Child To Parent Synchronization in DNS
draft-ietf-dnsop-child-syncronization-04

Abstract

   This document specifies how a child zone in the DNS can publish a
   record to indicate to a parental agent that the parental agent may
   copy and process certain records from the child zone. The existence
   of the record and any change in its value can be monitored by a
   parental agent and acted on depending on local policy.

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 May 30, 2015.

Copyright Notice

Table of Contents

1. Introduction

   This document specifies how a child zone in the DNS ([RFC1034],
   [RFC1035]) can publish a record to indicate to a parental agent that
   it can copy and process certain records from the child zone. The
   existence of the record and any change in its value can be monitored
   by a parental agent and acted on depending on local policy.

   Currently today, some resource records (RRs) in a parent zone are
   typically expected to be in sync with the source data in the child's
   zone. The most common records that should match are the nameserver
   (NS) records and any necessary associated address records (A and
   AAAA), also known as "glue records". These records are referred to
   as "delegation records".

   It has been challenging for operators of child DNS zones to update
   their delegation records within the parent's set in a timely fashion.
   These difficulties may stem from operator laziness, as well as from
   the complexities of maintaining a large number of DNS zones. Having
   an automated mechanism for signaling updates will greatly ease the
   child zone operator's maintenance burden and improve the robustness
   of the DNS as a whole.

   This draft introduces a new Resource Record Type (RRType) named
   "CSYNC" that indicates which delegation records published by a child
   DNS operator should be processed by a parental agent and used to
   update the parent zone's DNS data.

   This specification was not designed to synchronize DNSSEC security
   records, such as DS RRsets. For a solution to this problem, see the
   complementary solution [RFC7344], which is designed to maintain
   security delegation information. In addition, this specification
   does not address how to perform bootstrapping operations, including
   to get the required initial DNSSEC-secured operating environment in
   place.

1.1. Terminology Used in This Document

   The terminology used in this document is defined in this section.

   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].

   Terminology describing relationships between the interacting roles
   involved in this document are defined in the following list:

   Child: The entity on record that has the delegation of the domain
      from the parent

   Parent: The domain in which the child is registered

   Child DNS operator: The entity that maintains and publishes the zone
      information for the child DNS

   Parental agent: The entity that the child has relationship with, to
      change its delegation information

2. Definition of the CSYNC RRType

   The CSYNC RRType contains, in its RDATA component, these parts: an
   SOA serial number, a set of flags and a simple bit-list indicating
   the DNS RRTypes in the child that should be processed by the parental
   agent in order to modify the DNS delegation records within the
   parent's zone for the child DNS operator. Child DNS operators
   wanting a parental agent to perform the synchronization steps
   outlined in this document MUST publish a CSYNC record at the apex of
   the child zone. Parental agent implementations MAY choose to query
   child zones for this record and process DNS record data as indicated
   by the Type Bit Map field in the RDATA of the CSYNC record. How the
   data is processed is described in Section 3.

   Parental agents MUST process the entire set of child data indicated
   by the Type Bit Map field (i.e., all record types indicated along
   with all of the necessary records to support processing of that type)
   or else parental agents MUST NOT make any changes to parental records
   at all. Errors due to unsupported Type Bit Map bits, or otherwise
   nonpunishable data, SHALL result in no change to the parent zone's
   delegation information for the Child. Parental agents MUST ignore a
   Child's CSYNC RDATA set if multiple CSYNC resource records are found;
   only a single CSYNC record should ever be present.

   The parental agent MUST perform DNSSEC validation ([RFC4033],
   [RFC4034], [RFC4035]), of the CSYNC RRType data and MUST perform
   DNSSEC validation of any data to be copied from the Child to the
   Parent. Parents MUST NOT process any data from any of these records
   if any of the validation results indicate any anything other than
   "Secure" [RFC4034] or if any the required data can not be
   successfully retrieved.

2.1. The CSYNC Resource Record Format

2.1.1. **The CSYNC Resource Record Wire Format**

   The CSYNC RDATA consists of the following fields:

```
                        1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          SOA Serial                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |       Flags           |            Type Bit Map        /
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   /                     Type Bit Map (continued)                 /
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
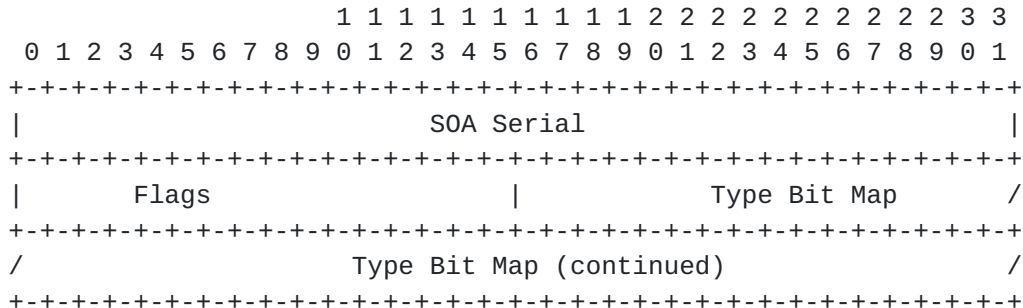
2.1.1.1. **The SOA Serial Field**

   The SOA Serial field contains a copy of the 32-bit SOA serial number
   from the child zone. If the soaminimum flag is set, parental agents
   querying children's authoritative servers MUST NOT act on data from
   zones advertising an SOA serial number less than this value. See
   [RFC1982] for properly implementing "less than" logic. If the
   soaminimum flag is not set, parental agents MUST ignore the value in
   the SOA Serial Field. Clients can set the field to any value if the
   soaminimum flag is unset, such as the number zero.

   Note that a child zone's current SOA serial number may be greater
   than the number indicated by the CSYNC record. A child SHOULD update
   the SOA Serial field in the CSYNC record every time the data being
   referenced by the CSYNC record is changed (e.g. an NS record or
   associated address record is changed). A child MAY choose to update
   the SOA Serial field to always match the current SOA serial field.

   Parental agents MAY cache SOA serial numbers from data they use and
   refuse to process data from zones older than the last instance they
   pulled data from.

   Although Section 3.2 of [RFC1982] describes how to properly implement
   a less-than comparison operation with SOA serial numbers that may
   wrap beyond the 32-bit value in both the SOA record and the CSYNC
   record, it is important than a child using the soaminimum flag must
   not increment it's SOA serial number value more than 2^{16} within the
   period of time that a parent might wait between polling the child for
   the CSYNC record.

2.1.1.2. **The Flags Field**

   The Flags field contains 16 bits of boolean flags that define
   operations which affect the processing of the CSYNC record. The

flags defined in this document are as follows:

   0x00 0x01: "immediate"

   0x00 0x02: "soaminimum"

The definitions for how the flags are to be used can be found later
in Section [Section 3](#).

The remaining flags are reserved for use by future specifications.
Undefined flags MUST be set to 0 by CSYNC publishers. Parental
agents MUST NOT process a CSYNC record if it contains a 1 value for a
flag that is unknown to or unsupported by the parental agent.

[2.1.1.2.1](#). The Type Bit Map Field

The Type Bit Map field indicates the record types to be processed by
the parental agent, according to the procedures in Section [Section 3](#).
The Type Bit Map field is encoded in the same way as the Type Bit
Maps field of the NSEC record, described in [[RFC4034], Section 4.1.2](#).
If a bit has been set that a parental agent implementation does not
understand, the parental agent MUST NOT act upon the record.
Specifically: a parental agent must not just copy the data and must
understand the semantics associated with an bit in the Type Bit Map
field that has been set to 1.

[2.1.2](#). The CSYNC Presentation Format

The CSYNC presentation format is as follows:

   The SOA Serial field is represented as an integer.

   The Flags field is represented as an integer.

   The Type Bit Map field is represented as a sequence of RR type
   mnemonics. When the mnemonic is not known, the TYPE
   representation described in [[RFC3597], Section 5](#), MUST be used.
   Implementations that support parsing of presentation format
   records SHOULD be able to read and understand these TYPE
   representations as well.

[2.1.3](#). CSYNC RR Example

The following CSYNC RR shows an example entry for "example.com" that
indicates the NS, A and AAAA bits are set and should be processed by
the parental agent for example.com. The parental agent should pull
data only from a zone using a minimum SOA serial number of 66 (0x42
in hexadecimal).

example.com. 3600 IN CSYNC 66 3 A NS AAAA

The RDATA component of the example CSYNC RR would be encoded on the
wire as follows:

```
0x00 0x00 0x00 0x42             (SOA Serial)
0x00 0x03                       (Flags = immediate | soaminimum)
0x00 0x04 0x60 0x00 0x00 0x08   (Type Bit Map)
```

3. CSYNC Data Processing

The CSYNC record and associated data must be processed as an "all or
nothing" operation set. If a parental agent fails to successfully
query for any of the required records, the whole operation MUST be
aborted. (Note that a query resulting in "no records exist" as
proven by NSEC or NSEC3 is to be considered successful).

Parental agents MAY:

   Process the CSYNC record immediately if the "immediate" flag is
   set. If the "immediate" flag is not set, the parental agent MUST
   NOT act until the zone administrator approves the operation
   through an out-of-band mechanism (such as through pushing a button
   via a web interface).

   Choose not to process the CSYNC record immediately, even if the
   "immediate" flag is set. That is, a parental agent might require
   the child zone administrator approve the operation through an out-
   of-band mechanism (such as through pushing a button via a web
   interface).

Note: how the approval is done out-of-band is outside the scope of
this document and is implementation-specific to parental agents.

3.1. Processing Procedure

The following shows a sequence of steps that SHOULD be used when
collecting and processing CSYNC records from a child zone. Because
DNS queries are not allowed to contain more than one "question" at a
time, a sequence of requests is needed. When processing a CSYNC
transaction request, all DNS queries should be sent to a single
authoritative name server for the child zone. To ensure a single
host is being addressed, DNS over TCP SHOULD be used to avoid
conversing with multiple nodes at an anycast address.

1. Query for the child zone's SOA record

   2. Query for the child zone's CSYNC record

   3. Query for the child zone's data records, as required by the CSYNC
       record's Type Bit Map field

   4. Query for the child zone's SOA record again

   If the SOA records from the first and last steps have different
   serial numbers (for example, because the zone was edited and
   republished during the interval between steps 1 and 4), then the
   CSYNC record obtained in the second set MUST NOT be processed. The
   operation MAY be restarted or retried in the future.

   If the soaminimum flag is set and the SOA serial numbers are equal
   but less than the CSYNC record's SOA Serial Field [RFC1982], the
   record MUST NOT be processed. If state is being kept by the parental
   agent and the SOA serial number is less than the last time a CSYNC
   record was processed, this CSYNC record SHOULD NOT be processed.
   Similarly, if state is being kept by the parental agent and the SOA
   Serial Field of the CSYNC record is less than the SOA Serial Field of
   the CSYNC record from last time, then this CSYNC record SHOULD NOT be
   processed.

   If a failure occurs of any kind while trying to obtain any of the
   required data, or if DNSSEC fails to validate all of the data
   returned for these queries as "secure", then this CSYNC record MUST
   NOT be processed.

   See the "Operational Consideration" section (Section Section 4) for
   additional guidance about processing.

3.2. CSYNC Record Types

   This document defines how the following record types may be processed
   if the CSYNC Type Bit Map field indicates they are to be processed.

3.2.1. The NS type

   The NS type flag indicates that the NS records from the child zone
   should be copied into the parent's delegation information records for
   the child.

   NS records found within the child's zone should be copied verbatim
   and the result published within the parent zone should be an exact
   matching set of NS records. If the child has published a new NS
   record within their set, this record should be added to the parent
   zone. Similarly, if NS records in the parent's delegation records
   for the child contain records that have been removed in the child's

NS set, then they should be removed in the parent's set as well.

Parental agents MAY refuse to perform NS updates if the replacement records fail to meet NS record policies required by the parent zone (e.g. "every child zone must have at least 2 NS records"). Parental agents MUST NOT perform NS updates if there are no NS records returned in a query, as verified by DNSSEC denial of existence protection. This situation should never happen unless the child nameservers are misconfigured.

Note that it is permissible for a child's nameserver to return a CSYNC record that removes the queried nameserver itself from the future NS or address set.

3.2.2. The A and AAAA types

The A and AAAA type flags indicates that the A and AAAA address glue records for in-bailiwick NS records within the child zone should be copied into the parent's delegation information.

Queries should be sent by the parental agent to determine the A and AAAA record addresses for each NS record within a NS set for the child that are in-bailiwick.

Note: only the matching types should be queried. E.g., if the AAAA bit has not been set, then the AAAA records (if any) in the parent's delegation should remain as is. If a given address type is set and the child's zone contains no data for that type (as proven by appropriate NSEC or NSEC3 records), then the result in the parent's delegation records for the child should be an empty set. However, if the end result of processing would leave no glue records present in the parent zone for any of the of the in-bailiwick NS records, then the parent MUST NOT update the glue address records. I.E., if the result of the processing would leave no in-bailiwick A or AAAA records when there are in-bailiwick NS records, then processing of the address records can not happen as it would leave the parent/child relationship without any address linkage.

The procedure for querying for A and AAAA records MUST occur after the procedure, if required, for querying for NS records as defined in Section Section 3.2.1. This ensures that the right set of NS records is used as provided by the current NS set of the child. I.e., for CSYNC records that have the NS bit set, the NS set used should be the one pulled from the child while processing the CSYNC record. For CSYNC records without the NS bit set, the existing NS records within the parent should be used to determine which A and/or AAAA records to update.

[4](#). **Operational Considerations**

   There are a number of important operational aspects to consider when
   deploying a CSYNC RRType.

[4.1](#). **Error Reporting**

   There is no inline mechanism for a parental agent to report errors to
   operators of child zones. Thus, the only error reporting mechanisms
   must be out of band, such as through a web console or over email.
   Parental agents should, at a minimum, at least log errors encountered
   when processing CSYNC records. Child operators utilizing the
   "immediate" flag that fail to see an update within the parental
   agent's specified operational window should access the parental
   agent's error logging interface to determine why an update failed to
   be processed.

[4.2](#). **Child Nameserver Selection**

   Parental agents will need to poll child nameservers in search of
   CSYNC records and related data records.

   Parental agents MAY perform best-possible verification by querying
   all NS records for available data to determine which has the most
   recent SOA and CSYNC version (in an ideal world, they would all be
   equal, but this is not possible in practice due to synchronization
   delays and transfer failures).

   Parental agents may offer a configuration interface to allow child
   operators to specify which nameserver should be considered the master
   to send data queries too. Note that this master could be a different
   nameserver than the publically listed nameservers in the NS set
   (i.e., it may be a "hidden master").

   Parental agents with a large number of clients may choose to offer a
   programmatic interface to let their children indicate that new CSYNC
   records and data are available for polling rather than polling every
   child on a frequent basis.

   Children that wish to phase out a nameserver will need to publish the
   CSYNC record to the nameserver being removed and wait for the
   parental agent to process the published record before turning off the
   service. This is required because the child can not control which
   nameserver in the existing NS set the parental agent may choose to
   query when performing CSYNC processing.

4.3. Out-of-balliwick NS Records

   When a zone contains NS records where the domain-name pointed at does
   not fall within the zone itself, there is no way for the parent to
   safely update the associated glue records. Thus, the child DNS
   operator MAY indicate that the NS records should be synchronized, and
   MAY set any glue record flags (A, AAAA) as well, but the parent will
   only update those glue records which are below the child's delegation
   point.

   Children deploying NS records pointing to domain-names within their
   own children (the "grandchildren") SHOULD ensure the grandchildren's
   associated glue records are properly set before publishing the CSYNC
   record. I.e., it is imperative that proper communication and
   synchronization exist between the child and the grandchild.

4.4. Documented Parental Agent Type Support

   Parental agents that support processing CSYNC records SHOULD publicly
   document the following minimum processing characteristics:

      The fact that they support CSYNC processing

      The Type Bit Map bits they support

      The frequency with which they poll clients (which may also be
      configurable by the client)

      If they support the "immediate" flag

      If they poll a child's single nameserver, a configured list of
      nameservers, or all of the advertised nameservers when querying
      records

      If they support SOA serial number caching to avoid issues with
      regression and/or replay

      Where errors for CSYNC processing are published

      If they support sending queries to a "hidden master".

5. Security Considerations

   This specification requires the use of DNSSEC in order to determine
   that the data being updated was unmodified by third-parties.
   Parental agents implementing CSYNC processing MUST ensure all DNS
   transactions are validated by DNSSEC as "secure". Clients deploying

   CSYNC MUST ensure their zones are signed, current and properly linked
   to the parent zone with a DS record that points to an appropriate
   DNSKEY of the child's zone.

   This specification does not address how to perform bootstrapping
   operations to get the required initial DNSSEC-secured operating
   environment in place. Additionally, this specification was not
   designed to synchronize DNSSEC security records, such as DS pointers,
   or the CSYNC record itself. Thus, implementations of this protocol
   MUST NOT use it to synchronize DS records, DNSKEY materials, CDS
   records, CDNSKEY records, or CSYNC records. Similarly, future
   documents extending this protocol MUST NOT offer the ability to
   synchronize DS, DNSKEY materials, CDS records, CDNSKEY records, or
   CSYNC records. For such a solution, please see the complimentary
   solution [RFC7344] for maintaining security delegation information.

   To ensure that an older CSYNC record making use of the soaminimum
   flag can not be replayed to revert values, the SOA serial number MUST
   NOT be incremented by more than 2^{16} during the lifetime of the
   signature window of the associated RRSIGs signing the SOA and CSYNC
   records. Note that this is independent of whether or not the
   increment causes the 2^{32} bit serial number field to wrap.

6. IANA Considerations

   This document defines a new DNS Resource Record Type, named "CSYNC".
   The IANA is requested to assign a code point from the "Resource
   Record (RR) TYPEs" sub-registry of the "Domain Name System (DNS)
   Parameters" registry (http://www.iana.org/assignments/dns-parameters)
   for this record.

      TYPE   Value   Meaning   Reference
      -----   ------ --------------------------   -----------
      CSYNC   TBD   Child To Parent Synchronization   [This document]

   The IANA is also requested to create and maintain a sub-registry (the
   "Child Synchronization (CSYNC) Flags" registry) of the "Domain Name
   System (DNS) Parameters" registry. The initial values for this
   registry are below. Assignment of new flag values are subject to
   "RFC Required" specifications [RFC5226].

   This registry will hold a set of single-bit "Flags" for use in the
   CSYNC record within the 16 bit Flags field. Thus, a maximum of 16
   flags may be defined.

   The initial assignments in this registry are:

```
   Bit      Flag         Description            Reference
   ----     ------       -------------          -----------
   Bit 0    immediate    Immediately process this [This document,
                         CSYNC record.            Section 3]

   Bit 1    soaminimum   Require a SOA serial   [This document,
                         number greater than the  Section 2.1.1.1]
                         one specified.
```

   For new assignments to be made to this registry, a new standards
   track RFC must be published via a Standards Action.

7. Acknowledgments

   A thank you goes out to Warren Kumari and Olafur Gu[eth]mundsson,
   who's work on the CDS record type helped inspire the work in this
   document, as well as the definition for the "parental agent"
   definition and significant contributions to the text. A thank you
   also goes out to Ed Lewis, with whom the author held many
   conversations with about the issues surrounding parent/child
   relationships and synchronization. Much of the work in this document
   is derived from the careful existing analysis of these three esteemed
   colleagues. Thank you to the following people who have contributed
   text to the document: Matthijs Mekking and Petr Spacek.

   A special thanks goes to Roy Arends, for taking the "bite out of that
   hamburger" challenge while discussing this document.

8. References

8.1. Normative References

   [RFC1982] Elz, R. and R. Bush, "Serial Number Arithmetic", RFC 1982,
              August 1996.

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

   [RFC3597] Gustafsson, A., "Handling of Unknown DNS Resource Record
              (RR) Types", RFC 3597, September 2003.

   [RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S.
              Rose, "Resource Records for the DNS Security Extensions",
              RFC 4034, March 2005.

8.2. Informative References

   [RFC1034] Mockapetris, P., "Domain names - concepts and facilities",
              STD 13, RFC 1034, November 1987.

   [RFC1035] Mockapetris, P., "Domain names - implementation and
              specification", STD 13, RFC 1035, November 1987.

   [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S.
              Rose, "DNS Security Introduction and Requirements",
              RFC 4033, 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.

   [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an
              IANA Considerations Section in RFCs", BCP 26, RFC 5226,
              May 2008.

   [RFC7344] Kumari, W., Gudmundsson, O., and G. Barwood, "Automating
              DNSSEC Delegation Trust Maintenance", RFC 7344,
              September 2014.

Author's Address

   Wes Hardaker
   Parsons, Inc.
   P.O. Box 382
   Davis, CA 95617
   US

   Phone: +1 530 792 1913
   Email: ietf@hardakers.net