

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: 13 March 2026

B. Zhang, Ed.  
Y. Zhang, Ed.  
Pengcheng Laboratory  
J. Yao, Ed.  
CNNIC  
R. Yang, Ed.  
Y. Feng, Ed.  
Pengcheng Laboratory  
9 September 2025

A Technique for Querying the Designated Authoritative Server Directly on  
the Recursive Server at the Enterprise Level  
draft-zhang-dnsop-zb-01

Abstract

A DNS lookup usually requires the recursive server to start an iterative query process from the DNS root server to the bottom authoritative server. Attacks may occur at any level when querying authoritative servers. If the DNS recursive resolver operator can obtain the IP addresses of the authoritative servers for the queried domain, they may want to query the authoritative server directly on the recursive server. In such a way, the resolver can start the iterative query process from the designated authoritative server, greatly decreasing the round-trip time, avoiding the attacks on the upper-level and preventing snooping by third parties of requests sent to upper-level authoritative servers. This document shows a technique for querying the designated authoritative server directly on the recursive server at the enterprise level, at the cost of adding some operational fragility for the operator.

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 <https://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 13 March 2026.

## Copyright Notice

Copyright (c) 2025 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Requirements Language . . . . .	3
2. Use Case . . . . .	4
3. Applicable Scenarios of Static-stub Zone . . . . .	8
4. Requirements . . . . .	8
5. Operation of the Static-stub Zone on the Resolver . . . . .	9
6. IANA Considerations . . . . .	10
7. Security Considerations . . . . .	10
8. Normative References . . . . .	10
Authors' Addresses . . . . .	11

## 1. Introduction

DNS recursive resolvers have to answer all queries from their customers. For each queried name that has a top level domain (TLD) that is not in the recursive resolver's cache, the resolver must start an iterative query from DNS root server to the bottom authoritative server. If there is a slow path between the recursive resolver and the queried servers, getting slow responses to these queries has a negative effect on the resolver's customers.

Many of the queries from recursive resolvers to root or authoritative servers get answers that are referrals to other servers. Malicious third parties might be able to observe that traffic on the network between the recursive resolver and authoritative servers.

[RFC7706] and [RFC8806] propose the idea of running root zone replicas locally to achieve the goal of reducing root zone resolution latency and reducing DNS information leakage. The scheme can achieve "decentralization" through local root zone resolution.

This document describes a method for the operator of a recursive resolver to have the authoritative servers' referral locally and to hide queries for the upper-level authoritative servers from outsiders. The basic idea is to make the recursive server start the iterative query from the designated authoritative server instead of the root server for a domain lookup.

This document presents a more flexible scheme for local configuration of authoritative servers' information than [RFC7706] and [RFC8806]. The scheme is recommended to be used on recursive servers at the enterprise level to serve his own employees on which the authoritative domain information at the lower level can be configured. The configured authoritative domain is usually the enterprise own authoritative domain whose information can be wholly controlled by the operator.

The primary goal of this design is to provide more reliable answers for queries to the authoritative zone during network attacks that affect the root or upper-level authoritative servers and to prevent queries and responses from being visible on the network. It also provides a way to alleviate cache pollution.

This design uses an authoritative service running on the same machine as the recursive resolver. Common open source recursive resolver software does not need to add new functionality to act as an authoritative server for some zones, but other recursive resolver software might need to be able to talk to an authoritative server running on the same host.

The scheme is realized by a special zone called "static-stub" zone running on the authoritative service. The zone data in a static-stub zone is statically configured, rather than transferred from a primary server; and when recursion is necessary for a query that matches a static-stub zone, the locally configured data is always used, even if different authoritative information is cached.

### 1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 2. Use Case

Resolvers handle recursive user queries and provide complete answers; that is, they issue one or more iterative queries to the DNS hierarchy. For example, visiting `www.pcl.ac.cn` will make the resolver start the iterative query process from the root server to the authoritative server of `.cn`, and to the authoritative server of `.pcl.cn`, until to the bottom authoritative server of `.pcl.ac.cn`.

Having obtained a complete answer (or an error), a resolver passes the answer to the user and places it in its cache. Subsequent user requests for the same query will be answered from the resolver's cache until the TTL of the cached answer has expired, when it will be flushed from the cache; the next user query that requests the same information results in a new series of queries to the DNS hierarchy.

However, some attacks may happen in the iterative query procedure, which may lead to a domain cannot be resolved, or wrong/malicious resolving results.

- \* The first attack may be the DDoS attack to the root servers, or level-1 authoritative servers of `.cn`, or level-2 authoritative servers of `.ac.cn`. The upper-level authoritative servers are always the targets of these malicious parties. For example, during the period of November 30th, 2015 to December 1st, 2015, most of the 13 root servers were attacked by DDoS, which had a serious impact on the global DNS service.
- \* The second attack is the snooping by third parties of requests sent to upper-level authoritative servers, which may lead to user privacy information leakage.
- \* The third attack may come from the cache poison or DNS hijack by third malicious parties or the registrar himself through hijacking/MITM attacks. For example, On June 5, 2025, Aliyun's monitoring system detected anomalies in the DNS resolution for the domain `aliyuncs.com`, resulting in widespread failures for global users attempting to access services such as Object Storage Service (OSS) and Content Delivery Network (CDN). The anomalies were caused by unauthorized modifications to the NS records for `aliyuncs.com` through the domain registrar VeriSign. DNSSEC is an effective mechanism to prevent hijacking attacks, but its deployment progress in enterprises is relatively slow. DNSSEC is mainly deployed in root servers, level-1 authoritative servers, some level-2 authoritative servers, but seldom deployed in level-3 or lower authoritative servers. This makes the DNS hijacks to the lower-level authoritative servers possible.

For an institution such as Pengcheng Laboratory, he may run a recursive server for his employees. He registered `pcl.ac.cn` with a registrar named the Beijing Xinwang Digital Information Technology Co., Ltd. The registrar can be asked to provide the authoritative information of `pcl.ac.cn` such as Table 1 shows to the recursive server of Pengcheng Laboratory, and any update of these information can be asked to be notified to the resolver in a timely manner. That is, the operator of the institution has complete control on the authoritative information of its own authoritative domain.

Domain	Authoritative servers	IP addresses
pcl.ac.cn	ns3.dnsv2.com	117.89.178.226
		1.12.0.29
		36.155.149.180
		163.177.5.85
		125.94.59.205
	ns4.dnsv2.com	117.89.178.204
		36.155.149.242
		163.177.5.38
		112.80.181.103

Table 1: Authoritative information of `pcl.ac.cn`

In such a scenario, the resolver may want to start the query for `www.pcl.ac.cn` access directly from the authoritative servers of `pcl.ac.cn` instead of the root servers for security considerations. This function can be realized by a special zone called "static-stub" zone in some open source recursive resolver software such as Bind or Unbound. The "static-stub" zone has also been realized in some commercial DNS resolution software such as Maple of ZDNS.

Here, Bind 9.18 is used to show the configuration of "static-stub" zone. BIND 9.18 acts both as a recursive resolver and an authoritative server. We configure the static-stub zone in the recursive server based on the authoritative data in Table 1 as Figure 1 shows.

```
zone "pcl.ac.cn" {  
    type static-stub;  
    server-addresses { 117.89.178.226; 1.12.0.29; 36.155.149.180; 163.177.5.85; 125.94.59  
.205; 163.177.5.38; 112.80.181.103; 117.89.178.204; 36.155.149.242;};  
};
```

Figure 1: Resolver Static-stub Zone Configuration

The zone data in a static-stub zone in Bind is configured via the `server-addresses` and `server-names` zone options. The zone data is maintained in the form of NS and (if necessary) glue A or AAAA RRs internally, which can be seen by dumping zone databases with `rndc dumpdb -all`. The configured RRs are considered local configuration parameters rather than public data. Non-recursive queries (i.e., those with the RD bit off) to a static-stub zone are therefore prohibited and are responded to with REFUSED.

The `server-addresses` option in Bind is especially set for static-stub zones. This is a list of IP addresses to which queries should be sent in recursive resolution for the zone. A non-empty list for this option internally configures the apex NS RR with associated glue A or AAAA RRs. For example, if `pcl.ac.cn` is configured as a static-stub zone as Figure 1 shows, the following RRs are internally configured:

```
pcl.ac.cn. NS pcl.ac.cn.  
pcl.ac.cn. A 117.89.178.226  
pcl.ac.cn. A 1.12.0.29  
pcl.ac.cn. A 36.155.149.180  
pcl.ac.cn. A 163.177.5.85  
pcl.ac.cn. A 125.94.59.205  
pcl.ac.cn. A 163.177.5.38  
pcl.ac.cn. A 112.80.181.103  
pcl.ac.cn. A 117.89.178.204  
pcl.ac.cn. A 36.155.149.242
```

Figure 2: Internal RR generated

With the configuration, The function of querying the designated authoritative servers directly on the resolver can be realized. That is, the recursive server can start the iterative query directly from the configured addresses of the longest domain match zone when resolving a domain. For example, when we query `www.pcl.ac.cn` on the recursive server, the recursive server will start the iterative query process directly from one of the `server-addresses` of `pcl.ac.cn` static-stub zone instead of the root server, thus avoiding the attacks happened in upper-level authoritative servers. This static-stub configuration of this use case can achieve four advantages.

- \* The configuration can make the resolution for pcl.ac.cn domains and sub-domains immune to the DDoS attacks of the upper-level. For example, if attacks happen in root or .cn/.ac.cn authoritative servers, the www.pcl.ac.cn query sent to the recursive server without static-stub configuration may not be resolved correctly when cache records expires, but will be resolved correctly with static-stub configuration.
- \* The configuration can avoid the snooping by third parties of requests sent to upper-level authoritative servers, and improve the resolution efficiency for pcl.ac.cn domains and sub-domains since it shortens the iterative process.
- \* The configuration can make the resolution for pcl.ac.cn domains and sub-domains immune to cache poisoned or DNS hijack by third malicious parties through hijacking/MITM attacks. The poisoned NS and glue records of pcl.ac.cn in cache has no influence to the resolution for pcl.ac.cn domains and sub-domains since the static-stub zone has higher priority than cache for the same record.
- \* The configuration can make the changed authoritative information take effect more fast than cache. When the authoritative server's addresses of .pcl.ac.cn change, the cache will still keep the stale records until they are expired, the resolving results may be influenced by the stale records. The in time update of the static-stub zone for .pcl.ac.cn will avoid the mistake.

Similar to BIND, Unbound, starting with version 1.8, can act both as a recursive resolver and an authoritative server. Unbound uses the stub-zone to realize the static-stub zone function of Bind. We configure the stub-zone in the unbound resolver implementation based on the authoritative data in Table 1 as Figure 3 shows.

```
stub-zone:
name: "pcl.ac.cn"
stub-addr: 117.89.178.226
stub-addr: 1.12.0.29
stub-addr: 36.155.149.180
stub-addr: 163.177.5.85
stub-addr: 125.94.59.205
stub-addr: 163.177.5.38
stub-addr: 112.80.181.103
stub-addr: 117.89.178.204
stub-addr: 36.155.149.242
stub-first: yes
```

Figure 3: Unbound stub-zone Configuration

### 3. Applicable Scenarios of Static-stub Zone

From the use case in Section 2, we describe the applicable scenarios of static-stub zone:

- \* The static-stub zone is recommended to be used for resolvers of an institution at the enterprise level to provide DNS services for his own employees, and the authoritative domain of this institution is recommended to be configured as a static-stub zone. The corresponding authoritative domain information of the institution should be completely controlled by the institution. That is, the institution and its registrar should build a secure connection for transporting the authoritative domain information configured in the static-stub zone. Any update of the authoritative domain information should be notified to the institution securely in a timely manner.
- \* The static-stub zone is recommended to be configured for level-2 or lower authoritative domains, without DNSSEC deployment.
- \* An institution with static-stub zone mechanism is recommended to build a static-stub alliance with another institution with static-stub mechanism if the two institutions have close businesses. The institution in a static-stub alliance can configure static-stub zones for other institutions' authoritative domains if they can share authoritative domains information synchronously and securely.

### 4. Requirements

In order to implement the mechanism described in this document:

- \* The system MUST be able to validate every signed record in a zone with DNSSEC [RFC4033].
- \* The system MUST have an up-to-date copy of the public part of the Key Signing Key (KSK) [RFC4033] used to sign the DNS root.
- \* The system MUST be able to run an authoritative service on the same host. The authoritative service MUST only respond to queries from the same host. One way to ensure that the authoritative service does not respond to queries from other hosts is to run an authoritative server for the static-stub zone service that responds only on one of the loopback addresses (that is, an address in the range 127/8 for IPv4 or ::1 in IPv6). Another method is to have the resolver software also act as an authoritative server for the static-stub zone, but only for answering queries from itself.



- \* The system MUST be able to get the authoritative data in the static-stub zone and update the static-stub zone in time when the authoritative data changes. The authoritative data MUST be identical to or part of the data in the authoritative zone for the DNS. It is possible to change the data (the NS or glue records) of the authoritative zone, but such changes could cause problems for the recursive server that accesses the local static-stub zone, and therefore any changes to the data SHOULD NOT be made.
- \* The system is recommended to establish an automatic static-stub zone update mechanism to avoid the manual configuration error.

## 5. Operation of the Static-stub Zone on the Resolver

The operation of an authoritative server for the static-stub zone in the system described here can be done separately from the operation of the recursive resolver, or it might be part of the configuration of the recursive resolver system.

The steps to set up the static-stub zone are:

1. Get the authoritative servers' information of some specific authoritative domains.
2. Build or update the static-stub zones based on the authoritative servers' information of the authoritative domains.
3. Start or restart the authoritative service for the static-stub zone in a manner that prevents any system other than a recursive resolver on the same host from accessing it.

Since the data is statically configured, no zone maintenance action takes place for a static-stub zone. For example, there is no periodic refresh attempt, and an incoming notify message is rejected with an rcode of NOTAUTH. Each static-stub zone is configured with internally generated NS and (if necessary) glue A or AAAA RRs.

There is a risk that a system using a local authoritative server for the static-stub zone cannot refresh the contents of the static-stub zone when a authoritative server's information changes. A system using a local authoritative server for the static-stub zone MUST NOT serve stale data. To mitigate the risk that stale data is served, the local resolver MUST immediately delete the stale data in the static-stub zone when it detects that it would be serving stale data.

In the event that refreshing the contents of the static-stub zone fails, the results can be disastrous. For example, sometimes all the NS records for a domain are changed in a short period of time (such

as 2 days); if the refreshing of the static-stub zone is broken during that time, the recursive resolver will have bad data for the entire domain zone.

An administrator using the procedure in this document SHOULD have an automated method to check that the contents of the static-stub zone are being refreshed; this might be part of the resolver software. One way to do this is to have a separate process that periodically checks the authoritative information to the corresponding authoritative server and makes sure that it is changing.

An administrator using the procedure in this document SHOULD have an automated method to build and update from data sources to static-stub zone. After building or updating the static-stub zone, the administrator should restart the static-stub zone service to function immediately without interrupting the resolving service.

## 6. IANA Considerations

This memo includes no request to IANA.

## 7. Security Considerations

The static-stub mechanism does not influence the DNSSEC validation procedure, since it is recommended to be configured for level-3 or lower authoritative domains, without DNSSEC deployment.

A system that does not follow the DNSSEC-related requirements given in Section 4 can be fooled into giving bad responses in the same way as any recursive resolver that does not do DNSSEC validation on responses from a remote authoritative server. Anyone deploying the method described in this document should be familiar with the operational benefits and costs of deploying DNSSEC [RFC4033].

## 8. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

- [RFC7706] Kumari, W. and P. Hoffman, "Decreasing Access Time to Root Servers by Running One on Loopback", RFC 7706, DOI 10.17487/RFC7706, November 2015, <<https://www.rfc-editor.org/info/rfc7706>>.
- [RFC8806] Kumari, W. and P. Hoffman, "Running a Root Server Local to a Resolver", RFC 8806, DOI 10.17487/RFC8806, June 2020, <<https://www.rfc-editor.org/info/rfc8806>>.
- [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, <<https://www.rfc-editor.org/info/rfc4033>>.

## Authors' Addresses

Bin Zhang (editor)  
Pengcheng Laboratory  
Ca Guang  
Shen Zhen  
Nan Shan, 518000  
China  
Email: [bin.zhang@pcl.ac.cn](mailto:bin.zhang@pcl.ac.cn)

Yu Zhang (editor)  
Pengcheng Laboratory  
Ca Guang  
Shen Zhen  
Nan Shan, 518000  
China  
Email: [zhangy08@pcl.ac.cn](mailto:zhangy08@pcl.ac.cn)

Jiankang Yao (editor)  
CNNIC  
Building 4, No.9, Qiche bowuguan xilu Road  
Beijing  
Fengtai District, 100070  
China  
Email: [yaojk@cnnic.cn](mailto:yaojk@cnnic.cn)

Rongwei Yang (editor)  
Pengcheng Laboratory  
Ca Guang  
Shen Zhen  
Nan Shan, 518000  
China  
Email: yangrw@pcl.ac.cn

Yuming Feng (editor)  
Pengcheng Laboratory  
Ca Guang  
Shen Zhen  
Nan Shan, 518000  
China  
Email: fengym@pcl.ac.cn