Domain Boundaries
Internet-Draft
Intended status: Standards Track
Expires: July 29, 2016

J. Yao
N. Kong
X. Li
CNNIC

January 26, 2016

# Resource Record for DNS Administrative Boundaries
# draft-yao-dbound-dns-solution-02

Abstract

Two or more DNS names may have the same DNS administrative boundaries. 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 July 29, 2016.

Table of Contents



## 1. Introduction

Two or more DNS [RFC1034] [RFC1035] names may have the same administrative boundaries. 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.

With the growth of new gTLD program, it is very common for a company to have many domain names for the same aim. So we should design a method for judging the two or more domain names which share the administrative boundaries.

## 2. Terminology

The basic key words such as "MUST", "MUST NOT", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "MAY", and "MAYNOT" 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. 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 Reserved 1 field, a 1 octet Reserved 2 field, and a Anchor Name / Name Collection field.

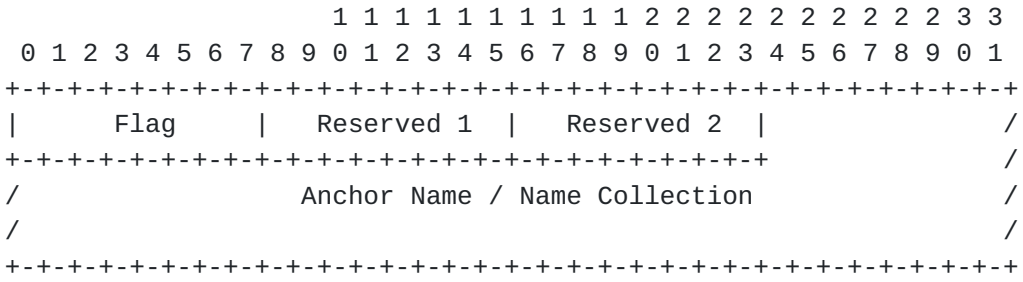


Figure 2. The structure of RDATA of Dbound resource record

Flag:
The Flag field identifies the usage of Anchor Name / Name Collection field. 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)

Reserved 1 field and Reserved 2 field:
These two fields will be kept for the future use.

Anchor Name / Name Collection:
There are two kinds of relationship mechanism, one is controlled by PSL; the other is specified by building the connection among names. If Flag is 0, the Anchor Name / Name Collection field will have the value of PSL; If Flag is 1, the Anchor Name / Name Collection field will have the value of the anchor name. The anchor name acts like a middleman. All names sharing the same administrative boundaries will point to the same anchor name; If Flag is 2, the Anchor Name / Name Collection field will have the value of name collection with names separated by comma (,).

## 4. Application Algorithm for Dbound Query

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

Case 1: If A and B's flag value in the dbound record are 0, application should confirm that the Anchor Name / Name Collection fields of both names have the value of PSL.

Case 2: If A and B's flag value in the dbound record are 1, application should check whether the names point to the same anchor.

Algorithm 1:
1)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.

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 for A, it checks whether there is a dbound record with the flag value of 0 or 1. If the value of flag of A's dbound records is 0, go to step 2; If the value of flag of A's dbound records is 1, go to step 3; Otherwise, go to step 4;

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 the value of flag of B's dbound records is 0, check whether the value of anchor name of A and B's dbound records are PSL. If yes, it means that A and B enjoys the same administrative boundaries under the PSL and exit. Otherwise go to step 4

Step 3, 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 the value of flag of B's dbound records is 1, check whether the value of anchor name of A and B's dbound records are same. If yes, it means that A and B enjoys the same administrative boundaries under the same anchor name and exit. Otherwise go to step 4

Step 4, Exit and display some error information

2)Given name A, check who shares the same administrative boundaries with A.
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 for A, it checks whether there is a dbound record with the flag value of 2. If yes, check the value of name collection of A's dbound record, find name list, check every name on the name list with A to confirm whether they enjoy the same administrative boundaries via the method 1) above.

3)Given more names than two, check whether they shares the same administrative boundaries.
The application selects one of the names as A and check whether every other name share the same administrative boundaries with A via the the method 1) above.

For examples:

EXAMPLE 1, if a.example and b.exmaple 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.exmaple
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.exmaple 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.

## 5. Wildcard issue

The parent name may announce that all names under it to share the same administrative boundaries with itself, but it needs two-way assertion here. Parents can say all its children are under its control and share the same boundaries. In the other hand, the children should confirm that they share the same boundary with its parents too.

For example:

example.com dbound 1 example.com
*.example.com dbound 1 example.com
example.com dbound 2 example.com, *.example.com

It means that example.com and its children share the same administrative boundaries.

In some cases, the children may lose its parent's control by configure some DNS records for themselves. The debound record has similar same limitation with the wildcard. Wildcards work for the non-configured sub-domain names only. Those names which can not queried through wildcard will not work for dbound too. Those names should configure their own dbound record separately instead of wildcard dbound configuration.

For example:
If there is an A record at a.b.example.com, the wildcard will not match a.b.example.com or b.example.com. In this exmaple, quering c.example.com will work if c.example.com is not configured in some ways. If there is a A record for a.b.example.com, it indicates that the a.b.exmaple.com or b.exmaple.com might exist. so under this situation, a.b.exmaple.com or b.exmaple.com should configure their own dbound record since a.b.exmaple.com or b.exmaple.com may be out of control of the parents.

## 6. Discussion

This section will be removed if it is published.

It is an initial design. It is open to change and will follow the WG's decision

## 7. IANA Considerations

The IANA should allocate the new DNS type for DBOUND.

## 8. Security Considerations

To be decided.

## 9. Acknowledgements

Thanks a lot for WG discussion in dbound WG. Especially thanks for Andrew Sullivan and John R Levine's kind comments and helpful debate.

## 10. Change History

RFC Editor: Please remove this section.

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

- One solution for DBOUND problem.

### 10.2. draft-yao-dbound-dns-solution: Version 01

- add the new method in section of discussion

### 10.3. draft-yao-dbound-dns-solution: Version 02

- update the draft based on the new method discussed in Japan IETF meeting 2015.

## 11. References

### 11.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <http://www.rfc-editor.org/info/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <http://www.rfc-editor.org/info/rfc1035>.

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

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <http://www.rfc-editor.org/info/rfc4033>.

[RFC5585] Hansen, T., Crocker, D., and P. Hallam-Baker, "DomainKeys Identified Mail (DKIM) Service Overview", RFC 5585, DOI 10.17487/RFC5585, July 2009, <http://www.rfc-editor.org/info/rfc5585>.

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, DOI 10.17487/RFC6125, March 2011, <http://www.rfc-editor.org/info/rfc6125>.

[RFC6265] Barth, A., "HTTP State Management Mechanism", RFC 6265, DOI 10.17487/RFC6265, April 2011, <http://www.rfc-editor.org/info/rfc6265>.

[RFC7208] Kitterman, S., "Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1", RFC 7208, DOI 10.17487/RFC7208, April 2014, <http://www.rfc-editor.org/info/rfc7208>.

## 11.2. Informative References

[Boundaries-Concepts]
Deccio, C. and J. Levine, "Concepts for Domain Name Relationships", draft: dbound concepts, July 2015.

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

[Boundaries-Problem]
Sullivan, A., Hodges, J., and J. Levine, "DBOUND: DNS Administrative Boundaries Problem Statement", draft: dbound problem, July 2015.

https://tools.ietf.org/html/draft-sullivan-dbound-problem-statement-01

[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