Abstract

Two DNS names may have the same DNS administrative boundaries for some services. This document adds the function of lookup of domain name administrative boundary to domain name system, which describes a new method for using dbound resource record for judging domain name administrative boundaries.

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 18, 2016.

Copyright Notice

Copyright (c) 2015 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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction ............................................. 2
2. Terminology ............................................. 3
3. Design Consideration ................................... 3
4. Framework .............................................. 4
5. Application Algorithm for Dbound Query ............... 6
6. Discussion .............................................. 7
   6.1. New design of dbound for Discussion ............. 8
7. IANA Considerations ..................................... 9
8. Security Considerations ................................. 10
9. Acknowledgements ....................................... 10
10. Change History ......................................... 10
10.1. draft-yao-dbound-dns-solution: Version 00 ........ 10
11. References ........................................... 11
   11.1. Normative References .............................. 11
   11.2. Informative References ............................ 11
Authors’ Addresses ......................................... 12

1. Introduction

Two DNS [RFC1034] [RFC1035] names may have the same administrative boundaries to some services. If they share the same DNS administrative boundaries, we regard that they have a relationship. Otherwise they have not a relationship. This document describes an method for using dbound resource record for judging domain name administrative boundaries.

The drafts [Boundaries-Problem] [Boundaries-Concepts] list many use cases where some applications may use domain name administrative boundaries. With the growth of Internet, there should have more Internet applications which will use domain name administrative boundaries technology.
Some applications use the centralized service via the unified platform to get the policy of the domain names. With the policy sharing feature, the administrator of domain names can give applications access to the centralized policy platform to fetch the information. This platform is in the unified control of some administrators. By using this "Policy Delegates function" mechanism, the administrator of domain names delegates his policy power to the centralized platform. With the permission, the administrator of the centralized platform can create, edit, and delete the policy information.

Some domain names may have a time based boundary with other domain names. After the time expiration, the boundary relationship will expire. The administrators may set some domain names to share the same boundary policy within the fixed time. For example, after 1 July, 2017, the relationship of sharing the same administrative boundary between domain names A and B for SSL and TLS [RFC6125] services will expire.

There have many kind of services. Two domain names may share the policy boundary only if they use the same service type or some specific service types. For example, only for http service, example.com and example.net enjoys the same policy boundary.

All the issues above should be considered for the design of administrative boundary technology.

2. Terminology

The basic key words such as "MUST", "MUST NOT", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "MAY", and "MAY NOT" are to be interpreted as described in [RFC2119].

The basic DNS terms used in this specification are defined in the documents [RFC1034] and [RFC1035].

3. Design Consideration

In order to be favor of the future services, the design should be scalable and extensible. While services will be running in different platform, it should have the feature of easily switching to the different environment. Since some current mechanisms have supported some functions of domain name administrative boundary, it is good to be compatible with them. The new mechanism should have a good design to have all the components of the proposed system worked together.

The following key factors are considered for designing the protocol in this document:
portability:
A protocol can be considered to be portable if it can work with many different systems or platforms. Portability is the usability of the same protocol in different environments. When the protocol with the same functionality is produced for different environments, portability is the key issue for easy deployment. From this point of view, the DNS based solution is perfect for portability since DNS can work with many different systems. DNS is one of the important parts of these systems.

scability:
A protocol can be considered to scale if it is suitably efficient and practical when applied to large situations. Scalability is the ability of a protocol to handle a growing amount of use cases in a capable manner. The problem statement [Boundaries-Problem][Boundaries-Concepts] has listed some important use cases. In the future, more use cases will appear. The solution should consider this issue and have the scalability.

Compatibility:
The protocol can be considered to be compatible if it can work with some current technology such as a legacy system. In this document, the current system or technology mainly refers to the Public Suffix List technology (PSL). Some good features of the current mechanisms which have been deployed for many years should still be considered in the new design of framework since it has been running stably for years. For an example, the Public Suffix List (PSL) [PSL] works well for some use cases.

Complexity:
A protocol can be considered to be complexity when it has many parts where those parts interact with each other in multiple ways. If a system satisfies with the requirements above, it should have many components. These components should work together, and can deal with many situations in different way.

The design philosophy of this solution is that it tries to meet the above requirements and satisfy the current use cases and possible future use cases.

4. Framework

This section presents a mechanism to lookup of the administrative boundary between two domains. The mechanism defines a new resource record type (RRTYPE) to satisfy the requirements specified in the previous section. The RDATA for an Dbound RR consists of a 1 octet Flag field, a 1 octet Relation field, a Service Type field, a 4 octet
Dbound Expiration field, a 4 octet Dbound Inception field, and a Target Names field.

```
  1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Flag     |   Relation    |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Dbound    Expiration                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Dbound    Inception                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Target Names                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1. The structure of RDATA of Dbound resource record

**Flag**
The Flag field identifies the usage of Target Names field. 0 means that Target Names field will specify the service location such as URL to look for the policy. 1 means that Target Names field will specify the domain names which share the same administrative boundary with the owner name.

**Relation**
In [Boundaries-Concepts] document, there are two types of domain name relationships identified: ancestry and policy. The Relation field identifies the property of the relationship. 0 means that it is ancestry; 1 means that it is policy. In the future, more values may be defined for this field. This field is reserved for future use.

**Service Type**
Service Type identifies which service the domain name and Target Names are trying to use for the shared administrative boundary. This field include a character-string that specifies the service type. The service type should be registered under the control of IANA.

**Dbound Expiration and Inception**
The Dbound Expiration and Inception fields specify a validity period for the dbound partnership. The application MUST NOT use the partnership for whatever purposes prior to the inception date and MUST NOT use it for whatever purposes after the expiration date.
Dbound Expiration and Inception field values specify a date and time in the form of a 32-bit unsigned number of seconds elapsed since 1 January 1970 00:00:00 UTC, ignoring leap seconds, in network byte order. The Dbound Expiration field value should be larger than the Dbound Inception field value. If the Dbound Inception field value is zero, it means that the application can use the domain name partnership immediately unless the starting date is larger than the expiration date. If every bit of the Dbound Expiration field value is set to be ‘1’, it means that the application can use the domain name partnership forever if the starting date is larger than the inception date.

Target Names
There are two kinds of Target Names, one is domain name; the other is service location. If the Target Names are domain names, the Target Name field identifies the Target domain names which may share the same DNS administrative boundary with the owner name. There may have more than one Target domain names which will be separated by comma (,). For example, domain name A may share the administrative boundary with B, C and D.

If the Target name is a specific service location, it should be a string which specifies where the service is located. For an example, it may be a URL [publicsuffix.org] such as that http://mxr.mozilla.org/mozilla-central/source/netwerk/dns/effective_tld_names.dat?raw=1.

5. Application Algorithm for Dbound Query

There are two cases where application can determine whether domain names A and B share the same administrative boundaries.

Case 1: If A and B’s flag in the dbound record are 0, application should confirm the following requirement before that applications go to the service location to confirm that A and B SHOULD share the same administrative boundary.
1). A points to the same service location in A’s dbound record.
2). B points to the same service location in B’s dbound record.
3). A and B’s value of dbound expiration SHOULD be bigger than current time value
4). A and B SHOULD share the same service type.

Case 2: If flag is 1, application should check the following things before confirming that A and B SHOULD share the same administrative boundary.
1). A points to B in A’s dbound record.
2). B points to A in B’s dbound record.
3). A and B’s value of dbound expiration SHOULD be bigger than current time value
4). A and B SHOULD share the same service type.

Algorithm:
When the application needs to know whether two names A and B share the same administrative boundary, it needs to do the following steps to confirm it. If A does not know who may share the same administrative boundary with itself, it finds its dbound record first, check it to find the possible domain name and regard it as B. Step 1, the application sends the query of A for dbound record to the DNS servers, and analyzes the response. If the application gets the dbound RR, it checks this RR. If flag is 0, go to step 2; If flag is 1, check whether A points to B in A’s dbound record. If yes, go to step 2. Otherwise go to step 6
Step 2, the application sends the query of B for dbound record to the DNS servers, and analyzes the response. If the application gets the dbound RR, it checks this RR. If flag is 0, go to step 3; If flag is 1, check whether B points to A in B’s dbound record. If yes, go to step 3. Otherwise go to step 6
Step 3, If flag is 0, it means that relationship between two names are defined in the specific service location. The dbound-aware application should check the specific location for looking for the relationship. For an example, the administrator of .com and .net may set PSL as the service location; those who hope to check the relationship between .com and .net should go to PSL for the answer. If flag is 0, compare the A and B’s dbound RR’s Target Names field values. If values are same, and the values specify the URI, go to step 4. If flag is 1, compare the A and B’s dbound RR’s type field values. If values are same, go to step 5
Step 4, If flag is 0, check A or B’s dbound RR’s Target Names field value and go to the centralized service location specified in the URI gotten from Target Names field value. This step SHOULD be specified in other documents which specifies the URI information. The URI information should be registered in IANA registry if the URI need to be controlled globally.
Step 5, If flag is 1, check A and B’s dbound RR’s Dbound Expiration and Inception field value.
Set applications’s policy inception value= max.(A’s Dbound Inception field value, B’s Dbound Inception field value)
Set applications’s policy Expiration value = min.(A’s Dbound Expiration field value, B’s Dbound Expiration field value)
Exit
Step 6, Exit and display some error information

6. Discussion

This section will be changed if it is published.

It is an initial design. It is open to change and will follow the WG’s decision
6.1. New design of dbound for Discussion

After following the discussion in the WG discussion list, the following possible alternative mechanism is proposed:

```
  1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Flag     |   Reserved    |   Reserved    |               /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Anchor Name / Name Collection                /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2. The structure of RDATA of Dbound resource record

If flag=0, the Anchor Name / Name Collection is the anchor name, the anchor name will be the string of PSL. Through it, the DNS administrators can configure the relationship between the owner name and PSL. Those which point to the PSL will share the same DNS administrative boundaries;

If flag=1, the Anchor Name / Name Collection is the anchor name, it means that dbound record is to try to build a connection between the owner name and the anchor name which is a FQDN. Through it, the DNS administrators can configure the relationship between the owner name and the anchor name. Those which share the same anchor name will share the same DNS administrative boundaries;

If flag=2, the Anchor Name / Name Collection is the name collection, the Name Collection will be a collection of names which are supposed to share the same DNS boundaries under the same anchor name and will be separated by comma(,). The owner name is some names’ anchor name in other dbound RR. Through it, the application can learn how many names share the same DNS boundaries under the owner name (some names’ anchor name in other dbound RRs)

For examples:

EXAMPLE 1, if a.example and b.example want to share the same DNS administrative boundaries, it can configure the following RRs:

- a.example dbound 1 c.example
- b.example dbound 1 c.example
- c.example dbound 2 a.example,b.example

or the anchor name can also be one of the names who share the same dns administrative boundaries:

- a.example dbound 1 b.example
- b.example dbound 1 b.example
b.example dbound 2 a.example, b.example

**USAGE:** if the application wants to check whether a.example and b.example share the same dns boundaries, it find a.example and b.example share the same anchor under the flag’s value of 1 under the RRs above, and verify that a.example and b.example share the same dns boundaries.

if the application wants to check which domain names share the same DNS boundaries with a.example, it find a.example and b.example are supposed to have the same DNS boundaries under the flag’s value of 2, and verify that a.example and b.example share the same dns boundaries through checking a.example and b.example sharing the same anchor under the flag’s value of 1.

**EXAMPLE 2,** if a.example and b.example want to share the same DNS administrative boundaries under PSL, it can configure the following RRs:

a.example dbound 0 http://mxr.mozilla.org/mozilla-central/source/netwerk/dns/effective_tld_names.dat?raw=1
b.example dbound 0 http://mxr.mozilla.org/mozilla-central/source/netwerk/dns/effective_tld_names.dat?raw=1

**USAGE:** if the application wants to check whether a.example and b.example share the same dns boundaries, it find a.example and b.example share the same anchor under the flag’s value of 0, and verify that a.example and b.example share the same dns boundaries via the PSL link.

**ADVANTAGES:** This new mechanism builds a relationship through the anchor name (middleman) to avoid to construct too many pairwise relationship. It will help to reduce the RRs configuration and checking when there are many domain names which are supposed to share the same DNS boundaries.

7. **IANA Considerations**

The IANA should allocate the new DNS type for DBOUND.
The IANA should create the "Dbound service type" registry of the "Domain Name System (DNS) Parameters" registry, and add some values according to the following data:
<table>
<thead>
<tr>
<th>Service Type</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>http state management</td>
<td>[RFC****] [RFC6265]</td>
</tr>
<tr>
<td>DMARC</td>
<td>Email verification</td>
<td>[RFC****]</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Socket Layer (SSL)</td>
<td>[RFC****]</td>
</tr>
</tbody>
</table>

Figure 2. The service Type of Dbound resource record

The IANA should create the "Dbound Public Service" registry of the "Domain Name System (DNS) Parameters" registry, based on the following format:

<table>
<thead>
<tr>
<th>public service</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>****</td>
<td>*******</td>
<td>[RFC****]</td>
</tr>
</tbody>
</table>

Figure 3. The public service of Dbound resource record

8. Security Considerations

For the centralized public service, its security depends on the DNS and the service which supports the lookup of the policy of the domain name boundary. If there is a security problem for lookup service, the domain boundary will fall in the security issues.

9. Acknowledgements

TBD

10. Change History

RFC Editor: Please remove this section.

10.1. draft-yao-dbound-dns-solution: Version 00

- One solution for DBOUND problem.
11. References

11.1. Normative References


11.2. Informative References

https://tools.ietf.org/html/draft-deccio-dbound-name-relationships-00

[Boundaries-Problem]

[publicsuffix.org]
Mozilla Foundation, "Public Suffix List", also known as: Effective TLD (eTLD) List.
https://publicsuffix.org/

Authors’ Addresses

Jiankang Yao  
CNNIC  
4 South 4th Street, Zhongguancun, Haidian District  
Beijing, Beijing 100190  
China  
Phone: +86 10 5881 3007  
Email: yaojk@cnnic.cn

Ning Kong  
CNNIC  
4 South 4th Street, Zhongguancun, Haidian District  
Beijing, Beijing 100190  
China  
Phone: +86 10 5881 3147  
Email: nkong@cnnic.cn

Xiaodong Li  
CNNIC  
4 South 4th Street, Zhongguancun, Haidian District  
Beijing, Beijing 100190  
China  
Phone: +86 10 5881 3020  
Email: xl@cnnic.cn