

Network Working Group
Internet-Draft
Expires: May 9, 2014

M. Andrews
ISC
November 5, 2013

Automated Delegation of IP6.ARPA reverse zones with Prefix Delegation
draft-andrews-dnsop-pd-reverse-02

Abstract

This document describes a method to automate the delegation of IP6.ARPA reverse zones when performing Prefix Delegations.

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 9, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Method	3
3. Example	4
4. IANA Considerations	4
5. Security Considerations	4
6. Normative References	5
Author's Address	5

1. Introduction

This document describes a method to automate the delegation of IP6.ARPA reverse zones when performing Prefix Delegations.

This will allow home users and small businesses to have IP6.ARPA zones without manual intervention on the part of the ISP.

2. Method

1) CPE device generates a RSA key pair and stores this in non-volatile memory.

2) CPE device generates a DHCPv6 Prefix Delegation [RFC3633] request which includes a KEY-RDATA option (code point TBA), which contains a the rdata of a DNS KEY record containing a RSASHA256 key using the public components of the previously generated RSA key pair.

3) DHCP server updates DNS server based on the prefix it is delegating and the KEY-RDATA, using TSIG [RFC2845] for authentication, and responds with prefix. If this is a new prefix delegation, it will clear out all the old DNS records as part of the delegation process. If there are multiple prefixes being delegated the ISP's DNS server will be updated for all of them. If the delegated prefix is not nibble aligned then the server will update all the reverse apex names that cover the address space, i.e. 1, 2, 4 or 8 KEY records will be added all with the same rdata contents.

4) CPE device configures the nameserver built into it to serve the reverse of the delegated prefixes. Alternatively it may configure other nameservers to serve these zones, however the method to do that is out of scope for this document.

5) CPE device generates a DNS UPDATE [RFC2136] which delegates the reverse name space to itself and others if they have been configured. It uses SIG(0) [RFC2931] to sign the request, with owner name matching the reverse of the delegated prefix.

6) The ISP's DNS server is configured to accept self-signed requests (the owner name used in the SIG(0) signature matches the owner name of the data to be updated). It examines the request, looks at the KEY record added by the DHCPv6 server, and decides whether the request is valid.

3. Example

If 2001:DB8:1:4::/62 is delegated then KEY records for

```
4.0.0.0.1.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA KEY ...
5.0.0.0.1.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA KEY ...
6.0.0.0.1.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA KEY ...
7.0.0.0.1.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA KEY ...
```

will be added. The CPE device will configure the nameservers to serve all of the following zones

```
4.0.0.0.1.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA
5.0.0.0.1.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA
6.0.0.0.1.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA
7.0.0.0.1.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA
```

then will send individual UPDATE messages to delegate each of the reverse zones.

```
% nsupdate -k K4.0.0.0.1.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA
update add 4.0.0.0.1.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA NS ...
send
% nsupdate -k K5.0.0.0.1.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA
update add 5.0.0.0.1.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA NS ...
send
% nsupdate -k K6.0.0.0.1.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA
update add 6.0.0.0.1.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA NS ...
send
% nsupdate -k K7.0.0.0.1.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA
update add 7.0.0.0.1.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA NS ...
send
```

4. IANA Considerations

Allocate a DHCPv6 code point for KEY-RDATA.

5. Security Considerations

The UPDATE requests are all signed. This is a proven method for securing UPDATE requests in the DNS.

As a RSA key is being used there is no issue with key material being sent in the clear.

Only the CPE device and the ISP itself is capable of creating,

updating or destroying the delegation.

6. Normative References

- [RFC2136] Vixie, P., Thomson, S., Rekhter, Y., and J. Bound,
"Dynamic Updates in the Domain Name System (DNS UPDATE)",
RFC 2136, April 1997.
- [RFC2845] Vixie, P., Gudmundsson, O., Eastlake, D., and B.
Wellington, "Secret Key Transaction Authentication for DNS
(TSIG)", RFC 2845, May 2000.
- [RFC2931] Eastlake, D., "Secret Key Transaction Authentication for
DNS (TSIG)", RFC 2931, September 2000.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic
Host Configuration Protocol (DHCP) version 6", RFC 3633,
December 2003.

Author's Address

M. Andrews
Internet Systems Consortium
950 Charter Street
Redwood City, CA 94063
US

Email: marka@isc.org

DNSOP
Internet-Draft
Intended status: Standards Track
Expires: April 24, 2014

W. Hardaker
Parsons, Inc.
October 21, 2013

Child To Parent Synchronization in DNS
draft-hardaker-dnsop-csync-02

Abstract

This document specifies how a child zone in the DNS can publish a record to indicate to a parental agent that it may copy and process certain records from the child zone. The existence and change of the record may be monitored by a parental agent, after which the parent may act on the data appropriately.

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 April 24, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Terminology Used in This Document	3
2. Definition of the CSYNC RRTYPE	4
2.1. The CSYNC Resource Record Format	4
2.1.1. The CSYNC Resource Record Wire Format	4
2.1.2. The CSYNC Presentation Format	6
2.1.3. CSYNC RR Example	6
2.2. CSYNC Data Processing	7
2.2.1. Processing Procedure	7
2.2.2. CSYNC Record Types	8
2.3. Operational Considerations	9
2.3.1. Error Reporting	9
2.3.2. Child Nameserver Selection	9
2.3.3. Documented Parental Agent Type Support	10
2.3.4. Other Considerations	10
3. Security Considerations	10
4. IANA Considerations	11
5. Acknowledgments	11
6. References	11
6.1. Normative References	11
6.2. Informative References	11
Author's Address	12

1. Introduction

This document specifies how a child zone in the DNS can publish a record to indicate to a parental agent that it may copy and process certain records from the child zone. The existence and change of the record may be monitored by a parental agent, after which the parent may act on the data appropriately.

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, to date, that should match are the nameserver (NS) records and any necessary associated address "glue" records (A and AAAA). These records are referred to as "delegation records".

It has been traditionally challenging for children to update their delegation records within the parent's set in a timely fashion. This difficulty is frequently from simple operator laziness or because of 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 RR type (RRType) named "CSYNC" that indicates which delegation records published by a child should be processed by a parental agent and used to update the parent zone's DNS data.

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. For such a solution, please see the complimentary solution [I-D.kumari-ogud-dnsop-cds] for maintaining security delegation information.

1.1. Terminology Used in This Document

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].

This document is aimed at the case where there is an organizational separation of the child and parent. In this case there are many different operating situations. A common case is the Registrant/Registrar/Registry relationship, used by many Top Level Domains in the DNS. In this case, the parent consists of Registrar and Registry, with different rules about what each can do or not do. To remain operating model neutral we will use the neutral word "Parental

Agent" as the entity that uses results of DNS queries discussed in this document to update the delegation records into the parent zone. The entity that performs the changes in the the DNS is called "DNS Publisher".

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 for the child within the parent's zone. Children 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 later in Section Section 2.2.

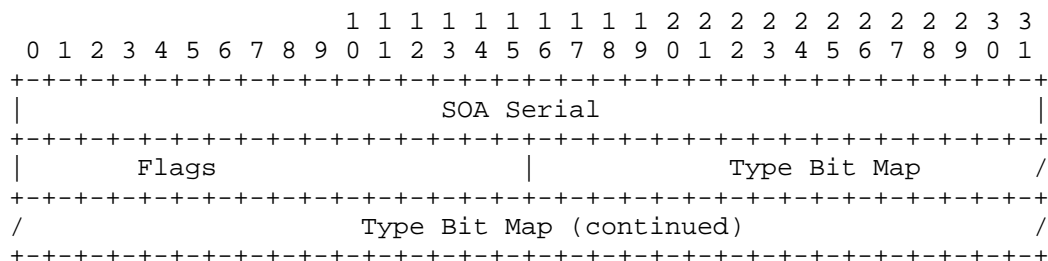
Parental agents MUST process the entire set of child data indicated by the Type Bit Map field (i.e., all record types indicated along with all of the necessary records to support processing of that type) or else parental agents MUST NOT make any changes to parental records at all. Errors due to unsupported 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 expected.

The parental agent MUST perform DNSSEC validation 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 anything other than "Secure" [RFC4034].

2.1. The CSYNC Resource Record Format

2.1.1. The CSYNC Resource Record Wire Format

The CSYNC RDATA consists of the following fields:



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 value is non-zero, parental agents querying children's authoritative servers **MUST NOT** act on data from zones advertising an SOA serial number less than this value. A special value of 0 indicates that no such restriction is in place.

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.

2.1.1.2. The Flags Field

The Flags field contains 16 bits of flags defining operations that affect the processing of the CSYNC record. The flags defined in this document are as follows:

```
0x00 0x01: "immediate"
```

The definitions for how the flags are to be used can be found later in Section 2.2.

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 2.2. 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 copy data blindly; An IETF proposed (or higher) standard specification must exist that defines how the data should be processed for a given bit.

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 1 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 0x01	(Flags [the immediate bit is set])
0x00 0x04 0x60 0x00 0x00 0x08	(Type Bit Map)

2.2. 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 after noticing it 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).

Require that the child zone administrator approve the operation through an out-of-band mechanism (such as through pushing a button via a web interface). I.e., a parental agent MAY choose not to support the "immediate" flag.

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

2.2.1. 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, then the CSYNC record obtained in the second set MUST NOT be processed.

If the SOA serial numbers are equal but less than the CSYNC record's SOA Serial Field, 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 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 2.3) for additional guidance about processing.

2.2.2. CSYNC Record Types

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

2.2.2.1. 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").

2.2.2.2. The A and AAAA types

The A and AAAA type flags indicates that the A and AAAA, respectively, 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.

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 2.2.2.1. This ensures that the right set 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 ones 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.

2.3. Operational Considerations

There are a number of important things to consider when deploying a CSYNC RRTYPE.

2.3.1. 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. 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.

2.3.2. 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. This master may not be one of the publically listed nameservers in the NS set (i.e., it may be a "hidden master").

2.3.3. 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".

2.3.4. Other Considerations

XXX: Discuss complete replacement scenarios and if allowed.

3. 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. For such a solution, please see the complimentary solution [I-D.kumari-ogud-dnsop-cds] for maintaining security delegation information.

4. IANA Considerations

TBD

5. 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 "Parental Agent" and "DNS Publisher" definitions. A thank you also goes out to Ed Lewis, who 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.

6. References

6.1. Normative References

- [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.

6.2. 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.
- [I-D.kumari-ogud-dnsop-cds]

Kumari, W., Gudmundsson, O., and G. Barwood, "Automating
DNSSEC delegation trust maintenance",
draft-kumari-ogud-dnsop-cds-05 (work in progress),
October 2013.

Author's Address

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

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

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: April 21, 2014

J. Abley
Dyn, Inc.
W. Kumari
Google
October 18, 2013

Requirements for a Mechanism for Remote-Triggered DNS Cache Flushes
draft-jabley-dnsop-flush-reqs-00

Abstract

Operational calamities in the DNS happen from time to time, and in many cases problems persist due to DNS caching of bad data. Lacking any robust mechanism to signal that bad data has been flushed, the operators of DNS authority servers often resort to unauthenticated requests for help being sent to mailing lists, the results of which are frequently unsatisfying to all concerned.

This document aims to present requirements for a more robust mechanism by which authoritative server operators can signal to recursive server operators that bad data has been published, and that targetted cache flushes may be beneficial.

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 April 21, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

1. Terminology

This document makes use of the following taxonomy. Note that although it is thought that these terms (and the meanings presented here) are in common use, overloading and ambiguity abounds in practice and hence the definitions presented here should not be considered universally-applicable.

Authoritative Server: A DNS server that serves one or more DNS zones authoritatively, and which does not process recursive queries. An Authoritative Server may function as a Master Server, or a Slave Server, or both.

Master Server: An Authoritative Server with the ability to respond to zone transfer requests from one or more Slave Servers and hence replicate zone data from master to slave.

Slave Server: An Authoritative Server configured to replicate zone data from one or more Master Servers.

Recursive Server: A DNS server that processes Recursive Queries on behalf of Stub Resolvers. Recursive Servers ultimately obtain responses from Authoritative Servers, although particular queries from Stub Resolvers may be satisfied using data stored in a local cache or obtained from one or more other Recursive Servers.

Stub Resolver: A DNS client that communicates with one or more configured Recursive Servers in order to obtain responses to queries on behalf of an application.

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].

2. Introduction

Operational calamities in the DNS happen from time to time, and in many cases problems persist due to DNS caching of bad data. Lacking any robust mechanism to signal that bad data has been flushed, the operators of DNS authority servers often resort to unauthenticated requests for help being sent to mailing lists, the results of which are frequently unsatisfying to all concerned.

This document aims to present requirements for a more robust mechanism by which authoritative server operators can signal to recursive server operators that bad data has been published, and that targetted cache flushes may be beneficial.

3. Use Cases

The following are examples of failures that have been observed to cause service disruption, and that a mechanism meeting the requirements that follow might provide some relief from.

This is all woefully incomplete. Specific examples of domain names (TLD and otherwise) have been omitted in the interests of avoiding undue embarrassment.

3.1. Registrar Compromise, Domain Hijack

Unauthorised access to a registrant's account at a registrar (or some other registrar-level compromise) facilitates the unauthorised redelegation of a domain to new nameservers, which serve malicious data with long TTLs.

Remediation is achieved by restoring the correct delegation. Service disruption continues until the malicious data has expired from Recursive Server caches used by end-users.

3.2. Zone Signing Failure

A failure to sign a zone correctly (e.g. using the wrong keys) results in correct zone data being published with signatures that cannot be validated.

Remediation is achieved by fixing the signing problem (e.g. signing with the correct keys) and publishing a new revision of the zone. Service disruption may continue until particular elements have expired from caches (e.g. apex DNSKEY RRsets).

3.3. Zone Integrity Failure

Publication of an incomplete (e.g. truncated) zone results in missing data, and the absence of that data is subject to negative caching [RFC2308].

Remediation is achieved by resolving the operational problem that led to the incomplete zone being published, and publishing a successor zone that is complete. Service disruption may continue until all negatively-cached elements have expired from caches.

4. Requirements

[These various requirements are somewhat arbitrarily spread over the subsections that follow. Some better organisation would make them more readable.]

4.1. Functional Requirements

1. The mechanism **MUST** be effective; that is, it must be capable of being deployed to the extent that it provides a significant improvement in remediation of zone publication problems.
2. The mechanism **MUST** accommodate a maximally-constrained scope for a flush; that is, the resulting cache flushes **MUST** be constrained to specific parts of the namespace where a flush is beneficial to resolve a problem, and **MUST** minimise collateral damage to other cached data.
3. The mechanism **MUST** be opt-in from the perspective of a Recursive Server operator; that is, no Recursive Server operator should be compelled to act upon a request to flush their cache.
4. The mechanism **MUST** accommodate timely responses to problems.
5. The mechanism **MUST** support idempotency; that is, transmission of multiple identical requests to flush **MUST NOT** result in more than one flush operation in a single cache.
6. The mechanism **SHOULD** require minimal changes to DNS software, but **MIGHT** reasonably involve changes to or deployment of surrounding administrative scaffolding (scripts, etc).

4.2. Operational Requirements

1. It **SHOULD** be possible to automate the reception and processing of a request using the mechanism on a Recursive Server.
2. The mechanism **SHOULD** be simple for Recursive Server operators to implement and operate, since those operators might be the recipient of many requests whereas the operators of Authoritative Servers should only reasonably expect to exercise this mechanism in the event of serious operational failures.

4.3. Manageability Requirements

1. The mechanism's effectiveness **SHOULD** be easily measurable; Authoritative Server operators using the mechanism ought to be able to gauge its effect, and Recursive Server operators ought to

be able to tell whether problem data was present (i.e. whether the remedy actually corresponded to a problem).

4.4. Security Requirements

1. The mechanism MUST be authenticated, such that a Recursive Server operator can trust that a request to flush is authentic.
2. It MUST be possible to rate-limit reception of cache flush requests in order to avoid the mechanism being used as a denial-of-service attack vector.
3. The mechanism MUST NOT facilitate new exploits or compromises against Authoritative Servers or Recursive Servers.

5. IANA Considerations

This document makes no request of the IANA.

6. Security Considerations

This document presents requirements for future work, and does not directly impact the security of the Internet.

Security requirements are described in Section 4.4.

7. Acknowledgements

Your name here, etc.

8. References

8.1. Normative References

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

8.2. Informative References

- [RFC2308] Andrews, M., "Negative Caching of DNS Queries (DNS NCACHE)", RFC 2308, March 1998.

Appendix A. Editorial Notes

This section (and sub-sections) to be removed prior to publication.

A.1. Change History

00 Initial idea, circulated for the purposes of entertainment.

Authors' Addresses

Joe Abley
Dyn, Inc.
470 Moore Street
London, ON N6C 2C2
Canada

Phone: +1 519 670 9327
Email: jabley@dyn.com

Warren Kumarui
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043

Email: warren@kumari.net

Network Working Group
Internet-Draft
Intended status: BCP
Expires: April 21, 2014

J. Abley
Dyn, Inc.
October 18, 2013

DNS Reverse Mapping for Multicast Addresses
draft-jabley-multicast-ptr-00

Abstract

The mapping of IPv4 and IPv6 addresses to names using the Domain Name System (DNS) is colloquially known as "Reverse Mapping". Reverse Mapping support for registered multicast address assignments in IPv4 is currently incomplete and ad-hoc; in IPv6 there is no support at all.

This document describes procedures to be followed that will result in more systematic and predictable support for Reverse Mapping for IPv4 multicast address assignments, and introduces analogous support for IPv6.

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 April 21, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. General Approach	4
3. Naming Scheme	5
3.1. IPv4 Multicast Addresses	5
3.2. IPv6 Multicast Addresses	5
4. Use of MCAST.ARPA	6
5. IAB Considerations	7
6. IANA Considerations	8
6.1. Registry Changes	8
6.1.1. IPv6 Multicast Scope Registry	8
6.1.2. IPv4 Multicast Address Space Registries	8
6.1.3. IPv6 Multicast Address Space Registries	9
6.2. Delegation of MCAST.ARPA and MCAST6.ARPA	9
6.3. Initial Zone Contents	9
6.4. Process Changes	11
6.5. Ongoing Support for MCAST.NET	11
7. Security Considerations	13
8. Acknowledgements	14
9. References	15
9.1. Normative References	15
9.2. Informative References	15
Appendix A. Editorial Notes	16
A.1. Change History	16
Author's Address	17

1. Introduction

The Domain Name System (DNS), as originally specified in [RFC1034] and [RFC1035], provides support for the mapping of IPv4 addresses to names using a namespace convention within the IN-ADDR.ARPR domain and the PTR resource record type.

The analogous mapping of IPv6 address to names is specified in [RFC3596], adopting a similar namespace convention within the IP6.ARPA domain.

Multicast addresses are assigned by the IANA, and assignments are documented in various IANA registries.

For IPv4, Reverse Mapping of assigned multicast addresses to names has historically been provided in an ad-hoc and incomplete fashion, without tight coordination with IANA multicast address assignment processes. Names assigned to IPv4 multicast addresses have been chosen somewhat arbitrarily within the MCAST.NET domain. For IPv6, no Reverse Mapping is provided.

This document describes procedures to be followed by the IANA to support predictable and consistent Reverse Mapping for registered multicast addresses in IPv4 and IPv6.

2. General Approach

This document specifies extensions to existing IPv4 and IPv6 multicast registries to include a mandatory column "DNS Label". This field is required to be populated with a unique, valid DNS label for all future multicast address assignments except in the case where reverse mapping for an address is explicitly not desirable.

The procedures at the IANA relating to multicast address assignment are extended to include the provisioning of appropriate changes in the DNS at the time of registration or de-registration of any multicast addresses. Specific actions requested of the IANA are described in Section 6.

Names for multicast addresses are assigned under MCAST.ARPA for IPv4 addresses, and MCAST6.ARPA for IPv6 addresses. The naming schemes to be used in each case are described in Section 3. The use of MCAST.ARPA rather than MCAST.NET is discussed in Section 4.

3. Naming Scheme

3.1. IPv4 Multicast Addresses

Each assigned IPv4 multicast address has an accompanying DNS Label. The name associated with an IPv4 multicast address with DNS Label SOMENAME is SOMENAME.MCAST.ARPA.

For example, suppose the assigned IPv4 multicast address 224.0.1.1 has the DNS Label "NTP". The address 224.0.1.1 maps to the name "NTP.MCAST.ARPA"; the name "NTP.MCAST.ARPA" maps to the address 224.0.1.1.

3.2. IPv6 Multicast Addresses

Each assigned IPv6 multicast address has an accompanying DNS Label ("Address Label").

IPv6 multicast addresses may be of fixed or variable scope. The naming scheme for these addresses incorporates a scope identifier using an additional DNS label ("Scope Label"), specified in a dedicated registry (see Section 6.1.1). Both fixed and variable scope multicast addresses use the same naming scheme.

The name associated with an IPv6 multicast address with Scope Label SCOPE and Address Label SOMENAME is SOMENAME.SCOPE.MCAST6.ARPA.

For example, suppose ff01::1 is an assigned IPv6 multicast address with Scope Label "NODE-LOCAL" and Address Label "ALL-NODES". The address ff01::1 maps to the name "ALL-NODES.NODES-LOCAL.MCAST6.ARPA"; the name "ALL-NODES.NODES-LOCAL.MCAST6.ARPA" maps to the address ff01::1.

The variable-scope multicast address ff0x::fb will have different Reverse Mapping depending on the scope specified in the address (i.e. the value of x), although the Address Label in each case will be the same ("MDNSV6"). The Site-Local address ff05::fb has an associated Scope Label "SITE-LOCAL", and is therefore named MDNSV6.SITE-LOCAL.MCAST6.ARPA. The Link-Local address ff02::fb has an associated Scope Label "LINK-LOCAL" and hence is named MDNSV6.LINK-LOCAL.MCAST6.ARPA.

4. Use of MCAST.ARPA

Use of the MCAST.ARPA domain rather than MCAST.NET for IPv4 multicast addresses is specified for the same reasons that led IP6.INT to be superceded by "IP6.ARPA" [RFC3152].

It is prudent to assume that hard-coded assumptions about names in MCAST.NET exist, and will persist for some time. This document specifies that names in the MCAST.ARPA domain also be available in the MCAST.NET domain, to provide support for software with those assumptions. Ongoing support for the MCAST.NET zone is described in Section 6.5.

It is possible that in the future empirical measurement will confirm that the use of names under MCAST.NET is no longer required and that provisioning of the MCAST.NET domain can safely cease. This document provides no such measurement and makes no such recommendation, however.

5. IAB Considerations

This document proposes a delegation within the ARPA domain, and, in accordance with [RFC3172], IAB review and approval of the delegation of MCAST.ARPA and MCAST6.ARPA as described in Section 6.2 is required.

Once IAB approval has been obtained, this section may be removed prior to publication or updated to include text that confirms the IAB's decision, at the IAB's discretion.

6. IANA Considerations

6.1. Registry Changes

6.1.1. IPv6 Multicast Scope Registry

The IANA is directed to create a new registry as follows:

Registry Name: IPv6 Multicast Address Scopes

Registration Procedure: Standards Action

Reference: This document

Schema: See initial contents, below. Note that the Scope Value and DNS Label fields are mandatory for all rows, and that values chosen for future DNS Label fields are required to be unique within this registry.

The initial contents of this new registry should be:

Scope Value	DNS Label	Scope Name	Reference
0x0	(none)	Reserved	[RFC4291]
0x1	NODE-LOCAL	Node-Local Scope	[RFC4291]
0x2	LINK-LOCAL	Link-Local Scope	[RFC4291]
0x3	(none)	Reserved	[RFC4291]
0x4	ADMIN-LOCAL	Admin-Local Scope	[RFC4291]
0x5	SITE-LOCAL	Site-Local Scope	[RFC4291]
0x8	ORG-LOCAL	Organisation-Local Scope	[RFC4291]
0xE	GLOBAL	Global Scope	[RFC4291]

The IANA may add "Date Registered" and "Last Revised" columns to the schema at its discretion.

6.1.2. IPv4 Multicast Address Space Registries

The IANA is directed to add a mandatory "DNS Label" column to all IPv4 Multicast Address Space registries. The initial contents of the

DNS Label field for each row should be taken from the corresponding MCAST.NET zone owner names where available; addresses with no existing mapping in MCAST.NET should have DNS Labels assigned by the IANA at their discretion.

All existing assignments should have a DNS Label assigned. A DNS Label should be mandatory for all future registrations. DNS Labels are required to be unique for all IPv4 multicast address assignments.

6.1.3. IPv6 Multicast Address Space Registries

The IANA is directed to add a mandatory "DNS Label" column to all IPv6 Multicast Address Space registries. The initial contents of the DNS Label field for each row should be assigned by the IANA at their discretion.

All existing assignments should have a DNS Label assigned. A DNS Label should be mandatory for all future registrations. DNS Labels are required to be unique for all IPv6 multicast address assignments.

6.2. Delegation of MCAST.ARPA and MCAST6.ARPA

The IANA is directed to create and host the MCAST.ARPA and MCAST6.ARPA zones on name servers of their choosing. The MCAST.ARPA and MCAST6.ARPA zones should be signed using DNSSEC, with DNSSEC parameters chosen by the IANA. The initial zone contents should be as described in Section 6.3.

The IANA is directed to provision secure delegations for the MCAST.ARPA and MCAST6.ARPA zones from the ARPA zone (i.e. delegations with accompanying DS RRsets).

6.3. Initial Zone Contents

The IANA is directed to populate the MCAST.ARPA and MCAST6.ARPA zones, and the corresponding reverse mapping zones under IN-ADDR.ARPA and IP6.ARPA, directly from the IPv4 and IPv6 Multicast Address Registries, amended as described in Section 6.1.

As an example, if the IPv6 Variable Scope Multicast Addresses sub-registry contained the following entry:

Address(es)	Description	DNS Label
FF0X::FB	mDNSv6	MDNSV6


```
$ORIGIN MCAST.ARPA.  
;  
; SGI-Dogfight address  
;  
SGI-DOG                A          224.0.1.2  
  
$ORIGIN 224.IN-ADDR.ARPA.  
;  
; SGI-Dogfight address  
;  
2.1.0                  PTR        SGI-DOG.MCAST.ARPA.
```

6.4. Process Changes

The IANA is directed to require a valid and unique DNS Label to be specified within the existing processes of multicast address assignment in IPv4 and IPv6.

The IANA is further directed to maintain the MCAST.ARPA, MCAST6.ARPA and related domains under IP6.ARPA and IN-ADDR.ARPA such that any additions, changes or deletions from the corresponding address registries are reflected accurately in the DNS.

6.5. Ongoing Support for MCAST.NET

IANA is directed to remove all non-apex resource records from the MCAST.NET zone and to add an apex DNAME [RFC6672] with target MCAST.ARPA. The intention is to provide backwards compatibility for software that has hard-coded assumptions about naming conventions for IPv4 multicast addresses.

For example, the following describes the result of this change for MCAST.NET SOA serial 2012123836, with DNSSEC resource records omitted for clarity:

\$ORIGIN MCAST.NET.

; beginning of zone

```
@      SOA      SNS.DNS.ICANN.ORG. NOC.DNS.ICANN.ORG. (
                                2012123836
                                7200
                                3600
                                604800
                                3600 )
```

```
NS      A.IANA-SERVERS.NET.
NS      B.IANA-SERVERS.NET.
NS      C.IANA-SERVERS.NET.
NS      NS.ICANN.ORG.
```

```
DNAME   MCAST.ARPA.
```

; end of zone

7. Security Considerations

This document presents no known additional security concerns to the Internet.

8. Acknowledgements

Your name here, etc.

9. References

9.1. Normative 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.
- [RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", RFC 3596, October 2003.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

9.2. Informative References

- [RFC3152] Bush, R., "Delegation of IP6.ARPA", BCP 49, RFC 3152, August 2001.
- [RFC3172] Huston, G., "Management Guidelines & Operational Requirements for the Address and Routing Parameter Area Domain ("arpa")", BCP 52, RFC 3172, September 2001.
- [RFC6672] Rose, S. and W. Wijngaards, "DNAME Redirection in the DNS", RFC 6672, June 2012.

Appendix A. Editorial Notes

This section (and sub-sections) to be removed prior to publication.

A.1. Change History

00 Initial draft, circulated for the purposes of entertainment.

Author's Address

Joe Abley
Dyn, Inc.
470 Moore Street
London, ON N6C 2C2
Canada

Phone: +1 519 670 9327
Email: jabley@dyn.com

HOMENET
Internet-Draft
Intended status: Standards Track
Expires: April 24, 2014

D. Migault (Ed)
Orange
October 21, 2013

DNSSEC Validators DHCP Options
draft-mglt-homenet-dnssec-validator-dhc-options-02.txt

Abstract

DNSSEC provides data integrity and authentication for DNSSEC validators. However, without valid trust anchor(s) and an acceptable value for the current time, DNSSEC validation cannot be performed. As a result, there are multiple cases where DNSSEC validation MUST NOT be performed. In addition, this list of exceptions is expected to become larger over time.

Considering an increasing number of cases where DNSSEC is disabled adds complexity to the DNSSEC validator implementations and increases the vectors that disable security.

This document assumes that DNSSEC adoption by end devices requires that end devices MUST be able to support a DNSSEC validation always set. This MUST be valid today as well as in the future.

This document describes DHCP Options to provision the DHCP Client with valid trust anchors and time so DNSSEC validation can be performed.

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 April 24, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Requirements notation	2
2. Introduction	2
3. Threat Model	3
3.1. Motivations for providing DNSSEC Trust Anchor	3
3.2. Motivations for providing Time	5
4. Terminology	5
5. DHCP DNSSEC Trust Anchor Options	6
5.1. DHCP DNSSEC KSK RR Trust Anchor Options	6
5.2. DHCP DNSSEC KSK CERT Trust Anchor Options	6
6. DHCP Time Option	7
7. DHCP Client Behavior	8
8. DHCP Server Behavior	9
9. DHCP Relay Agent Behavior	10
10. IANA Considerations	10
11. Security Considerations	10
12. Acknowledgment	10
13. References	10
13.1. Normative References	10
13.2. Informational References	11
Appendix A. Document Change Log	12
Author's Address	12

1. Requirements notation

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].

2. Introduction

DNSSEC [RFC4033], [RFC4034], [RFC4035] adds data authentication and integrity checks to DNS [RFC1034], [RFC1035]. For signature validation, DNSSEC requires a trust anchor such as the Key Signing

Key (KSK) of the Root Zone or any other zone. Without a trust anchor, DNSSEC validation cannot be performed. In addition KSKs and signatures are valid for a given period of time. As a result, DNSSEC validation cannot be performed if time shifting is too large.

This document considers DHCP DNSSEC Trust Anchor Option and DHCP Time Option to provision a device with trusted KSKs and current time. Although our priority is to provide the Root Zone KSK, we also consider the case other trusted KSK MAY be provided, for example, if a Zone does not provide secure delegation, or to mitigate badly configured DNSSEC zones (like TLDs zones).

The main motivation for these DHCP Options is that DHCP enabled devices have DNSSEC validation always set and do not need to perform DNS resolution without DNSSEC validation. In fact, enabling DNS with no validation represents a potential way to remove security and MAY be used by attackers. Similarly, DNSSEC configuration implemented in the end users device, MAY not consider future cases and MAY introduce vulnerabilities. DHCP Options prevent this as long as the relationship between DHCP Client and DHCP Server is trusted.

This document assumes that the channel between the DHCP Client and the DHCP Server is trusted and secured with DHCP mechanisms described in [RFC3315], or IPsec [RFC4301].

3. Threat Model

This document addresses the case of a device configured with DNSSEC validation set that is plugged in, gets connectivity (using DHCP for example), but fails DNSSEC resolutions because its trust anchor KSK is not valid anymore or its local time is not valid.

This threat mainly addresses devices that can be switched off for a long period of time or devices that MAY be off-shelves for a long time before being plugged in. CPEs as well as any homenet devices are concerned by this use case.

This threat also addresses DNSSEC emergency key roll over operations. Devices that have cached the out-of-date KSK will not be able to check the signatures until the TTL has expired on all caches.

This document proposes DHCP Options that provide the necessary parameters to perform DNSSEC validation. These Options MUST be used on a trusted network over a trusted channel between the DHCP Client and the DHCP Server. These options MAY be used in conjunction of additional mechanisms.

3.1. Motivations for providing DNSSEC Trust Anchor

The first motivation for providing trusted KSKs is to provide automatic configuration of devices to enable DNSSEC validation. This avoids validator initial KSK provisioning issue as well as KSK roll over issues.

A validator MAY not be able to perform signature check with an authenticated KSK because:

- 1) It does not have a trust anchor (like the Root Zone KSK)
- 2) The KSK MAY have been authenticated, stored or cached with an expiration date valid but is not valid anymore. This MAY happen in the case of an emergency key roll over, if the device has been offline during the key roll over, or if the key roll over is not performed as described in [DPS-KSK], [RFC5011].
- 3) The chain of trust MAY have been broken. This can happen to non Root Zone KSK only and MAY not involve the responsibility of the owner of the zone. The deeper the Zone is in the hierarchy, the more likely this happens.
- 4) A DNSSEC zone MAY have been badly signed or a KSK MAY have been badly generated. The DNSSEC MAY be correct, but DNSSEC validator MAY keep for a long time the badly generated KSK, ZSK...

The goal of the DHCP DNSSEC Trust Anchor Option is to provide these validators trusted anchors like the Root Zone KSK, as well as other KSKs (TLDs...) so the validator has the proper KSKs to perform DNSSEC validation.

Most documents are currently focused on the Root Zone KSK for which recommendations and alternative mechanisms have been described. [I-D.jabley-dnsop-validator-bootstrap] provides guide lines on how to retrieve and select DNSSEC Trust Anchors. Section 5.3 and [I-D.jabley-dnssec-trust-anchor] describes mechanisms to retrieve securely the Root Zone KSK relying on TLS security. It suggests to use insecure DNS resolution to set HTTPS connections. Using HTTPS requires downloading the keyDigest id (key-label) from <https://data.iana.org/root-anchors/root-anchors.xml>, followed by an HTTPS request at <https://data.iana.org/root-anchors/key-label.crt> to get the whole certificate.

The key advantages of the DHCP DNSSEC Trust Anchor Option described in this document are that we extend the mechanism to any KSK, and validators can set DNSSEC validation for all DNS queries. However, we do not see any contradiction between recommendations provided by [I-D.jabley-dnsop-validator-bootstrap] and [I-D.jabley-dnssec-trust-

anchor] and believe the principle described in these documents SHOULD be applied by the validators. Note also that DHCP DNSSEC Trust Anchor Option only benefits to validators that are configured via DHCP.

To recover from a DNSSEC failure and remove a particular data from cache, [I-D.jabley-dnsop-dns-flush] suggests to use a NOTIFY message between Authoritative Servers and Resolvers. This mechanism is set between Recursive Server and Authoritative Servers with a specific trusted relationship. This is probably a selection of TLDs. This document, does not address the DNSSEC failure over Recursive Servers, but addresses more specifically DHCP configured devices. These are typically CPEs or End Users. We believe that configuring and restarting DNSSEC validators with DHCP Option, is an easier way to cope with this issue. First the trust relation between DHCP Server already exists, we do not need additional trusted channel between Authoritative Servers or eventually the Recursive Servers. Then basic implementations of stub resolvers, in CPE or desktops may not address NOTIFY message.

3.2. Motivations for providing Time

KSKs and signatures are always associated to an expiration time. As a result, DNSSEC validation requires that the validator knows the current time.

A number of mechanisms exists like [TSLDATE] or [RFC5905] for setting the time of the device. In addition, [RFC5908] provides a Network Time Protocol (NTP) Server Option for DHCP. The DHCP Time Option described in this document differs from [RFC5908] as it provides an estimation of the current time, instead of providing the NTP servers location information. The time value provided by the DHCP Time Option should be used only if previously mentioned mechanisms are either not implemented on the device or are unavailable. One of the reason MAY be that you MAY need valid DNS(SEC) resolution to use these protocols. The time provided by the DHCP Time Option does not have the accuracy of NTP and SHOULD be considered as a best effort value. [I-D.jabley-dnsop-validator-bootstrap] also recommends that when time has not been verified by the validator, the signature validation SHOULD be done with time off.

The key advantage of the DHCP Time Option is that it makes possible to have DNSSEC validation always set. It limits the possible DNSSEC validation variants which potentially expose the device to disable DNSSEC validations. Note also that DHCP Time Option only benefits to validators that are configured via DHCP.

4. Terminology

5. DHCP DNSSEC Trust Anchor Options

This section describes two options:

- DHCP DNSSEC KSK Trust Anchor Options: carries the KSK RRset as described in [RFC1035] with a DNSKEY RDATA as described in [RFC4033]. This data is not integrity protected, nor it can be authenticated. Such data SHOULD be trusted over a trusted DHCP channel.
- DHCP DNSSEC CERT Trust Anchor Options: Carries a certificate encoded as described in [RFC4398]. The advantage of the Certificate is that it enables authentication of the received information by a trusted party. For example, CPE providers MAY provide a trusted certification authority. Unlike DNSSEC key roll over, the CPE provider controls the key roll over of the certification authority it provides.

5.1. DHCP DNSSEC KSK RR Trust Anchor Options

The DHCP DNSSEC KSK Trust Anchor Option provides the RRset as mentioned in the DNS(SEC) Zone. In other words, it carries the RR as defined in Section 3.2. of [RFC1035] and a RDATA DNSKEY as defined in Section 2.1 of [RFC4033]. As the RR has a variable length, the DHCP DNSSEC KSK Trust Anchor Options follows the recommendation format of Section 5.9 of [I-D.ietf-dhc-option-guidelines].

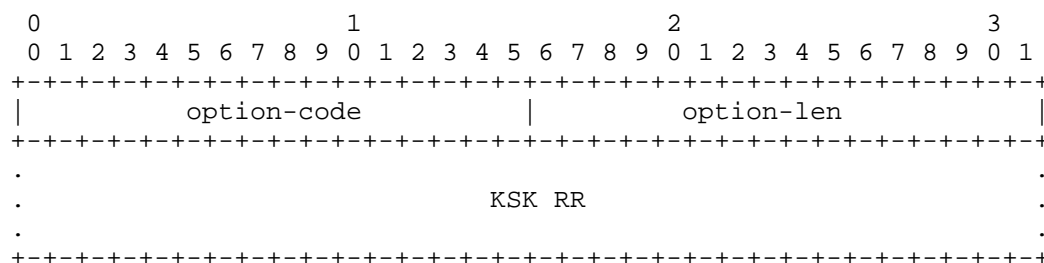


Figure 1: DHCP DNSSEC KSK Trust Anchor Options
Payload Description

- option-code: OPTION_DNSSEC_KSK_RR_TRUST_ANCHOR
- option-len: An unsigned integer giving the length of the KSK RR field in this option in octets

5.2. DHCP DNSSEC KSK CERT Trust Anchor Options

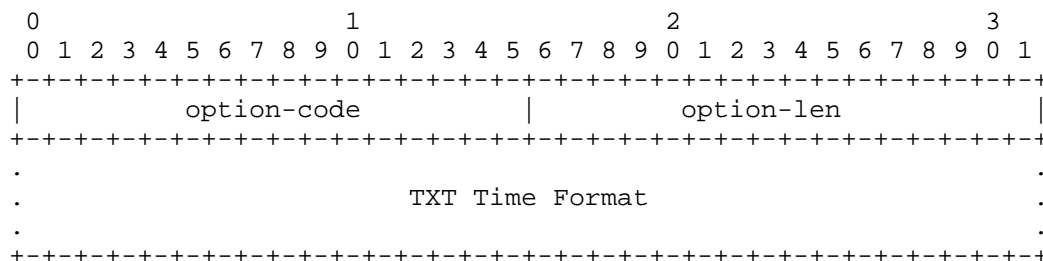


Figure 2: DHCP Time Options
Payload Description

- option-code: OPTION_TIME
- option-len: A string representing the Time

7. DHCP Client Behavior

DHCP DNSSEC KSK Trust Anchor Option, DHCP DNSSEC CERT Trust Anchor Option or DHCP Time Option described in this document are intended for DNSSEC validation. If a connected device is not performing DNSSEC validation, it MUST NOT send a DHCP an Option Request DHCP Option (ORO) [RFC3315] for any of these options, and MUST ignore all these options if provided by the DHCP Server.

The DHCP sends a DHCP ORO for one or multiple options described in the document. Motivations for sending this Option Request DHCP Option is out of scope of the document. It could be a device switched off for a long time, a device that cannot validate the DNSSEC responses.

A channel is considered trusted if 1) the DHCP Server is trusted and authenticated and 2) exchanged data between the DHCP Client and the DHCP Server is integrity protected. IPsec [RFC4301], for example, MAY be used to establish a secure channel.

Over a trusted channel, the DHCP Client that performs DNSSEC validation MAY send an ORO for any of the DHCP DNSSEC KSK Trust Anchor Option, the DHCP DNSSEC CERT Trust Anchor Option or the DHCP Time Option to a DHCP Server.

Over a trusted channel, the DHCP Client that performs DNSSEC validation SHOULD consider the DHCP DNSSEC KSK Trust Anchor Option, the DHCP DNSSEC CERT Trust Anchor Option or the DHCP Time Option sent by the DHCP Server.

Over a non trusted channel, the DHCP Client MAY only send ORO for a DHCP DNSSEC CERT Trust Anchor Option. This option is the only one that MAY be considered by the DHCP Client if sent by the DHCP Server. If the DHCP Client does not trust the signer of the certificate, the option MUST be ignored.

When a DHCP DNSSEC KSK Trust Anchor Option or a DHCP DNSSEC CERT Trust Anchor Option is accepted by the DHCP Client, it MUST remove overwrite old values for the KSK with the new one.

When a DHCP Time Option is accepted by the DHCP Client, it MUST check the difference between its clock and the time provided by the Option. It SHOULD overwrite its clock value only if the difference is too large.

In any other case, ORO requests MUST NOT be sent by the DHCP Client, and options received by the DHCP Server MUST NOT be considered by the DHCP Client. The remaining of the section details when the options MUST NOT be requested by the DHCP Client and MUST be ignored by the DHCP Client when received by the DHCP Server.

The DHCP Client MUST NOT send an ORO for a DHCP DNSSEC KSK Trust Anchor Option, a DHCP DNSSEC CERT Trust Anchor Option or a DHCP Time Option to a DHCP Server that is either not trusted or not authenticated.

All DHCP DNSSEC KSK Trust Anchor Option, a DHCP DNSSEC CERT Trust Anchor Option or a DHCP Time Option received from DHCP Server that is not authenticated or that is not trusted MUST be ignored by the DHCP Client.

The DHCP Client MUST NOT send an ORO for a DHCP DNSSEC KSK Trust Anchor Option or a DHCP Time Option to a trusted DHCP Server over an untrusted channel. A DHCP DNSSEC CERT Trust Anchor Option MAY be requested over an untrusted channel since the certificate is signed and thus can be authenticated. A DHCP DNSSEC CERT Trust Anchor Option signed by an untrusted authority MUST be ignored by the DHCP Client.

All DHCP DNSSEC KSK Trust Anchor Option or a DHCP Time Option received from DHCP Server over a channel that is not trusted MUST be ignored by the DHCP Client.

8. DHCP Server Behavior

The DHCP Server SHOULD properly answer with the requested options in the ORO, even if the DHCP Server does not consider the channel with DHCP Client as trusted.

The DHCP Server MAY also provide DHCP DNSSEC KSK Trust Anchor Option, DHCP DNSSEC CERT Trust Anchor Option or DHCP Time Option without being requested by the DHCP Client. This could for example prevent failures not detected by the DHCP Client.

9. DHCP Relay Agent Behavior

The DHCP Options described in the document do not impact the Relay Agent.

10. IANA Considerations

The DHCP options detailed in this document is:

- OPTION_DNSSEC_KSK_RR_TRUST_ANCHOR: TBD
- OPTION_DNSSEC_KSK_CERT_TRUST_ANCHOR: TBD
- OPTION_TIME: TBD

11. Security Considerations

Security has been discussed in the "DHCP Client Behavior Section". As information contained in the payloads are use to enable signature validation, these pieces of information MUST be considered only when issued by a trusted party, and when integrity protection is provided.

12. Acknowledgment

Bringing DNSSEC in Home Networks discussion has started during the IETF87 in Berlin with Ted Lemon, Ralph Weber, Normen Kowalewski, and Mikael Abrahamsson. An email discussion has also been initiated by Jim Gettys with among others, helpful remarks from Paul Wouters, Joe Abley, Michael Ridchardson.

13. References

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

- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.
- [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.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.
- [RFC4398] Josefsson, S., "Storing Certificates in the Domain Name System (DNS)", RFC 4398, March 2006.
- [RFC5011] StJohns, M., "Automated Updates of DNS Security (DNSSEC) Trust Anchors", STD 74, RFC 5011, September 2007.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.
- [RFC5905] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.
- [RFC5908] Gayraud, R. and B. Lourdelet, "Network Time Protocol (NTP) Server Option for DHCPv6", RFC 5908, June 2010.

13.2. Informational References

- [DPS-KSK] Ljunggren, F., Okubo, T., Lamb, R., and J. Schlyter, "DNSSEC Practice Statement for the Root Zone KSK Operation", Root DNSSEC Design Team, URL: <http://www.root-dnssec.org/wp-content/uploads/2010/06/icann-dps-00.txt>, 2010.

[I-D.ietf-dhc-option-guidelines]

Hankins, D., Mrugalski, T., Siodelski, M., Jiang, S., and S. Krishnan, "Guidelines for Creating New DHCPv6 Options", draft-ietf-dhc-option-guidelines-14 (work in progress), September 2013.

[I-D.jabley-dnsop-dns-flush]

Abley, J., "A Mechanism for Remote-Triggered DNS Cache Flushes (DNS FLUSH)", draft-jabley-dnsop-dns-flush-00 (work in progress), June 2013.

[I-D.jabley-dnsop-validator-bootstrap]

Abley, J. and D. Knight, "Establishing an Appropriate Root Zone DNSSEC Trust Anchor at Startup", draft-jabley-dnsop-validator-bootstrap-00 (work in progress), January 2011.

[I-D.jabley-dnssec-trust-anchor]

Abley, J., Schlyter, J., and G. Bailey, "DNSSEC Trust Anchor Publication for the Root Zone", draft-jabley-dnssec-trust-anchor-07 (work in progress), June 2013.

[TSLDATE] error, IO., "tlsdate: secure parasitic rdate replacement", URL: <https://github.com/ioerror/tlsdate>, 2013.

Appendix A. Document Change Log

[RFC Editor: This section is to be removed before publication]

-00: First version published.

Author's Address

Daniel Migault
Orange
38 rue du General Leclerc
92794 Issy-les-Moulineaux Cedex 9
France

Phone: +33 1 45 29 60 52
Email: mglt.ietf@gmail.com