regext
Internet-Draft
Intended status: Standards Track
Expires: March 16, 2018

J. Latour
CIRA
O. Gudmundsson
Cloudflare, Inc.
P. Wouters
Red Hat
M. Pounsett
Rightside Group, Ltd.
September 12, 2017

Third Party DNS operator to Registrars/Registries Protocol draft-ietf-regext-dnsoperator-to-rrr-protocol-04

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 uses DNSSEC it necessary to make regular (sometimes annual) changes to the delegation, updating DS record(s) in order to track KSK rollover. Under the current model this is prone to delays and errors, as the Registrant must participate in updates to DS records.

This document describes a simple protocol that allows a third party DNS operator to: establish the initial chain of trust (bootstrap DNSSEC) for a delegation; update DS records for a delegation; and, remove DS records from a secure delegation. The DNS operator may do these things in a trusted manner, without involving the Registrant for each operation. This same protocol can be used by Registrants to maintain their own domains if they wish.

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 https://datatracker.ietf.org/drafts/current/.

Copyright Notice

Table of Contents

1. Introduction

After a domain has been registered, one of three parties will maintain the DNS zone loaded on the "primary" DNS servers: the Registrant, the Registrar, or a third party DNS operator. DNS registration systems were originally designed around making registrations easy and fast, however after registration the complexity of making changes to the delegation differs for each of these parties. The Registrar can make changes directly in the Registry systems through some API (typically EPP [RFC5730]). The Registrant is typically limited to using a web interface supplied by the Registrar or Reseller. Typically, a third party DNS Operator must to go through the Registrant to update any delegation information.

Unless the responsible Registration Entity is scanning child zones for CDS records in order to bootstrap or update DNSSEC, the operator must contact and engage the Registrant in updating DS records for the delegation. New information must be communicated to the Registrant, who must submit that information to the Registrar. Typically this involves cutting and pasting between email and a web interface, which is error prone. Furthermore, involving Registrants in this way does not scale for even moderately sized DNS operators. Tracking thousands (or millions) of changes sent to customers, and following up if those changes are not submitted to the Registrar, or are submitted with errors, is itself expensive and error prone.

The current system does not work well, as there are many types of failures that have been reported at all levels in the registration model. The failures result in either the inability to use DNSSEC or in validation failures that cause the domain to become unavailable to users behind validating resolvers.

The goal of this document is to create a protocol for establishing a secure chain of trust that involves parties not in the traditional Registrant/Registrar/Registry (RRR) model, and to reduce the friction in maintaining DNSSEC secured delegations in these cases. It describes a REST-based [RFC6690] protocol which can be used to

establish DNSSEC initial trust (to enable or bootstrap DNSSEC), and to trigger maintenance of DS records.

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 "child" and "parent" refer to the relationship between DNS zone operators (see [RFC7719] and [I-D.ietf-dnsop-terminology-bis]). In this document, unless otherwise noted, the child is the third-party DNS operator and the parent is the Registry.

Use of the term "Registration Entity" in this document may refer to any party that engages directly in registration activities with the Registrant. Typically this will be a Reseller or Registrar, but in some cases, such as when a Registry directly sells registrations to the public, may apply to the Registry. Even in cases where a Registrar is involved, this term may still apply to a Registry if that Registry normally accepts DS/DNSKEY updates directly from Registrants.

The CDS and CDNSKEY DNS resource records, having substantially the same function but for different record types, are used interchangably in this document. Unless otherwise noted, any use of "CDS" or "CDNSKEY" can be assumed to also refer to the other.

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. Process Overview

3.1. Identifying the Registration Entity

As of publication of this document, there has never been a standardized or widely deployed method for easily and scalably identifying the Registration Entity for a particular registration.

At this time, WHOIS [RFC3912] is the only widely deployed protocol to carry such information, but WHOIS responses are unstructured text, and each implementor can lay out its text responses differently. In addition, Registries may include referrals in this unstructured text

to the WHOIS interfaces of their Registrars, and those Registrar WHOIS interface in turn have their own layouts. This presents a text parsing problem which is infeasible to solve.

RDAP, the successor to WHOIS, described in [RFC7480], solves the problems of unstructured responses, and a consistently implemented referral system, however at this time RDAP has yet to be deployed at most Registries.

With no current mechanism in place to scalably discover the Registrar for a particular registration, the problem of automatic discovery of the base URL of the API is considered out of scope of this document. The authors recommend standardization of an RDAP extension to obtain this information from the Registry.

3.2. Establishing a Chain of Trust

After signing the zone, the child DNS Operator needs to upload the DS record(s) to the parent. The child can signal its desire to have DNSSEC validation enabled by publishing one of the special DNS records CDS and/or CDNSKEY as defined in [RFC7344] and [RFC8078].

Registration Entities MAY regularly scan the child name servers of unsecured delegations for CDS records in order to bootstrap DNSSEC, and are advised to do so. At the time of publication, some ccTLD Registries are already doing this. A Registration Entity that regularly scans all child zones under its responsibility (both secured and unsecured) for CDS will not require the API described in this document. However, such a Registration Entity should follow the guidelines discussed in Section 3.5 below when using CDS to bootstrap DNSSEC on a previously unsecured delegation.

In the case where the Registration Entity is not normally scanning child zones for CDS records, the Registration Entity SHOULD implement the API from this document, allowing child operators to notify the Registration Entity to begin such a scan.

Once the Registration Entity finds CDS records in a child zone it is responsible for, or receives a signal via this API, it SHOULD start acceptance processing as described below.

3.3. Maintaining the Chain of Trust

Once the secure chain of trust is established, the Registration Entity SHOULD regularly scan the child zone for CDS record changes. If the Registration Entity implements the protocol described in this document, then it SHOULD also accept signals via this protocol to immediately check the child zone for CDS records.

Server implementations of this protocol MAY include rate limiting to protect their systems and the systems of child operators from abuse.

Each parent operator and Registration Entity is responsible for developing, implementing, and communicating their DNSSEC maintenance policies.

3.4. Acceptance Processing

The Registration Entity, upon receiving a signal or detecting through polling that the child desires to have its delegation updated, SHOULD run a series of tests to ensure that updating the parent zone will not create or exacerbate any problems with the child zone. The basic tests SHOULD include:

- checks that the child zone is is properly signed as per the Registration Entity and parent DNSSEC policies
- if updating the DS record, a check to ensure the child CDS RRset references a KSK which is present in the child DNSKEY RRset and signs the CDS RRset
- ensuring all name servers in the apex NS RRset of the child zone agree on the apex NS RRset and CDS RRset contents

The Registration Entity SHOULD NOT make any changes to the DS RRset if the child name servers do not agree on the CDS content.

3.5. Bootstrapping DNSSEC

Registration Entities SHOULD require compliance with additional tests in the case of establishing a new chain of trust.

- The Registration Entity SHOULD check that all child name servers respond with a consistent CDS RRset for a number of queries over an extended period of time. Any change in DS response or inconsistency between child responses in that time might indicate an attempted Man in the Middle (MITM) attack, and SHOULD reset the test. This minimizes the possibility of an attacker spoofing responses. An example of such a policy might be to scan all child name servers in the delegation NS RRset every two hours for a week.
- The Registration Entity SHOULD require all of the child name servers in the delegation NS RRset to send the same response to a CDS query whether sent over TCP or UDP.

- The Registration Entity MAY require the child zone to implement zone delegation best practices as described in [I-D.wallstrom-dnsop-dns-delegation-requirements].

- The Registration Entity MAY require the child operator to prove they can add data to the zone, for example by publishing a particular token. See Section 4.2.2 below.

4. API Definition

This protocol is partially synchronous, meaning the server can elect to hold connections open until operations have completed, or it can return a status code indicating that it has received a request, and close the connection. It is up to the child to monitor the parent for completion of the operation, and issue possible follow-up calls to the Registration Entity.

Clients may be denied access to change the DS records for domains that are Registry Locked (HTTP Status code 401). Registry Lock is a mechanism provided by certain Registries or Registrars that prevents domain hijacking by ensuring no attributes of the domain are changeable, and no transfer or deletion transactions can be processed against the domain name without manual intervention.

4.1. Authentication

The API does not impose any unique server authentication requirements. The server authentication provided by TLS fully addresses the needs of this protocol. The API MUST be provided over TLS-protected transport (e.g., HTTPS) or VPN.

Client authentication is considered out of scope of this document. The publication of CDS records in the child zone is an indication that the child operator intends to perform DS-record-updating activities (add/delete) in the parent zone. Since this protocol is simply a signal to the Registration Entity that they should examine the child zone for such intentions, additional authentication of the client making the request is considered unnecessary.

Registration Entities MAY implement their own policy to protect access to the API, such as with IP white listing, client TLS certificates, etc.. Registration Entities SHOULD take steps to ensure that a lack of additional authentication does not open up a denial of service mechanism against the systems of the Registration Entity, the Registry, or the child operator.

4.2. RESTful Resources

In the following text, "{domain}" is the child zone to be operated on.

4.2.1. CDS resource

Path: /domains/{domain}/cds

4.2.1.1. Establishing Initial Trust (Enabling DNSSEC)

4.2.1.1.1. Request

Syntax: POST /domains/{domain}/cds

Request that an initial set of DS records based on the CDS record in the child zone be inserted into the Registry and the parent zone upon the successful completion of the request. If there are multiple CDS records in the CDS RRset, multiple DS records will be added.

The body of the POST SHOULD be empty, however server implementations SHOULD NOT reject nonempty requests.

4.2.1.1.2. Response

- HTTP Status code 201 indicates a success.
- HTTP Status code 400 indicates a failure due to validation.
- HTTP Status code 401 indicates an unauthorized resource access.
- 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 409 indicates the delegation already has a DS RRset.
- HTTP Status code 429 indicates the client has been rate-limited.
- HTTP Status code 500 indicates a failure due to unforeseeable reasons.

This request is for setting up initial trust in the delegation. The Registration Entity SHOULD return a status code 409 if it already has a DS RRset for the child zone.

Upon receipt of a 403 response the child operator SHOULD issue a POST for the "token" resource to fetch a challenge token to insert into the zone.

4.2.1.2. Removing DS Records

4.2.1.2.1. Request

Syntax: DELETE /domains/{domain}/cds

Request that the Registration Entity check for a null CDS or CDNSKEY record in the child zone, indicating a request that the entire DS RRset be removed. This will make the delegation insecure.

4.2.1.2.2. Response

- HTTP Status code 200 indicates a success.
- HTTP Status code 400 indicates a failure due to validation.
- HTTP Status code 401 indicates an unauthorized resource access.
- HTTP Status code 404 indicates the domain does not exist.
- HTTP Status code 412 indicates the parent does not have a DS RRset
- HTTP Status code 429 indicates the client has been rate-limited.
- HTTP Status code 500 indicates a failure due to unforeseeable reasons.

4.2.1.3. Modifying DS Records

4.2.1.3.1. Request

Syntax: PUT /domains/{domain}/cds

Request that the Registration Entity modify the DS RRset based on the CDS/CDNSKEY available in the child zone. As a result of this request the Registration Entity SHOULD add or delete DS or DNSKEY records as indicated by the CDS/CDNSKEY RRset, but MUST NOT delete the entire DS RRset.

4.2.1.3.2. Response

- HTTP Status code 200 indicates a success.
- HTTP Status code 400 indicates a failure due to validation.

- HTTP Status code 401 indicates an unauthorized resource access.
- HTTP Status code 404 indicates the domain does not exist.
- HTTP Status code 412 indicates the parent does not have a DS RRset
- HTTP Status code 429 indicates the client has been rate-limited.
- HTTP Status code 500 indicates a failure due to unforeseeable reasons.

4.2.2. Token resource

Path: /domains/{domain}/token

4.2.2.1. Establish Initial Trust with Challenge

4.2.2.1.1. Request

Syntax: GET /domains/{domain}/token

The DNSSEC policy of the Registration Entity may require proof that the DNS Operator is in control of the domain. The token API call returns a random token to be included as a TXT record for the _delegate.@ domain name (where @ is the apex of the child zone) prior establishing the DNSSEC initial trust. This is an additional trust control mechanism to establish the initial chain of trust.

Once the child operator has received a token, it SHOULD be inserted in the zone and the operator SHOULD proceed with a POST of the cds resource.

The Registration Entity MAY expire the token after a reasonable period. The Registration Entity SHOULD document an explanation of whether and when tokens are expired in their DNSSEC policy.

Note that the _delegate TXT record is publicly available and not a secret token.

4.2.2.1.2. Response

- HTTP Status code 200 indicates a success. A token is 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.

4.3. Customized Error Messages

Registration Entities MAY provide a customized error message in the response body in addition to the HTTP status code defined in the previous section. This response MAY include an identifying number/ string that can be used to track the request.

5. Security considerations

When zones are properly provisioned, and delegations follow standards and best practices (e.g. [I-D.wallstrom-dnsop-dns-delegation-requirements]), the Registration Entity or Registry can trust the DNS information it receives from multiple child name servers, over time, and/or over TCP to establish the initial chain of trust.

In addition, the Registration Entity or Registry can require the DNS Operator to prove they control the zone by requiring the child operator to navigate additional hurdles, such as adding a challenge token to the zone.

This protocol should increase the adoption of DNSSEC, enabling more zones to become validated thus overall the security gain outweighs the possible drawbacks.

Registrants and DNS Operators always have the option to establish the chain of trust in band via the standard Registrant/Registrar/Registry model.

6. IANA Actions

This document has no actions for IANA

7. Internationalization Considerations

This protocol is designed for machine to machine communications. Clients and servers SHOULD use punycode [RFC3492] when operating on internationalized domain names.

8. References

8.1. Normative References

[RFC3492] Costello, A., "Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)", RFC 3492, DOI 10.17487/RFC3492, March 2003, <https://www.rfc-editor.org/info/rfc3492>.

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

[RFC7344] Kumari, W., Gudmundsson, O., and G. Barwood, "Automating DNSSEC Delegation Trust Maintenance", RFC 7344, DOI 10.17487/RFC7344, September 2014, <https://www.rfc-editor.org/info/rfc7344>.

[RFC8078] Gudmundsson, O. and P. Wouters, "Managing DS Records from the Parent via CDS/CDNSKEY", RFC 8078, DOI 10.17487/RFC8078, March 2017, <https://www.rfc-editor.org/info/rfc8078>.

8.2. Informative References

[I-D.ietf-dnsop-terminology-bis] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", draft-ietf-dnsop-terminology-bis-06 (work in progress), July 2017.

[I-D.wallstrom-dnsop-dns-delegation-requirements] Wallstrom, P. and J. Schlyter, "DNS Delegation Requirements", draft-wallstrom-dnsop-dns-delegation-requirements-03 (work in progress), October 2016.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3912] Daigle, L., "WHOIS Protocol Specification", RFC 3912, DOI 10.17487/RFC3912, September 2004, <https://www.rfc-editor.org/info/rfc3912>.

[RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, DOI 10.17487/RFC5730, August 2009, <https://www.rfc-editor.org/info/rfc5730>.

[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, <https://www.rfc-editor.org/info/rfc7480>.

[RFC7719] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", RFC 7719, DOI 10.17487/RFC7719, December 2015, <https://www.rfc-editor.org/info/rfc7719>.

Appendix A. Document History

A.1. regext Version 04

- changed uses of Registrar to Registration Entity and updated definitions to improve clarity
- adding note about CDS/CDNSKEY interchangability in this document
- added advice to scan all delegations (including insecure delegations) for CDS in order to bootstrap or update DNSSEC
- removed "Other Delegation Maintenance" section, since we decided a while ago not to use this to update NS

A.2. regext Version 03

- simplify abstract
- move all justification text to Intro
- added HTTP response codes for rate limiting (429), missing DS RRsets (412)
- expanded on Internationalization Considerations
- corrected informative/normative document references
- clarify parent/Registrar references in the draft
- general spelling/grammar/style cleanup
- removed references to NS and glue maintenance
- clarify content of POST body for 'cds' resource
- change verb for obtaining a 'token' to GET
- Updated reference to RFC8078

A.3. regext Version 02

- Clarified based on comments and questions from early implementors (JL)
- Text edits and clarifications.

A.4. regext Version 01

- Rewrote Abstract and Into (MP)
- Introduced code 401 when changes are not allowed
- Text edits and clarifications.

A.5. regext Version 00

- Working group document same as 03, just track changed to standard

A.6. Version 03

- Clarified based on comments and questions from early implementors

A.7. Version 02

- Reflected comments on mailing lists

A.8. Version 01

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

A.9. Version 00

- First rough version

Authors' Addresses

Jacques Latour
CIRA
Ottawa, ON

Email: jacques.latour@cira.ca

Olafur Gudmundsson
Cloudflare, Inc.
San Francisco, CA

Email: olafur+ietf@cloudflare.com

Paul Wouters
Red Hat
Toronto, ON

Email: paul@nohats.ca

Matthew Pounsett
Rightside Group, Ltd.
Toronto, ON

Email: matt@conundrum.com