

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: August 29, 2013

S. Bortzmeyer
AFNIC
February 25, 2013

JSON format to represent DNS data
draft-bortzmeyer-dns-json-01

Abstract

This document describes a profile of JSON to represent DNS data.

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 August 29, 2013.

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	2
---------------------------	---

2.	Requirements notation	2
3.	The format	3
3.1.	General rules	3
3.2.	Resource records	3
3.3.	DNS response	6
3.4.	Zone file	7
3.5.	Examples	7
3.6.	Open questions	8
4.	Security considerations	9
5.	References	9
5.1.	Normative References	9
5.2.	Informative References	9
	Author's Address	10

1. Introduction

The JavaScript Object Notation (JSON) format is specified in [RFC4627]. It is a structured data format suitable for a wide range of applications. It is specially popular on the Web, due to its JavaScript roots, but can be found in many other contexts.

The Domain Name System (DNS) is specified in [RFC1034] and [RFC1035]. It is one of the most important infrastructure components of the Internet. DNS data is today typically exchanged using two formats: the "zone file" format (partially) described in section 5 of [RFC1035] and the "wire format" of the section 4 for [RFC1035]. Other formats have been suggested, for an easier exchange of data, or for using DNS in new applications, such as DNS "looking glasses" or gateways to get DNS data over protocols such as HTTP ([RFC2616]).

For instance, a mechanism have been suggested for DNS data in XML, in [I-D.mohan-dns-query-xml].

This document suggests using the JSON format to represent DNS data. Note that a similar JSON-like (rather than JSON) description of DNS data already exists in [getdns].

Also note that some representations of DNS data use a data model which is quite close from the JSON one, even if the concrete syntax is different (for instance [dnspython]).

2. 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].

3. The format

3.1. General rules

Most data is represented by JSON objects, with their named members. It is common to omit some of these members, to save bandwidth or by pure laziness. So, clients who consume this sort of JSON objects should not assume every member is present. THIS IS AN IMPORTANT RULE (see Section 3.6, Paragraph 2 for a discussion).

3.2. Resource records

DNS resource records are JSON objects. The following members are common to all record types:

- o Name (owner name)
- o Type
- o Class
- o Time to live (TTL)

The other members depend on the record type. The following list gives the resource record type mnemonic and the JSON members for this type:

- o A:
 - * Address
- o AAAA:
 - * Address
- o MX:
 - * Preference
 - * MailExchanger
- o NS:
 - * Target
- o PTR:
 - * Target

- o CNAME:
 - * Target
- o SOA:
 - * MaintainerName
 - * MasterServerName
 - * Serial
 - * Refresh
 - * Retry
 - * Expire
 - * NegativeTtl
- o DNSKEY:
 - * Algorithm
 - * Length
 - * Flags
 - * Tag
- o DS:
 - * DelegationKey
 - * DigestType
- o DLV:
 - * DelegationKey
 - * DigestType
- o NSEC3PARAM:
 - * Algorihm
 - * Flags

- * Salt
- * Iterations
- o SSHFP:
 - * Algorithm
 - * DigestType
 - * Fingerprint
- o NAPTR:
 - * Flags
 - * Order
 - * Services
 - * Preference
 - * Regexp
 - * Replacement
- o SRV:
 - * Server
 - * Port
 - * Priority
 - * Weight
- o LOC:
 - * Longitude
 - * Latitude
 - * Altitude
- o SPF:
 - * Text

Note there is no concept of resource record sets (see Section 3.6, Paragraph 3 for a discussion).

3.3. DNS response

A DNS response is represented as a JSON object with a member named "Query". The main members of this object (the names are self-explanatory) are:

- o QuestionSection
- o AnswerSection
- o AdditionalSection
- o AuthoritySection
- o ReturnCode (alphabetical, e.g. NOERROR, NXDOMAIN, SERVFAIL, etc)
- o ID
- o AA (Authoritative Answer)
- o TC (TrunCation)
- o RD (Recursion Desired)
- o RA (Recursion Available)
- o AD (Authentic Data)
- o Query

The Question Section is an object with members Qname, Qtype and Qclass. The other three sections are JSON arrays, each DNS record is an item in the array. They may be empty arrays (for instance, if the request returns NOERROR, ANSWER=0, the AnswerSection will be an empty array).

The Query object has members about the query: Duration is the time taken to process the request, Server the resolver used (preferably as an IP address).

3.4. Zone file

A DNS zone file is represented as a JSON object with a member named "Zone". The main member of this object is an array of resource records.

The member "Name" cannot be omitted in resource records (unlike the text format of [RFC1035], JSON does not guarantee the order of records, so the trick of "previous resource record" does not work). But you can use relative names, and @ to denote the origin.

3.5. Examples

```
{ "Query": { "Server": "127.0.0.1" },
  "AnswerSection": [
    { "Name": "bortzmeyer.fr.",
      "TTL": 3600,
      "MasterServerName": "ns3.bortzmeyer.org.",
      "MaintainerName": "hostmaster.bortzmeyer.org.",
      "Serial": 2012060801, "Expire": 604800,
      "Refresh": 10800, "Retry": 3600,
      "NegativeTTL": 10800,
      "Type": "SOA" } ],
  "ReturnCode": "NOERROR",
  "AD": true,
  "QuestionSection": { "Qtype": "SOA", "Qname": "bortzmeyer.fr." } }
}
```

An answer with a SOA resource record

```
{ "Query": { "Duration": "0.167317", "Server": "127.0.0.1" },
  "AnswerSection": [
    { "Name": "facebook.com", "TTL": 6666, "Type": "AAAA",
      "Address": "2a03:2880:10:8f01:face:b00c::25" },
    { "Name": "facebook.com", "TTL": 6666, "Type": "AAAA",
      "Address": "2a03:2880:2110:3f01:face:b00c::" },
    { "Name": "facebook.com", "TTL": 6666, "Type": "AAAA",
      "Address": "2a03:2880:10:1f02:face:b00c::25" } ],
  "ReturnCode": "NOERROR" }
```

An answer with several resource records

```
{ "Zone": { "Origin": "isi.edu" },
  [
    { "Type": "SOA", "Name": "@",
      "MasterServerName": "venera",
      "MaintainerName": "action.domains.",
      "Serial": 20 },
```

```
{
  "Type": "NS", Name: "@",
  "Target": "a.isi.edu"},
{
  "Type": "NS", Name: "@",
  "Target": "venera"},
{
  "Type": "NS", Name: "@",
  "Target": "vaxa"},
{
  "Type": "MX", Name: "@",
  "MailExchanger": "venera",
  "Preference": 10},
{
  "Type": "MX", Name: "@",
  "MailExchanger": "vaxa",
  "Preference": 20},
{
  "Type": "A", Name: "a",
  "Address": "26.3.0.103"},
{
  "Type": "A", Name: "venera",
  "Address": "10.1.0.52"},
{
  "Type": "A", Name: "venera",
  "Address": "128.9.0.32"}
]
```

The zone file of RFC 1035

3.6. Open questions

Would it be a good idea to document a formal way to derive member names for the resource record JSON objects? It would allow 1) to document the rationale for the current names 2) to automatically allow representation of new DNS resource records. A possible candidate for such derivation is [I-D.levine-dnsexlang].

Should we define mandatory members for some objects, in order to have something the consumers can rely on? It seems there is a clear consensus to do so, making fields with non-default values mandatory.

In resource records objects, members such as TTL are redundant (since they are actually RRset-wide). Should we have a new level of objects, for RRsets?

Should we use JSON schema ([I-D.zyp-json-schema] and [I-D.fge-json-schema-validation]) to define the profile?

Should we add a normative reference to every RFC describing one of the RR types used here or simply refer to the IANA registry?

Should we have a way to represent unknown RR types, following [RFC3597]?

How binary data should be represented, for types like DNSKEY? Should we use Base64 or is the key value an escaped binary string?

4. Security considerations

These JSON documents are not signed (see [I-D.barnes-jose-use-cases]) and therefore not authenticated, even if the original data was secured with DNSSEC. If transported over an insecure transport, they can be read by a sniffer.

Also, see the security considerations of [RFC4627].

5. References

5.1. 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.
- [RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, July 2006.

5.2. Informative References

- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.
- [RFC3597] Gustafsson, A., "Handling of Unknown DNS Resource Record (RR) Types", RFC 3597, September 2003.
- [I-D.levine-dnsexlang]
Levine, J. and P. Vixie, "An Extension Language for the DNS", draft-levine-dnsexlang-05 (work in progress), December 2012.
- [I-D.barnes-jose-use-cases]
Barnes, R., "Use Cases and Requirements for JSON Object Signing and Encryption (JOSE)", draft-barnes-jose-use-cases-01 (work in progress), October 2012.
- [I-D.mohan-dns-query-xml]

Parthasarathy, M. and P. Vixie, "Representing DNS messages using XML", draft-mohan-dns-query-xml-00 (work in progress), September 2011.

[I-D.zyp-json-schema]

Galiegue, F., Zyp, K., and G. Court, "JSON Schema: core definitions and terminology", draft-zyp-json-schema-04 (work in progress), January 2013.

[I-D.fge-json-schema-validation]

Zyp, K. and G. Court, "JSON Schema: interactive and non interactive validation", draft-fge-json-schema-validation-00 (work in progress), January 2013.

[getdns] Hoffman, P., "Description of the getdns API", February 2013.

[dnspython]

, "dnspython: A DNS toolkit for Python", February 2013.

Author's Address

Stephane Bortzmeyer
AFNIC
Immeuble International
Saint-Quentin-en-Yvelines 78181
France

Phone: +33 1 39 30 83 46
Email: bortzmeyer+ietf@nic.fr
URI: <http://www.afnic.fr/>

template
Internet-Draft
Intended status: Informational
Expires: August 27, 2013

W. Kumari
Google
O. Gudmundsson
Shinkuro Inc.
G. Barwood
February 25, 2013

Automating DNSSEC delegation trust maintenance
draft-kumari-ogud-dnsop-cds-01

Abstract

This document describes a method to allow DNS operators to more easily update DNSSEC Key Signing Keys using DNS as communication channel. This document does not address the initial configuration of trust anchors for a domain.

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 August 27, 2013.

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	2
1.1. Requirements notation	2

2.	Background	3
2.1.	DNS delegations	3
2.2.	DNSSEC key change process	4
3.	CDS Record	4
3.1.	CDS Resource Record Format	5
3.2.	CDS Behavior	5
3.2.1.	Periodic check by Parental Agent	5
3.3.	Usage	6
3.3.1.	Going unsigned	6
4.	IANA Considerations	7
5.	Security Considerations	7
6.	Acknowledgements	7
7.	References	7
7.1.	Normative References	7
7.2.	Informative References	8
	Appendix A. Changes / Author Notes.	8
	Authors' Addresses	9

1. Introduction

When a DNS operator first signs their zone they need to communicate their DS record(s) (or DNSKEY(s)) to their parent through some out of band method to complete the chain of trust. In many cases this is a fairly annoying and manual process. Unfortunately, every time the child rolls their KSK (Key Signing Key) key they have to repeat the process, possibly multiple times. As this is a manual process DNS operators often avoid rolling their keys, as they don't want to have to do go through the annoyance of publishing the new DS records at the parent.

DNSSEC provides data integrity to information published in DNS, thus DNS publication can be used to automate maintenance of delegation information. This document describes a method to automate publication of subsequent DS records, after the initial one has been published.

Readers are expected to be familiar with DNSSEC, including [RFC4033], [RFC4034], [RFC4035], [RFC5011] and [RFC6781].

This document is a compilation of two earlier drafts, draft-barwood-dnsop-ds-publish and draft-wkumari-dnsop-ezkeyroll

This document outlines a technique in which the "parent" (frequently registrar / registry) periodically (or upon request) polls its signed children and automatically publish new DS records. To a large extent the procedures this document follows are in [RFC6781] section 4.1.2

This technique is in some ways similar to RFC 5011 style rollovers, but for sub-domains DS records, instead of trust anchors

1.1. 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. Background

2.1. DNS delegations

DNS operation consists of delegations of authority, for each delegation there are (most of the time) two parties the parent and child.

The parent publishes a NS set that is authoritative for the existence of the delegation but is hint to the contents of the NS record, as well as DS record that expresses what DNSKEY records are to be trusted to sign the DNSKEY RRset in the child. The NS in parent is unsigned as it is hint, the NS bit in the NSEC/NSEC3 record is the proof that the delegation exists. The DS record on the other hand is signed.

The child publishes a signed NS record that as it is authoritative for the contents of the NS set. The child on the other hand can not via current DNS mechanisms express all its desires in which DS records to publish.

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. In this case the parent consists of Registrar and Registry, with different rules on 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 to inject delegation changes into the parent zone. The entity that inserts the changes in the the DNS is called "DNS Publisher"

In many R/R/R cases the Registrar and Registry communicate via EPP[RFC5730] and use the EPP DNSSEC extension [RFC5910].

The "ICANN TLD case" is a common case and we will expand on that here. The registrant registers a domain through a registrar, who then enters information into a database(s), the DNS information (NS, DS and address records) are placed in a database at the registry, and published in the TLD DNS servers. Frequently registrations and subsequent updates take place via web interfaces. When the registrant wants to change NS or DS information it needs to go access the web interface which may take few minutes and many pages to enter the new information. In the ICANN TLD case the Registry operator is by contract not allowed to change the delegation information without the registrar consent, what this means is all changes MUST flow through the registrar. In the context of ICANN TLD's the "Parental Agent" can be assumed to be an registrar, but in other context the "Parental Agent" can be function of the registry.

A further complication is when the DNS Operation is separate from the Registrant. There are two common cases of this, registrar handles the DNS operation and a third party does the DNS operation. In the case of a third party DNS operator the Registrant either needs to relay changes in DNS delegation changes or give the operator access to its registration account. If the Registrar is the DNS operators, life is much easier, as it can inject any delegation changes directly into the Registry data bases. The techniques described below are not needed in the case when Registrar is the DNS operator. To reflect that the Registrant is not always the DNS Operator we will use the word "Child" to describe the party that makes changes in the child zone.

In some cases Registries want to receive DNSKEY records instead of DS records from as the registry calculates the DS records itself. That operating model constrains what the child can do to automate maintenance of DS records, as the child can not publish a DS record for a key that is not in its DNSKEY RRset. Similarly the Child can not control what digest algorithms are used.

2.2. DNSSEC key change process

After an DNS operator first signs its zone, there is a need to interact with the parent via the registration interface to "paste in the zone's DS information". The action of logging in through the registration interface authenticates that the user is authorized to change delegation information published in the parent zone.

Eventually the Child may want to publish a new DS record in the parent, either because they are rolling their keys, or because they want to publish a stand-by key DS record. This involves performing the same process -- logging into the registration interfaces, selecting the domain, finding the link to change DNSSEC information, pasting (or typing) their DS record (often in a non-standard format) and clicking submit. In a real world test, on web interface this took 12 steps and approximately 3 minutes). As humans (especially DNS operators :-)) dislike tedious, repetitive steps they avoid rolling their DNSSEC keys to avoid having to perform this. Furthermore as this is manual process with cut and paste operations mistakes will happen.

3. CDS Record

As the DS record can only be present at the parent some other method is needed to automate the expression of what the parental zone DS records contents ought to be. One possibly is to use flags in DNSKEY record, the SEP bit is an optional bit to indicate that the key is allowed to sign the DNSKEY RRset, and the Parental Agent can calculate DS records based on that. But this fails to meet some operating needs, including the child has no influence what DS digest algorithms are used and DS records can only be published for keys that are in the DNSKEY RRset.

The CDS record can be published in the child zone and gives the child full control of what is published for it in the parental zone.

3.1. CDS Resource Record Format

The wire and presentation format of the CDS ("Child DS") record is identical to the DS record. IANA has allocated RR code 59 for the CDS record.

No special processing is performed by authoritative servers or by resolvers, when serving or resolving. CDS unlike a DS resides in the child zone.

The CDS record MUST be at the zone apex, and MUST be signed with a key that is represented in the current DNSKEY and DS RRset's. If these conditions are not met the CDS record MUST be ignored.

3.2. CDS Behavior

The CDS RRset MAY be used by the Parental Agent to update the DS RRset in the parent zone

Transfer of the contents of the CDS record can be accomplished in a number of ways. A Parental Agent MAY periodically check the child zone to see if the CDS RRset has changed. The child MAY request that the parent check the CDS set via registration interface, or via some other automated mechanism.

If at least one DS and one CDS records exist, the Parental Agent validates and then copies the contents of the CDS RRset and replaces the entire existing DS set with the new one.

The Child MUST make sure that the CDS RRset is at all times can be validated using a DNSKEY that is referenced from the current DS set in the parent. This can be accomplished by making sure that at all times during a KEY rollover there are either two DS records or two DNSKEY records with SEP bit published in the DNS.

When using CDS to publish its key rollover information it is the child's responsibility to monitor the parent for changes to the DS RRset before performing the next action in the key rollover sequence. What this implies is that the child MUST NOT follow a strict time-line but rather strict sequence of steps with time checks.

3.2.1. Periodic check by Parental Agent

In this case the Parental Agent will query each child zone that has a DS RRset, looking for CDS RRset

If present the Parental Agent MUST validate [[RFC4035]] the CDS RRset with a DNSKEY that is represented in the current DS RRset in parent. The Parental Agent should submit a request to the DNS Publisher to publish the contents of the CDS RR(s) as the new a DS record(s) for that zone. The Parental Agent SHOULD log the date and time when of this action including the signature initiation time on the CDS record. The DNS Publisher should log if possible the source of the update, user interface/CDS etc.

The Parental Agent SHOULD NOT check more often than . * TTL on the CDS records.

3.3. Usage

The Parental Agent SHOULD ensure that old versions of the CDS RRset do not overwrite newer versions, which can occur the parent performs the checks too frequently. In that case when there is a delay updating secondary name servers for the child zone. This MAY be accomplished by checking that the signature inception in the RRSIG for CDS is newer and/or the serial number on the child's SOA is greater.

If the CDS RRset does not exist, the parent MUST take no action. Specifically it MUST NOT delete the existing DS RRset.

If the child zone loses the secret key(s) for the zone, and needs to reset the parent DS RRset, this can only be accomplished by an out-of-band mechanism not defined here.

To mitigate situations where a key signing key has been compromised, the Parental Agent MAY take extra security measures, for example informing (by email or other methods) the child zone administrator of the change, or by delaying the acceptance of the new DS RRset for some period of time. However the precise out-of-band measures that a parent zone SHOULD take are outside the scope of this document.

3.3.1. Going unsigned

In theory the child can use the CDS to reflect the parent to remove the DS records. This can be accomplished by publishing CDS record with the following contents:

```
@ IN CDS 0 0 0
```

This is an suggestion and its security implications have not been fully examined but an RFC11 like process should be used before this is accepted. It is important that the Child remain signed until the DS record has been removed from the parent and has timed out from caches.

Note: maybe it is better to register a special DS digest algorithm number for this ?

If the child zone does go unsigned, the Parental Agent should not treat that as intent to go unsigned since that could be an attack. An attacker could spoof unsigned responses to queries from the Parental Agent in an attempt to force a break in the DNSSEC chain.

4. IANA Considerations

IANA has assigned RR Type code 59 for CDS. This was done for an earlier version of this document (draft-barwood-dnsop-ds-publish).

5. Security Considerations

[This needs a more work, suggestions welcome.]

In the event of a compromise of the server generating signatures for a zone, attacker might be able to generate and publish new CDS records. The modified CDS records, will be picked up by this technique and so may allow the attacker to extend the effective time of his attack. This can be dealt with by contacting the parent (possibly via a registrar web interface) and removing any compromised DS keys.

A compromise of the registrar, will not be mitigated by this technique, as the "new registrant" can delete/modify the DS records

While it may be tempting, this SHOULD NOT be used for initial enrollment of keys since there is no way to ensure that the initial key is the correct one. If it is used, strict rules for inclusion of keys like hold down times, challenge data inclusion etc., ought to be used.

The CDS RR type should allow for enhanced security by simplifying process. Since rollover is automated, updating a DS RRset by other means may be regarded as unusual and subject to extra security checks.

6. Acknowledgements

This is by no means the invention of the authors. This idea has been floating around for a long time. This simply documents it for discussion.

We would like to thank: Joe Abley, Roy Arends, Jim Galvin, Cricket Liu, Stephan Lagerholm, Matt Larson, Olaf Kolkman, Suzanne Woolf, Paul Wouters.

There were a large number of other folk with whom we discussed this, apologies for not remembering everyone.

7. References

7.1. Normative References

[IANA.AS_Numbers]

IANA, "Autonomous System (AS) Numbers", , <<http://www.iana.org/assignments/as-numbers>>.

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

7.2. Informative References

- [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.
- [RFC5011] StJohns, M., "Automated Updates of DNS Security (DNSSEC) Trust Anchors", STD 74, RFC 5011, September 2007.
- [RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, August 2009.
- [RFC5734] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Transport over TCP", STD 69, RFC 5734, August 2009.
- [RFC5910] Gould, J. and S. Hollenbeck, "Domain Name System (DNS) Security Extensions Mapping for the Extensible Provisioning Protocol (EPP)", RFC 5910, May 2010.
- [RFC6781] Kolkman, O., Mekking, W. and R. Gieben, "DNSSEC Operational Practices, Version 2", RFC 6781, December 2012.

Appendix A. Changes / Author Notes.

[RFC Editor: Please remove this section before publication]

From - to -1.

- o Removed from section .1: "If a child zone has gone unsigned, i.e. no DNSKEY and no RRSIG in the zone, the parental representative MAY treat that as intent to go unsigned. (NEEDS DISCUSSION)." Added new text at end. -- suggestion by Scott Rose 20/Feb/13.
- o Added some background on the different DNS Delegation operating situations and how they affect interaction of parties. This moved some blocks of text from later sections into here.
- o Number of textual improvements from Stephan Lagerholm
- o Added motivation why CDS is needed in CDS definition section

- o Unified terminology in the document.
- o Much more background

Authors' Addresses

Warren Kumari
Google
1600 Amphitheatre Parkway
Mountain View, CA, 94043
US

Email: warren@kumari.net

Olafur Gudmundsson
Shinkuro Inc.
4922 Fairmont Av, Suite 250
Bethesda, MD 20814
USA

Email: ogud@ogud.com

George Barwood
33 Sandpiper Close
Gloucester, GL2 4LZ
United Kingdom

Email: warren@kumari.net

Domain Name System Operations
Internet-Draft
Intended status: Informational
Expires: August 26, 2013

J. Livingood
C. Griffiths
Comcast
February 22, 2013

Definition and Use of DNSSEC Negative Trust Anchors
draft-livingood-negative-trust-anchors-06

Abstract

DNS Security Extensions (DNSSEC) is now entering widespread deployment. However, domain signing tools and processes are not yet as mature and reliable as is the case for non-DNSSEC-related domain administration tools and processes. One potential technique to mitigate this is to use a Negative Trust Anchor, which is defined in this document.

This document discusses Trust Anchors for DNSSEC and defines a Negative Trust Anchor, which is potentially useful during the transition to ubiquitous DNSSEC deployment. These are configured locally on a particular instance of a validating DNS recursive resolver and can shield end users of such a resolver from the DNSSEC-related authoritative name server operational errors that appear to be somewhat typical during the transition to ubiquitous DNSSEC deployment. Negative Trust Anchors are intended to be temporary, and should not be distributed by IANA or any other organization outside of the administrative boundary of the organization locally implementing a Negative Trust Anchor. Finally, Negative Trust Anchors pertain only to DNSSEC and not to Public Key Infrastructures (PKI) such as X.509.

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 August 26, 2013.

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. Definition of a Negative Trust Anchor	4
3. Limited Time and Scope of Use	4
4. Domain Validation Failures	5
5. End User Reaction	5
6. Switching to a Non-Validating Resolver is Not Recommended . .	6
7. Responsibility for Failures	6
8. Use of a Negative Trust Anchor	7
9. Managing Negative Trust Anchors	9
10. Removal of a Negative Trust Anchor	9
11. Comparison to Other DNS Misconfigurations	10
12. Intentionally Broken Domains	10
13. Other Considerations	10
13.1. Security Considerations	10
13.2. Privacy Considerations	11
13.3. IANA Considerations	11
14. Acknowledgements	11
15. References	12
15.1. Normative References	12
15.2. Informative References	13
Appendix A. Document Change Log	13
Appendix B. Open Issues	14
Authors' Addresses	16

1. Introduction

The Domain Name System (DNS), DNS Security Extensions (DNSSEC), and related operational practices are defined extensively [RFC1034] [RFC1035] [RFC4033] [RFC4034] [RFC4035] [RFC4398] [RFC4509] [RFC6781] [RFC5155].

This document discusses Trust Anchors for DNSSEC and defines a Negative Trust Anchor, which is potentially useful during the transition to ubiquitous DNSSEC deployment. These are configured locally on a particular instance of a validating DNS recursive resolver and can shield end users of such a resolver from the DNSSEC-related authoritative name server operational errors that appear to be somewhat typical during the transition to ubiquitous DNSSEC deployment. Negative Trust Anchors are intended to be temporary, and should not be distributed by IANA or any other organization outside of the administrative boundary of the organization locally implementing a Negative Trust Anchor. Finally, Negative Trust Anchors pertain only to DNSSEC and not to Public Key Infrastructures (PKI) such as X.509. [REFERENCE NECESSARY?]

DNSSEC has now entered widespread deployment. However, domain signing tools and processes are not yet as mature and reliable as is the case for non-DNSSEC-related domain administration tools and processes. As a result, operators of DNS recursive resolvers, such as Internet Service Providers (ISPs), occasionally observe domains incorrectly managing DNSSEC-related resource records. This mismanagement triggers DNSSEC validation failures, and then causes large numbers of end users to be unable to reach a domain. Many end users tend to interpret this as a failure of their DNS servers, and may switch to a non-validating resolver or contact their ISP to complain, rather than seeing this as a failure on the part of the domain they wanted to reach.

In the short-term, one potential way to address this is for DNS operators to use a Negative Trust Anchor to temporarily disable DNSSEC validation for a specific misconfigured domain name. This immediately restores access for end users while that domain's administrators fix their misconfiguration. While DNS operators likely prefer not to use this tool, during the global transition to DNSSEC it seems some tool is needed to reduce the negative impact on such operators.

A Negative Trust Anchor should be considered a transitional and temporary tactic which is not particularly scalable and should not be used in the long-term. Over time, however, the use of Negative Trust Anchors will become less necessary as DNSSEC-related domain administration becomes more resilient.

2. Definition of a Negative Trust Anchor

Trust Anchors are defined in [RFC5914]. A trust anchor should be used by a validating caching resolver as a starting point for building the authentication chain for a signed DNS response. The inverse of this is a Negative Trust Anchor, which creates a stopping point for a caching resolver to end validation of the authentication chain. This Negative Trust Anchor can potentially be placed at any level within the chain of trust and would stop validation at that point in the chain.

3. Limited Time and Scope of Use

As noted in Section 1, the use of Negative Trust Anchors should be temporary. These are key recommendations pertaining to this practice:

1. The general practice of using Negative Trust Anchors should be limited to the transition to widespread deployment of DNSSEC (including signing of domain names and validation in DNS recursive resolvers). Thus, the practice of using Negative Trust Anchors should not be permanent.
2. During this transition phase when Negative Trust Anchors may be useful, the use of a particular Negative Trust Anchor should be temporary and in most cases limited to no more than 1 day. Thus, the use of an individual Negative Trust Anchor should be strictly time limited and very short in duration.
3. So that the use of Negative Trust Anchors remains temporary and useful only during a transition to widespread DNSSEC deployment, the use and distribution of individual Negative Trust Anchors should not be centralized, beyond the borders of one organization's operational unit. Thus, no organization should endeavor to create and centrally distribute Negative Trust Anchors to other organizations as was the case with positive Trust Anchors prior to the signing of the root.
4. As noted in Section 12, organizations that utilize Negative Trust Anchors should not add a Negative Trust Anchor for any intentionally broken domain.
5. As noted in Section 8, use of a Negative Trust Anchor should not be automatic in any way, and must involve investigation by technical personnel trained in the operation of DNS servers.

4. Domain Validation Failures

A domain name can fail validation for two general reasons, a legitimate security failure such as due to an attack or compromise of some sort, or as a result of misconfiguration on the part of an domain administrator. As domains transition to DNSSEC the most likely reason for a validation failure will be due to misconfiguration. Thus, domain administrators should be sure to read [RFC6781] in full. They should also pay special attention to Section 4.2, pertaining to key rollovers, which appears to be the cause of many recent validation failures.

In one recent example [DNSSEC Validation Failure Analysis], a specific domain name failed to validate. An investigation revealed that the domain's administrators performed a Key Signing Key (KSK) rollover by (1) generating a new key and (2) signing the domain with the new key. However, they did not use a double-signing procedure for the KSK and a pre-publish procedure for the ZSK. Double-signing refers to signing a zone with two KSKs and then updating the parent zone with the new DS record so that both keys are valid at the same time. This meant that the domain name was signed with the new KSK, but it was not double-signed with the old KSK. So, the new key was used for signing the zone but the old key was not. As a result, the domain could not be trusted and returned an error when trying to reach the domain. Thus, the domain was in a situation where the DNSSEC chain of trust was broken because the Delegation Signer (DS) record pointed to the old KSK, which was no longer used for signing the zone. (A DS record provides a link in the chain of trust for DNSSEC from the parent zone to the child zone - in this case between TLD and domain name.)

In addition, it is possible that some DNSSEC validation failures could arise due to differences in how different software developers interpret DNSSEC standards and/or how those developers choose to implement support for DNSSEC. For example, it is conceivable that some domain may be DNSSEC signed properly, and Unbound-based DNS recursive resolvers will validate the domain but those using BIND or Nominum's Vantio software may fail to validate a domain.

5. End User Reaction

End users generally do not know what DNSSEC is, nor should they be expected to at the current time (especially absent widespread integration of DNSSEC indicators in end user software such as web browsers). As a result, end users may incorrectly interpret the failure to reach a domain due to DNSSEC-related misconfiguration as their ISP purposely blocking access to the domain or as a performance

failure on the part of their ISP (especially of the ISP's DNS servers). End users may feel less satisfied with their ISP's service, which may make them more likely to switch to a competing ISP. They may also contact their ISP to complain, which of course will incur cost for their ISP. In addition, they may use online tools and sites to complain of this problem, such as via a blog, web forum, or social media site, which may lead to dissatisfaction on the part of other end users or general criticism of an ISP or operator of a DNS recursive resolver.

As end users publicize these failures, others may recommend they switch from security-aware DNS resolvers to resolvers not performing DNSSEC validation. This is a shame since the ISP or other DNS recursive resolver operator is actually doing exactly what they are supposed to do in failing to resolve a domain name, as this is the expected result when a domain can no longer be validated, protecting end users from a potential security threat.

6. Switching to a Non-Validating Resolver is Not Recommended

As noted in Section 5 some people may consider switching to an alternative, non-validating resolver themselves, or may recommend that others do so. But if a domain fails DNSSEC validation and is inaccessible, this could very well be due to a security-related issue. In order to be as safe and secure as possible, end users should not change to DNS servers that do not perform DNSSEC validation as a workaround, and people should not recommend that others do so either. Even if a website in a domain seems to look "normal" and valid, according to the DNSSEC protocol, that domain is not secure. Domains that fail DNSSEC for legitimate reasons may be in control of hackers or there could be other significant security issues with the domain.

Thus, switching to a non-validating resolver to restore access to a domain that fails DNSSEC validation is not a recommended practice, is bad advice to others, is potentially harmful to end user security, and is potentially harmful to DNSSEC adoption.

7. Responsibility for Failures

A domain administrator is solely and completely responsible for managing their domain name(s) and DNS resource records. This includes complete responsibility for the correctness of those resource records, the proper functioning of their DNS authoritative servers, and the correctness of DNS records linking their domain to a top-level domain (TLD) or other higher level domain. Even in cases

where some error may be introduced by a third party, whether that is due to an authoritative server software vendor, software tools vendor, domain name registrar, or other organization, these are all parties that the domain administrator has selected and is responsible for managing successfully.

There are some cases where the domain administrator is different than the domain owner. In those cases, a domain owner has delegated operational responsibility to the domain administrator. So no matter whether a domain owner is also the domain administrator or not, the domain administrator is nevertheless operationally responsible for the proper configuration operation of the domain.

So in the case of a domain name failing to successfully validate, when this is due to a misconfiguration of the domain, that is the sole responsibility of the domain administrator.

Any assistance or mitigation responses undertaken by other parties to mitigate the misconfiguration of a domain name by a domain administrator, especially operators of DNS recursive resolvers, are optional and at the pleasure of those parties.

8. Use of a Negative Trust Anchor

When a domain has been confirmed to fail DNSSEC validation due to a DNSSEC-related misconfiguration, an ISP or other DNS recursive resolver operator may in some cases use a Negative Trust Anchor for a domain or sub-domain. This instructs a DNS recursive resolver to temporarily NOT perform DNSSEC validation for a specific domain name. This immediately restores access to the domain for end users while the domain's administrator corrects the misconfiguration(s).

In the case of a validation failure due to misconfiguration of a TLD or popular domain name (such as a top 100 website), this could make content or services in the affected TLD or domain to be inaccessible for a large number of users. A Negative Trust Anchor can therefore be useful in the short-term when used on a targeted and time-limited basis. It does not and should not involve turning off validation more broadly, and helps during the transition to DNSSEC as organizations that are new to signing their domains are still maturing their DNSSEC operational practices, alleviating end user issues as noted in Section 5 and restoring end user access. However, use of a Negative Trust Anchor should not be automatic in any way, and must involve investigation by technical personnel trained in the operation of DNS servers.

Technical personnel should also confirm that the domain is not

intentionally broken, such as for testing purposes as noted in Section 12. Such an investigation must confirm that a failure is due to misconfiguration, as a similar breakage could have occurred if an attacker gained access to a domain's authoritative servers and modified those records or had the domain pointed to their own rogue authoritative servers. In addition, personnel should make a reasonable attempt to contact a domain for which a Negative Trust Anchor may be used, and preferably prior to implementing it.

Furthermore, a Negative Trust Anchor MUST only be used for a short duration, such as for a day or less. Implementors SHOULD set an end time and date associated with any Negative Trust Anchor. Implementors SHOULD in most cases limit the maximum duration to one day, meaning the Negative Trust Anchor will be removed or invalidated from the point of implementation, plus 86,400 seconds. However, there may be corner cases where a Negative Trust Anchor is needed for a longer period of time. Optimally this time and date is set in a DNS recursive resolver's configuration, though in the short-term this may also be achieved via other systems or supporting processes.

Finally, a Negative Trust Anchor is used only in a specific domain or sub-domain and would not affect validation at other names up the authentication chain. For example, a Negative Trust Anchor for zone1.example.com would affect only names within zone1.example.com, and validation would still be performed on example.com, .com, and the root ("."). In another example, a Negative Trust Anchor for example.com would affect only names within example.com, and validation would still be performed on .com, and the root (".")

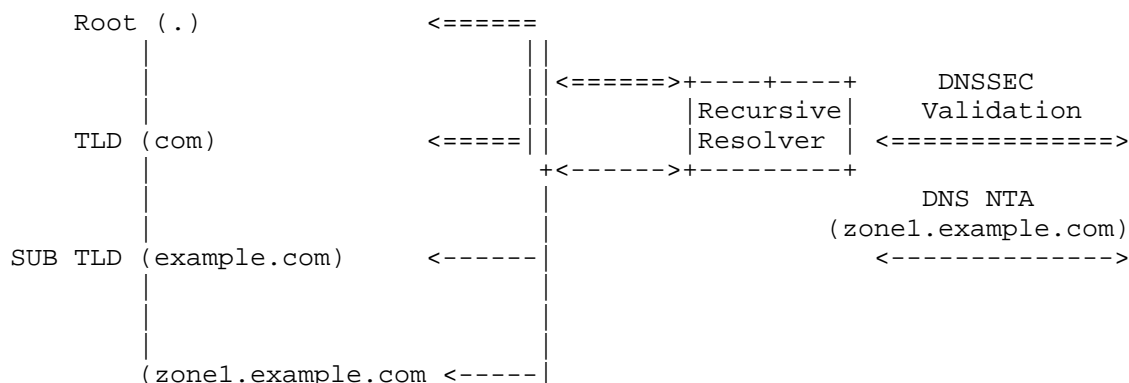


Figure 1: Negative Trust Anchor Diagram

9. Managing Negative Trust Anchors

This tool is unlikely to be and probably should not be used over the long-term since DNSSEC-related domain administration practices will naturally improve over time. In addition, however, continued and frequent use of Negative Trust Anchors is not scalable since it requires investigation by technical personnel and may involve manual processes, resulting in increased operational overhead (and therefore cost).

While Negative Trust Anchors have proven useful during the early stages of DNSSEC adoption, domain owners are ultimately responsible for managing and ensuring their DNS records are configured correctly Section 7.

Most current implementations of DNS validating resolvers currently follow [RFC4033] on defining the implementation of Trust Anchor as either using Delegation Signer (DS), Key Signing Key (KSK), or Zone Signing Key (ZSK). A Negative Trust Anchor should use domain name formatting that signifies where in a delegation a validation process should be stopped.

Different DNS recursive resolvers may have different configuration names for a Negative Trust Anchor. For example, Unbound calls their configuration "domain-insecure" [Unbound Configuration]

10. Removal of a Negative Trust Anchor

As explored in Section 13.1, if a Negative Trust Anchor is still in place after the point in time when the DNS misconfiguration that caused validation to break has been fixed, this could be problematic. It is therefore recommended that implementors should periodically or even continuously attempt to validate the domain in question, for the period of time that the Negative Trust Anchor is in place, until such validation is again successful. (Obviously a Negative Trust Anchor could be removed prior to validation succeeding again, alleviating an implementor of the need to continuing to test validation separate from their normal operations.)

Once validation is again successful, a Negative Trust Anchor should be removed as soon as is reasonably possible. Optimally this is automatic, though it may also be achieved via other systems or supporting processes.

11. Comparison to Other DNS Misconfigurations

As noted in Section 7 domain administrators are ultimately responsible for managing and ensuring their DNS records are configured correctly. ISPs or other DNS recursive resolver operators cannot and should not correct misconfigured A, CNAME, MX, or other resource records of domains for which they are not authoritative. Expecting non-authoritative entities to protect domain administrators from any misconfiguration of resource records is therefore unrealistic and unreasonable, and in the long-term is harmful to the delegated design of the DNS and could lead to extensive operational instability and/or variation.

12. Intentionally Broken Domains

Some domains, such as `dnssec-failed.org`, have been intentionally broken for testing purposes [Measuring DNSSEC Validation of Website Visitors] [Netalyzr]. For example, `dnssec-failed.org` is a DNSSEC-signed domain that is broken. If an end user is querying a validating DNS recursive resolver, then this or other similarly intentionally broken domains should fail to resolve and should result in a SERVFAIL error. If such a domain resolved successfully, then it is a sign that the DNS recursive resolver is not fully validating.

Organizations that utilize Negative Trust Anchors should not add a Negative Trust Anchor for any intentionally broken domain.

Organizations operating an intentionally broken domain may wish to consider adding a TXT record for the domain to the effect of "This domain is purposely DNSSEC broken for testing purposes".

13. Other Considerations

13.1. Security Considerations

End to end DNSSEC validation will be disabled during the time that a Negative Trust Anchor is used. In addition, the Negative Trust Anchor may be in place after the point in time when the DNS misconfiguration that caused validation to break has been fixed. Thus, there may be a gap between when a domain has have been re-secured and when a Negative Trust Anchor is removed. In addition, a Negative Trust Anchor may be put in place by DNS recursive resolver operators without the knowledge of the authoritative domain administrator for a given domain name.

End users of a DNS recursive resolver or other people may wonder why

a domain that fails DNSSEC validation resolves with a supposedly validating resolver. As a result, implementors should consider transparently disclosing those Negative Trust Anchors which are currently in place or were in place in the past, such as on a website [Disclosure Example]. This is particularly important since there is currently no special DNS query response code that could indicate to end users or applications that a Negative Trust Anchor is in place. Such disclosures should optimally include both the data and time that the Negative Trust Anchor was put in place and when it was removed.

13.2. Privacy Considerations

There are no privacy considerations in this document.

13.3. IANA Considerations

There are no IANA considerations in this document.

14. Acknowledgements

Several people made contributions of text to this document and/or played an important role in the development and evolution of this document. This in some cases included performing a detailed review of this document and then providing feedback and constructive criticism for future revisions, or engaging in a healthy debate over the subject of the document. All of this was helpful and therefore the following individuals merit acknowledgement:

- Joe Abley
- John Barnitz
- Tom Creighton
- Marco Davids
- Patrik Falstrom
- Tony Finch
- Chris Ganster
- Olafur Gudmundsson
- Wes Hardaker
- Paul Hoffman

- Shane Kerr
- Murray Kucherawy
- Marc Lampo
- Ted Lemon
- Antoin Verschuren
- Paul Vixie
- Patrik Wallstrom
- Nick Weaver
- Ralf Weber

15. References

15.1. 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.
- [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.
- [RFC4398] Josefsson, S., "Storing Certificates in the Domain Name System (DNS)", RFC 4398, March 2006.
- [RFC4509] Hardaker, W., "Use of SHA-256 in DNSSEC Delegation Signer (DS) Resource Records (RRs)", RFC 4509, May 2006.
- [RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS

Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, March 2008.

[RFC5914] Housley, R., Ashmore, S., and C. Wallace, "Trust Anchor Format", RFC 5914, June 2010.

[RFC6781] Kolkman, O., Mekking, W., and R. Gieben, "DNSSEC Operational Practices, Version 2", RFC 6781, December 2012.

15.2. Informative References

[DNSSEC Validation Failure Analysis]
Barnitz, J., Creighton, T., Ganster, C., Griffiths, C., and J. Livingood, "Analysis of DNSSEC Validation Failure - NASA.GOV", Comcast , January 2012, <http://www.dnssec.comcast.net/DNSSEC_Validation_Failure_NASAGOV_20120118_FINAL.pdf>.

[Disclosure Example]
Comcast, "faa.gov Failing DNSSEC Validation (Fixed)", Comcast , February 2013, <<http://dns.comcast.net/index.php/entry/faa-gov-failing-dnssec-validation-fixed>>.

[Measuring DNSSEC Validation of Website Visitors]
Mens, J., "Is my Web site being used via a DNSSEC-validator?", July 2012, <<http://jpmens.net/2012/07/30/is-my-web-site-being-used-via-dnssec-validator/>>.

[Netalyzr]
Weaver, N., Kreibich, C., Nechaev, B., and V. Paxson, "Implications of Netalyzr's DNS Measurements", Securing and Trusting Internet Names, SATIN 2011 SATIN 2011, April 2011, <<http://conferences.npl.co.uk/satin/presentations/satin2011slides-Weaver.pdf>>.

[Unbound Configuration]
Wijnngaards, W., "Unbound: How to Turn Off DNSSEC", June 2010, <http://unbound.net/documentation/howto_turnoff_dnssec.html>.

Appendix A. Document Change Log

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

-00: First version published as an individual draft.

-01: Fixed minor typos and grammatical nits. Closed all open editorial items.

-02: Simple date change to keep doc from expiring. Substantive updates planned.

-03: Changes to address feedback from Paul Vixie, by adding a new section "Limited Time and Scope of Use". Changes to address issues raised by Antoin Verschuren and Patrik Wallstrom, by adding a new section "Intentionally Broken Domains" and added two related references. Added text to address the need for manual investigation, as suggested by Patrik Falstrom. Added a suggestion on notification as suggested by Marc Lampo. Made several additions and changes suggested by Ralf Weber, Wes Hardaker, Nick Weaver, Tony Finch, Shane Kerr, Joe Abley, Murray Kucherawy, Olafur Gudmundsson.

-04: Moved the section defining a NTA forward, and added new text to the Abstract and Introduction per feedback from Paul Hoffman.

-05: Incorporated feedback from the DNSOP WG list received on 2/17/13 and 2/18/13. This is likely the final version before the IETF 86 draft cutoff date. Updated references to RFC6781 to RFC6781, per March Davids.

-06: Added more OPEN issues to continue tracking WG discussion. No changes in the main document - just expanded issue tracking.

Appendix B. Open Issues

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

Determine whether RFC 2119 language should be used or not when describing things like the duration of a NTA.

Determine whether this is an individual I-D or a DNSOP WG I-D.

Determine whether this is Informational or a BCP.

The DNSOP WG should discuss whether a 1 day limit is reasonable, whether a different time (more or less than 1 day, such as 1 hour or 1 week) should be specified, or whether no time should be specified (just a recommendation that it SHOULD generally be limited to X).

The DNSOP WG should discuss how to assess when critical DNSSEC deployment mass has been achieved so that this is no longer a common practice.

Olafur Gudmundsson has suggested that we may want to consider whether a non validatable RRSIG should be returned or not when a NTA is in place. This was raised in the context of NLnet Labs' DNSSEC-Trigger, which apparently acts like forwarding stub-validator. He said, "The reason for this is if NTA strips signatures the stub-validator thinks it is under attack and may a) go into recursive mode to try to resolve the domain, getting to the right answer the long way. b) Give the wrong error "Missing signatures" instead of the real error. If all the validator does is not to set the AD bit for RRsets at and below the NTA, stub-resolvers (and cascading resolvers) should be happy."

Determine whether an informative reference to X.509 in the Introduction is necessary.

Is it desirable to say that NTAs should not be distributed across organizational boundaries?

Per Warren Kumari on 2/19/2013, add examples to appendix. "it would be very helpful to actually show how this is used, with e.g and example in an Appendix, for -insert favorite resolver here-. The document contains a lot of really useful content about why you might use one, how to minimize damage, etc but (IMO) doesn't do a great job of explaining how to actually do so". Rick Lamb and Joe Abley also agreed on the need for this.

Per Rick Lamb on 2/20/2013, "it might be useful to have section 2 "Definition .." make that clear for slow people like me - that the NTA is not an RR and is more of a configuration. Maybe simply replacing "placed" with "implemented" in section 2? "This NTA can potentially be -placed/implemented- at any level within the chain of trust"

Per Olafur Gudmundsson on 2/18/2013, address fact that ALL authoritative name servers must be working. "section 10 you talk about possible early removal the NTA when validation succeeds but there may be instances where validation succeeds when using a sub-set of the authoritative servers thus NTA should only be removed if all servers are providing "good" signatures."

Per Olafur Gudmundsson on 2/18/2013, "Furthermore what to do if some names work but others do not, for example I remember a case where the records at the apex worked but all names below the apex were signed by a key not in the DNSKEY RRset, thus it is possible that either human or automated checks may assume there is no problem when there actually is one. What this is bringing to my mind is maybe you want a new section with guidelines on how to test for failures and in what cases failure justifies NTA and what tests MUST pass before

preemptive removal of an NTA."

Per Olafur Gudmundsson on 2/18/2013, "Also should there be guidance that removal of NTA should include cleaning the caches of all RRsets below the name?"

Authors' Addresses

Jason Livingood
Comcast Cable Communications
One Comcast Center
1701 John F. Kennedy Boulevard
Philadelphia, PA 19103
US

Email: jason_livingood@cable.comcast.com
URI: <http://www.comcast.com>

Chris Griffiths
Comcast Cable Communications
One Comcast Center
1701 John F. Kennedy Boulevard
Philadelphia, PA 19103
US

Email: chris_griffiths@cable.comcast.com
URI: <http://www.comcast.com>

Network Working Group
Internet-Draft
Updates: 6304 (if approved)
Intended status: Informational
Expires: August 29, 2013

W. Kumari
Google
W. Sotomayor
NRC-CNRC
J. Abley
ICANN
R. Bellis
Nominet UK
February 25, 2013

Omniscient AS112 Servers
draft-wkumari-dnsop-omniscient-as112-02

Abstract

The AS112 Project loosely coordinates Domain Name System (DNS) servers to which DNS zones corresponding to private use addresses are delegated. Queries for names within those zones have no useful responses in a global context. The purpose of this project is to reduce the load of such junk queries on the authoritative name servers that would otherwise receive them, and instead direct the load to name servers operated within the AS112 project.

Adding and dropping zones from the AS112 servers is difficult, due to the loosely-coordinated nature of the project. This document proposes a mechanism by which AS112 name servers could answer authoritatively for all possible zones. This eliminates the add/drop problem, changing it to a matter of delegation within the DNS and requiring no operational changes on the servers themselves.

This document updates RFC 6304.

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 August 29, 2013.

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	4
2. Terminology	4
3. Protocol Considerations	5
4. Operational Considerations	6
5. Addressing Considerations	7
6. Updates to RFC 6304	7
6.1. Changes to Section 3.4, Routing Software	7
6.2. Changes to Section 3.5, DNS Software	9
6.3. Changes to Section 3.6, Testing a Newly Installed Node	9
7. IANA Considerations	9
8. Security Considerations	9
9. Acknowledgements	10
10. References	10
10.1. Normative References	10
10.2. Informative References	10
Appendix A. Implementation / "Running Code"	10
Appendix B. Document Notes	11
B.1. Venue	11
B.2. Textual Substitutions	11
B.3. Open Questions	11
B.4. Change History	11
B.4.1. draft-wkumari-dnsop-omniscient-as112-00	12
B.4.2. draft-wkumari-omniscient-as112-00	13
Authors' Addresses	13

1. Introduction

The AS112 Project loosely coordinates Domain Name System (DNS) servers [RFC1034] to which DNS zones corresponding to private use addresses are delegated. Queries for names within those zones have no useful responses in a global context. The purpose of the project is to reduce the load of such junk queries on the authoritative name servers that would otherwise receive them, directing the load instead to name servers operated within the AS112 project.

To date, AS112 nameservers have been used exclusively for names corresponding to the reverse mapping for private-use IPv4 addresses. A description of current advice for AS112 operators, including motivations and guidance for technical deployment and operations can be found in [RFC6304].

Other DNS domains have analogously local significance. Examples corresponding to the reverse-mapping of special-use IPv4 and IPv6 addresses can be found in [RFC6303].

It is to be expected that new domains will be identified from time to time that fit the use pattern for which delegation to AS112 servers might be desirable. There is currently no mechanism by which particular zones can be reliably added to or dropped from AS112 servers, however. This is principally a consequence of the loosely-coordinated nature of the project, coupled with a desire to avoid lame delegations which might have unforeseen operational consequences.

This document proposes a mechanism by which AS112 servers could provide consistent, reliable negative responses for all DNS queries, eliminating the operational requirement to add or drop particular zones from all AS112 servers.

2. Terminology

An "Existing AS112 Server" is a DNS name server configured according to the guidance provided in [RFC6304] and listening on the IPv4 addresses 192.175.48.1 (PRISONER.IANA.ORG), 192.175.48.6 (BLACKHOLE-1.IANA.ORG) and 192.175.48.42 (BLACKHOLE-2.IANA.ORG).

An "Omniscient AS112 Server" is a DNS nameserver configured according to the guidance provided in [RFC6304], as extended by this document. Such servers listen on the same addresses as Existing AS112 Servers, but also additional addresses as described in Section 5.

Where discussions apply equally to Existing AS112 Servers and

Omniscient AS112 Servers, the unqualified phrase "AS112 Server" is used.

An "AS112 Zone" is a DNS zone which has been delegated to an AS112 Server.

An "Existing AS112 Zone" is an AS112 Zone which has been delegated to an existing AS112 Server.

3. Protocol Considerations

Familiarity with [RFC1034] and [RFC1035] is assumed.

In order to safely cache the response, DNS implementations require the closest-enclosing SOA to be returned. An omniscient AS112 server (which is not configured with a specific list of zones, and hence zone cuts) cannot necessarily know where that is. Removing labels and guessing (whether to the extreme case of removing all labels, or returning one, or anything in between) cannot be guaranteed to be appropriate, since the answers might clash with authentic answers already present in client caches. A client that has followed a referral to an omniscient AS112 server is guaranteed not to have a cached SOA that matches the QNAME, however, so Omniscient AS112 servers use the QNAME as the SOA and owner name.

Please see Appendix A for information on an implementation ("running code") that does this.

AS112 Servers do not respond to AXFR (QTYPE=252) or IXFR (QTYPE=251) requests.

A TYPE=6 (SOA) resource record for Omniscient AS112 servers contains:

- o MNAME = "a.as112.net."
- o RNAME = "hostmaster.as112.net."
- o SERIAL = 1
- o REFRESH = 604800 (7 days)
- o RETRY = 2592000 (30 days)
- o EXPIRE = 604800 (7 days)
- o MINIMUM = 604800 (7 days, negative caching TTL)

For all queries with QTYPE=2 (NS) an AS112 Server responds with an authoritative (AA=1) answer with NXDomain (RCODE=3), the owner name copied from the QNAME and two resource records of TYPE=2 (NS), one containing "B.AS112.NET." and the containing "C.AS112.NET.".

For all queries with QTYPE=6 (SOA) an AS112 Server responds with an authoritative (AA=1) answer with NXDomain (RCODE=3), the owner name

copied from the QNAME and one (ANCOUNT=1) resource record of TYPE=6 (SOA).

For all queries with QTYPE= 255 (*, also known as ANY) an AS112 Server responds with an authoritative (AA=1) answer with NXDomain (RCODE=3) the owner name copied from the QNAME and three (ANCOUNT=3) resource records, one containing the SOA (as described above), and two containing NS (also as described above).

For all other queries an AS112 Server responds with an authoritative (AA=1) NoError (RCODE=0) with the owner name copied from the QNAME in the request and no answers (ANCOUNT=0). The resource record of TYPE=6 (SOA) (as described above) should be returned in the authority section. The presence of the SOA is to allow the negative cache TTL to be set(see [RFC2308]).

4. Operational Considerations

Existing AS112 Servers address the protocol considerations described in Section 3 by serving each existing AS112 Zone explicitly. In each case the zone contents are identical, containing only required apex SOA and NS records. Adding or dropping a delegation for an Existing AS112 Zone requires coordination amongst all deployed Existing AS112 Server operators.

There is no practical expectation that AS112 Server operators coordinate the configuration of their infrastructure or even make their existence known in any systematic way. Delegation of new zones to Existing AS112 Servers is hence problematic; there is an expectation that such delegations would be lame for a significant client population. Since the predictable behaviour of AS112 Servers from clients is desirable, and it is possible that significant variation would have operational consequences, no new zones should be delegated to existing AS112 Servers.

Omniscient AS112 Servers generate a response (as described in Section 3 (Section 3)) as though they are authoritative for everything ("."). Adding or dropping a delegation for an AS112 Zone therefore imposes no operational requirements on Omniscient AS112 Server operators.

Delegation of new AS112 Zones should only be made to Omniscient AS112 Servers. Omniscient AS112 Servers, therefore, must listen on additional addresses to those used by existing AS112 Servers. Addressing is discussed in Section 5.

By ensuring that Omniscient AS112 Servers listen on Existing AS112

Servers' addresses as well as the new addresses specified in Section 5 a smooth migration is possible, allowing Existing AS112 Servers to be reconfigured as Omniscient AS112 Servers. Omniscient AS112 Servers are therefore a superset of AS112 Servers.

5. Addressing Considerations

Omniscient AS112 Servers listen on the following addresses:

- o IPv4-TBA1 (A.AS112.NET)
- o IPv6-TBA1 (A.AS112.NET)
- o IPv4-TBA2 (B.AS112.NET)
- o IPv6-TBA2 (B.AS112.NET)
- o IPv4-TBA3 (C.AS112.NET)
- o IPv6-TBA3 (C.AS112.NET)

IPv4-TBA1, IPv4-TBA2 and IPv4-TBA3 are covered by a single IPv4 prefix, IPv4-PREFIX-TBA. Similarly, IPv6-TBA1, IPv6-TBA2 and IPv6-TBA3 are covered by a single IPv6 prefix, IPv6-PREFIX-TBA.

The addresses specified for Omniscient AS112 Servers are deliberately different from those assigned to Existing AS112 Servers for reasons discussed in Section 4.

6. Updates to RFC 6304

6.1. Changes to Section 3.4, Routing Software

Omniscient AS112 Nodes with IPv4 connectivity should originate the IPv4 service prefix associated with Existing AS112 Nodes, 192.175.48.0/24, and also the IPv4 service prefix associated with Omniscient AS112 Nodes, IPv4-PREFIX.

Omniscient AS112 Nodes with IPv6 connectivity should originate the IPv6 service prefix IPv6-PREFIX-TBA.

Applying this direction to the "bgpd.conf" file included as an example in this section results in the configuration shown in Figure 1.

```
! bgpd.conf
!
hostname as112-bgpd
password <something>
enable password <supersomething>
!
```

```
! Note that all AS112 nodes use the local Autonomous System
! Number 112, and originate IPv4 and IPv6 prefixes (where IPv4
! and IPv6 connectivity is available) as follows:
!
!   IPv4:   192.175.48.0/24
!           IPv4-PREFIX-TBA
!
!   IPv6:   IPv6-PREFIX-TBA
!
! All other addresses shown below are illustrative, and
! actual numbers will depend on local circumstances.
!
router bgp 112
  bgp router-id 203.0.113.1
  !
  address-family ipv4
    network 192.175.48.0
    neighbor 192.0.2.1 remote-as 64496
    neighbor 192.0.2.1 next-hop-self
    neighbor 192.0.2.1 prefix-list AS112-v4 out
    neighbor 192.0.2.1 filter-list 1 out
    neighbor 192.0.2.2 remote-as 64497
    neighbor 192.0.2.2 next-hop-self
    neighbor 192.0.2.2 prefix-list AS112-v4 out
    neighbor 192.0.2.2 filter-list 1 out
    network 192.175.48.0/24
    network IPv4-PREFIX-TBA
  !
  address-family ipv6 unicast
    neighbor 2001:db8::1 remote-as 64496
    neighbor 2001:db8::1 next-hop-self
    neighbor 2001:db8::1 prefix-list AS112-v6 out
    neighbor 2001:db8::1 filter-list 1 out
    neighbor 2001:db8::2 remote-as 64497
    neighbor 2001:db8::2 next-hop-self
    neighbor 2001:db8::2 prefix-list AS112-v6 out
    neighbor 2001:db8::2 filter-list 1 out
    network IPv6-PREFIX-TBA
  !
  ip prefix-list AS112-v4 permit 192.175.48.0/24
  ip prefix-list AS112-v4 permit IPv4-PREFIX-TBA
  !
  ipv6 prefix-list AS112-v6 permit IPv6-PREFIX-TBA
  !
  ip as-path access-list 1 permit ^$
```

Figure 1

6.2. Changes to Section 3.5, DNS Software

Omniscient AS112 Servers should be configured to listen on the addresses Pv6-TBA1, IPv6-TBA, IPv6-TBA3, IPv4-TBA1, IPv4-TBA2 and IPv4-TBA3 in addition to the addresses used for Existing AS112 Servers.

Omniscient AS112 Servers generate an answer as described in Section 3 instead of explicitly serving the zones specified in [RFC6304].

As ISC BIND [BIND] does not provide the required functionality a custom nameserver implementation needs to be deployed, and so the example "named.conf" file in this section can be disregarded.

6.3. Changes to Section 3.6, Testing a Newly Installed Node

Testing should include all configured service addresses for an Omniscient AS112 Server (IPv4 or IPv6 or both, as appropriate). Note that the IPv4 service addresses include those described in [RFC6304] for Existing AS112 Servers.

7. IANA Considerations

This document describes infrastructure which could be used in the future to direct the IANA to delegate or redelegate infrastructure zones under its administrative control.

However, this document makes no request of the IANA.

8. Security Considerations

The contents of the Security Considerations section of [RFC6304] should be reviewed, since that discussion is pertinent to the operation of Omniscient AS112 Servers as well as Existing AS112 Servers.

The deployment of Omniscient AS112 Servers enables new delegations to AS112 Servers.

Queries received by an AS112 Server might reveal operational data for which there is an expectation of privacy. For example, leaked queries for an organisation's internal DNS names which are sent to an AS112 Server might reveal the existence of those names to the AS112 Server operator. The delegation of new zones to AS112 Servers has the potential to increase opportunities for such unintentional information leakage.

The delegation of new zones to AS112 Servers has the potential to increase the traffic received by those servers. AS112 Server operators are encouraged to monitor traffic levels, and to take appropriate steps if traffic levels threaten the stability of their networks.

9. Acknowledgements

The authors thank and acknowledge the contributions of Dr Paul Vixie, Bill Manning, George Michaelson, Mark Andrews, Shane Kerr, Brian Dickson, S. Moonesamy, Chris Thompson, Nick Hilliard and all the folk on the AS112 Project mailing lists in the preparation of this document.

10. References

10.1. 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.
- [RFC2308] Andrews, M., "Negative Caching of DNS Queries (DNS NCACHE)", RFC 2308, March 1998.
- [RFC6304] Abley, J. and W. Maton, "AS112 Nameserver Operations", RFC 6304, July 2011.

10.2. Informative References

- [BIND] Nominet UK, "Internet Systems Consortium, "BIND"", <<http://www.isc.org/>>.
- [RFC6303] Andrews, M., "Locally Served DNS Zones", BCP 163, RFC 6303, July 2011.
- [evldns] Bellis, R., "evldns", <<http://code.google.com/p/evldns/>>.

Appendix A. Implementation / "Running Code"

The "evldns" [evldns] library (written by Ray Bellis, Nominet UK) includes an Omniscient AS112 Server implementation in the file "oas112d.c"

Appendix B. Document Notes

This section (and sub-sections) contain information useful for development and review of this document, and should be removed prior to publication.

B.1. Venue

This document is an individual submission, and is not the product of an IETF working group. However, a suitable venue for discussion is the dnsop working group mailing list.

B.2. Textual Substitutions

The strings "IPv4-TBA1", "IPv4-TBA2" and "IPv4-TBA3" should be replaced in this document should be replaced with IPv4 addresses assigned for the purpose described. The covering IPv4 prefix for all three addresses should replace the string "IPv4-PREFIX-TBA".

Similarly, the strings "IPv6-TBA1", "IPv6-TBA2", "IPv6-TBA3" and "IPv6-PREFIX-TBA" should be substituted in the text with assigned production values.

B.3. Open Questions

1. Where to get IPv4 and IPv6 assignments from? There has already been an assignment to DNS-OARC by ARIN for v6 service for AS112 servers.

B.4. Change History

Template:

- o Initial draft
- o Initial draft, circulated privately, not submitted.

-00:

- o Rewrote much of the document (especially Section 3 to explain how (and why) responses should be generated.
- o Updated "Updates to RFC 6304" section to explain the BIND does not currently implement this, and so named.conf, etc should be ignored.
- o Removed example "empty" zone.
- o Changed the addressing bit at the suggestion of SM.

-01:

- o Document title changed to include the dnsop keyword, so that IETF document automation can send courtesy notifications of document actions to the dnsop working group.
- o Abstract and introduction expanded.
- o RFC2119 requirements notation removed, since this is an informational document and any normative language would be toothless.
- o Discussion broken out into Protocol Considerations, Operational Considerations and Addressing Considerations.
- o Reverted to the custom software / synthesized answers.
- o Added in the Ray Bellis evldns stuff.

-01 to -02:

- o s/NoError/NXDomain/ -- Suggestion from Paul Vixie (and others). "Nxd says there is no such name, no matter what the type was, and there are no children. No data/noerror says there are either other types or children or both. We know what the truth must be and we know which implications we want the requestor to follow. Right?" -- Paul.
- o Need to retest with empty root zone, and "minimal responses". Initial test didn't seem to suppress the 'Negative Answers from Authoritative Servers' (rfc2308)
- o Removed the ''Editor note: [NoError was chosen instead of NXDOMAIN because we did not think that we could reasonably return an SOA RR which clearly indicates that the QNAME does exist, and also return an NXDOMAIN.]" as we are now using NXDomain :-P
- o This version submitted by Warren, with no real discussion with co-authors. Trying to squeeze things under the -01 cutoff.

B.4.1. draft-wkumari-dnsop-omniscient-as112-00

Document title changed to include the dnsop keyword, so that IETF document automation can send courtesy notifications of document actions to the dnsop working group.

Abstract and introduction expanded.

RFC2119 requirements notation removed, since this is an informational document and any normative language would be toothless.

Discussion broken out into Protocol Considerations, Operational Considerations and Addressing Considerations.

Detailed updates to [RFC6304] added.

B.4.2. draft-wkumari-omniscient-as112-00

Authors' Addresses

Warren Kumari
Google
1600 Ampitheatre Parkway
Mountain View, CA 94043
USA

Email: warren@kumari.net

William F. Maton Sotomayor
National Research Council of Canada
1200 Montreal Road
Ottawa, ON K1A 0R6
Canada

Phone: +1 613 993 0880
Email: wfms@ryouko.imsb.nrc.ca

Joe Abley
ICANN
12025 Waterfront Drive, Suite 300
Los Angeles, CA 90094-2536
USA

Phone: +1 519 670 9327
Email: joe.abley@icann.org

Ray Bellis
Nominet UK
Edmund Halley Road
Oxford, OX4 4DQ
United Kingdom

Phone: +44 1865 332211
Email: ray.bellis@nominet.org.uk

