.

A.4. Version 00

First rough version

Authors' Addresses

Jacques Latour
CIRA

Email: jacques.latour@cira.ca

Olafur Gudmundsson
Cloudflare, Inc.

Email: olafur+ietf@cloudflare.com

Paul Wouters
Red Hat

Email: paul@nohats.ca

Matthew Pounsett
Rightside Group, Ltd.

Email: matt@conundrum.com