Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 28, 2015

W. Kumari, Ed.
Google
P. Hoffman, Ed.
VPN Consortium
October 25, 2014

# Decreasing Access Time to Root Servers by Running One on Loopback
## draft-wkumari-dnsop-root-loopback-00

## Abstract

Some DNS recursive resolvers have longer-than-desired round trip times to the closest DNS root server. Such resolvers can greatly decrease the rount trip time by running a copy of the full root zone on a loopback address (such as 127.0.0.1). This document shows how to start and maintain such a copy of the root zone in a manner that is secure for the operator of the recursive resolver and does not pose a threat to other users of the DNS.

## 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 28, 2015.

## Copyright Notice

Table of Contents

1. Introduction . . . . . 2
1.1. Requirements Notation . . . . . 3
2. Requirements . . . . . 3
3. Operation of the Root Zone on the Loopback Address . . . . . 4
4. Using the Root Zone Server on the Loopback Address . . . . . 4
5. IANA Considerations . . . . . 4
6. Security Considerations . . . . . 4
7. Acknowledgements . . . . . 5
8. Normative References . . . . . 5
Appendix A. Current Sources of the Root Zone . . . . . 5
Authors' Addresses . . . . . 5

## 1. Introduction

DNS recursive resolvers have to answer all queries from their customers, even those which are for domain names that do not exist. For each queried name that has a top level domain (TLD) that is not in the recurive resolver's cache, the resolver must send a query to a root server to get the information for that TLD, or to find out that the TLD does not exist. If there is a slow path between the recursive resolver and the closest root server, getting slow responses to these queries has a negative effect on the resolver's customers.

This document describes a method for the operator of a recursive resolver to greatly speed these queries. The basic idea is to create a validated, up-to-date root zone server on a loopback address on the same host as the recursive server, and that server is added to the list of root zones that the recursive resolver uses for looking up root information. If the new server is working correctly, it will quickly become the preferred root server for the recursive resolver; if the new server fails to work (such as because it cannot get updates to the zone), the recursive resolver will use other root servers, as it does now.

The primary goal of this design is to provide faster negative responses to stub resolver queries that contain junk queries. This

design will probably have little effect on getting faster positive responses to stub resolver for good queries on TLDs, because the data for those zones is usually long-lived and already in the cache of the recursive resolver; thus, getting faster positive responses is a non-goal of this design.





This design explicitly only allows the new root zone server to be run on a loopback address. This prevents the server from serving authoritative answers to any system other than the recurisve resolver.

This design can possibly be implemented by hand, but it is much more likely that the creators of recursive resolver software will implement it and the operator simply needs to turn on the feature. Note that this design requires the addition of authoritative name server software running on the same machine as the recursive resolver. Thus, recursive resolver software such as BIND will not need to add much new functionality, but recursive resolver software such as Unbound will need to add software that acts as an authoritative server.

## 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. Requirements

In the discussion below, the term "legacy operation" means the way that a recursive resolver acts when it is not using the mechanism describe in this document, namely as a normal validating recursive resolver with no other special features.

In order to implement the mechanism described in this document:

- The system MUST be able to validate a zone with DNSSEC.
- The system MUST have an up-to-date copy of the DNS root key.
- The system MUST be able to retrieve a copy of the entire root zone (including all DNSSEC-related records).

- The system MUST be able to run an authoritative server on one of the IPv4 loopback addresses (that is, an address in the range 127/8).

- The authoritative server in the system MUST send error responses (RCODE 2, also known as "SERVFAIL") if the validated data in the root zone it is serving is out of date.

- The recursive resolver MUST be able to add an additional address to its list of addresses of authoritative root servers, and MUST treat that additional address the same as the other addresses.





## 3. Operation of the Root Zone on the Loopback Address

The operation of an authoritative server for the root in the system described here can be done separately from the operation of the recursive resolver.

The steps to set up the root zone are:

1. Retrieve a copy of the root zone. (See Appendix A for some current locations of sources.)

2. Validate the zone using normal DNSSEC validation.

3. Start the authoritative server with the root zone on a loopback address that is not in use. This would typically be 127.0.0.1, but if that address is in use, any address in 127/8 is acceptable.

The contents of the root zone must be refreshed using the timers from the SOA record in root zone, as described in [RFC1035]. If the contents of the zone cannot be refreshed with validated information before the expire time, the server MUST return a SERVFAIL error response for all queries until the zone can be successfully be set up again.

## 4. Using the Root Zone Server on the Loopback Address

A recursive resolver that wants to use a root zone server operating as described in Section 3 simply adds the address of the server to

its list of authoritative servers for the root zone. The resolver's round-robin search mechanism will begin to strongly prefer this new server, just as it would any root zone server that has an extremely short round trip time.

## 5. IANA Considerations

This document requires no action from the IANA.

## 6. Security Considerations

A system that does not follow the DNSSEC-related requirements given in Section 2 can be fooled into givning bad responses in the same way as any recursive resolver that does not do DNSSEC validation on responses from the root zone.





## 7. Acknowledgements

The editors fully acknowledge that this is not a new concept, and that we have chatted with many people about this. In fact, this concept may already have been implemented without the knowledge of the authors.

## 8. Normative References

[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.

## Appendix A. Current Sources of the Root Zone

The root zone can be retrieved from anywhere as long as it comes with all the DNSSEC records needed for validation. Currently, there are three well-known sources of the root zone:

- From ICANN via FTP at ftp://rs.internic.net/domain/root.zone

- From ICANN via HTTP at http://www.internic.net/domain/root.zone

- From ICANN by AXFR from DNS servers at xfr.lax.dns.icann.org and xfr.cjr.dns.icann.org

## Authors' Addresses

Warren Kumari (editor)
Google

Email: Warren@kumari.net

Paul Hoffman (editor)
VPN Consortium

Email: paul.hoffman@vpnc.org