IPv6-Only DNS

Author: Alejandro Acosta, José G. Cotúa
Coordination and Revision: Guillermo Cicileo
Edition: Communications Department
Department: Technology

lacnic
www.lacnic.net

Introduction

The following article presents the considerations that should be taken into account and the steps that should be followed when operating an IPv6-only DNS server.
We will try to address various basic aspects and the main functionalities. However, it is important to remember that there are many additional features to consider.

Requirements

This article discusses the administration and operation of a DNS server on GNU/Linux, with a server-oriented installation. For this study we will use the DNS BIND9 server installed on Debian Linux.

BIND9 is one of the most widely used DNS servers in ISP and network operator network environments.

The BIND9 server must be previously installed on your Linux operating system, preferably an updated version and/or a version higher than v9.10. However, the concepts discussed in this article apply to any DNS server regardless of the Linux distribution, and even to any other DNS server.

In any case, the reader should be familiar with the basics of server operation and configuration in GNU/Linux environments and the basic operation of BIND9, including how to configure options, how to configure zones, and other configurations.

Network topology to be used for the study

IPv6 addressing for the topology:

- **Client network prefix:** 2001:db8:abcd::/48
- **Recursive DNS Server:** 2001:db8:123::2
- **Authoritative DNS servers:** 2001:db8:cafe::2 and 2001:db8:cafe::3

Important information to better understand this document:

The following configuration files will be used:

/etc/bind/named.conf
/etc/bind/named.conf.options
/etc/bind/named.conf.default-zones

General setup and first steps

a) First, we must make sure that Bind9 is installed. To do so, we can run the following command in our Linux console:

```
#named -v
BIND 9.16.15-Debian (Stable Release) <id:4469e3e>
```

We can also run the following command to check that the Bind9 server is up and running normally:

```
#systemctl status named
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2022-MM-DD HH:MM:SS UTC; 1 day 8h ago
     Docs: man:named(8)
 Main PID: 2779031 (named)
    Tasks: 26 (limit: 9507)
   Memory: 156.5M
      CPU: 2min 8.314s
   CGroup: /system.slice/named.service
           └─2779031 /usr/sbin/named -f -u bind
```

This command shows not only the status of the Bind9 server's operation, but also parameters such as PID, memory, uptime, and others.

b) Next, we must check the server's IPv6 addressing configuration. To do so, it is important to know the Linux device's interfaces and their IP addresses. It is important to know in advance the IPv6 address on which we want Bind9 to listen for incoming connections and those from which outgoing connections will be made.

It is even common to use 'bridge' interfaces in Linux to configure IPv6 addressing to be used by Bind9. This practice is recommended to avoid flapping and increase the stability of the DNS service.
There are many ways to manage the interfaces and IPv6 addressing of a Debian Linux server. Below are some of the most common.

Checking the server's network interface configuration:

```
        #ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group
default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT
group default qlen 1000
   link/ether 9e:11:7b:f0:98:b9 brd ff:ff:ff:ff:ff:ff
   altname enp0s18
3: loopback: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode
DEFAULT group default qlen 1000
   link/ether 4e:c0:be:49:cb:23 brd ff:ff:ff:ff:ff:ff
```

Checking the IPv6 addresses configured on the server:

```
#ip -6 address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 state UNKNOWN qlen 1000
   inet6 ::1/128 scope host
      valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 state UP qlen 1000
   inet6 2001:db8:cafe::2/64 scope global
      valid_lft forever preferred_lft forever
   inet6 fe80::9c11:7bff:fef0:98b9/64 scope link
      valid_lft forever preferred_lft forever
```

The commands above will show the server's network interfaces and display the IPv6 addresses assigned to them. In our case, we will use the ens18 interface with the 2001:db8:cafe::2/64 IPv6 address (GUA). We can also see the interfaces' Link Local Addresses (LLA).

Knowing the server's network interfaces and their IPv6 addresses is important when troubleshooting, both Linux and the Bind9 server.

DNS basics: authoritative vs. recursive servers

Many different types of DNS servers exist, but recursive servers and authoritative servers are the most common.

According to dnslookup.es [2]:

Authoritative DNS server: A server with a local database which holds the definitive records for a given domain name.

Recursive server: If a recursive server does not find an answer in its local database for a given name, it will query other DNS servers to try to solve the query it has received.

In our example (see topology above):

- The recursive DNS server provides DNS service to clients on the 2001:db8:abcd::/48 network
- The authoritative servers are the authority for the example.com domain
- In addition, the 2001:db8:cafe::3 server copies the zone via “zone transfer” from 2001:db8:cafe::2

Configuring an IPv6-only recursive DNS server

Step 1:

After having looked up and checked the network interfaces, the IPv6 addressing of the Linux server, and the normal operation of Bind9, it's time to start configuring the Bind9 server, first to make sure that it listens on IPv6 and allows incoming connections from the 2001:db8:abcd::/48 network.

The Bind9 server's main configuration file is '**/etc/bind/named.conf**'. We will now configure the correct IPv6 parameters in the '**options{ …};**' section located in the first part of this file. The '*include*' directive can be used in the configuration file to reference other files and to program the configurations in a more orderly manner, using several files and segmenting the configuration accordingly.

How to configure the 'options { … };' section for IPv6 operation is shown below:

```
options {
    directory "/var/cache/bind";
    forwarders {
        2001:db8:db8::8888;
        2001:db8:db8::8844;
    };
    listen-on-v6 { any; };
    query-source-v6 address 2001:db8:cafe::2;
    listen-on-v6 { ::1; 2001:db8:cafe::2; };
    recursion yes;
    …;
};
```

For simplicity reasons, we have only included the IPv6 parameters.

Below is a brief explanation of the most important IPv6 configuration parameters:

- The *'forwarders { 2001:db8:db8::8888; 2001:db8:db8::8844; };'* line is optional and should only be included if we wish to forward all DNS queries to another DNS server. In this case, we have configured two IPv6 forwarding servers.
- The *'listen-on-v6 { any; };'* option tells Bind9 on which IPv6 addresses it should listen for IPv6 DNS requests. We can use the 'any' directive to specify that Bind9 will listen on every IPv6 address configured on the Linux server, even localhost6 '::1'. We can also specify the IPv6 addresses on which we want Bind9 to listen, specifying the IPv6 addresses separated by ';'. For example:
  listen-on-v6 { ::1; 2001;db8:cafe::2; };
  Any IPv6 address we specify here must have already been configured on a Linux server interface.

- The *'query-source-v6 address 2001:db8:cafe::2;'* option specifies the IPv6 address(es) from which Bind9 will establish outgoing connections. Outgoing connections are essential for the server to resolve DNS queries either through forwarding or recursion.

It should be noted that, once the configuration '/etc/bind/named.conf' file has been updated, we must apply these changes by restarting the Bind9 server or reloading its configuration using the commands below:

#systemctl reload named
#systemctl restart named

After applying the changes, we can run the following commands to validate that Bind9 is listening on the configured IPv6 addresses:

#netstat -puan | grep named

Step 2:

We must allow recursion for our clients, specifically clients on the 2001:db8:abcd::/48 network. To do so, we must use the '**allow-recursion**' directive, but keep in mind that the behavior of this directive has changed in more recent versions of Bind. In this case, we create an ACL with the IPv6 address of the client network and apply the directive as follows:

```
;;;Bind9 configuration, file /etc/bind/named.conf, for recursion.
acl my_ipv6_net {
   2001:db8:abcd::/48;
};

options {
          …
       recursion yes;
          allow-recursion { my_ipv6_net; };
          …
};
```

In the previous example, we are creating an ACL called *'my_ipv6_net'* and then allowing recursion to clients whose addresses match the ACL. In other words, our clients will be perfectly able to use the server.

Important security recommendation: Never use the *'allow-recursion { any; }'* option, as that would create an 'Open Resolver' server with all that this implies.

Configuring an IPv6-only authoritative DNS server

Step 1: General and basic configuration

Just as we did for a recursive DNS server, we need to make sure that Bind9 listens and allows connections on IPv6. In other words, we maintain the IPv6 network configuration of the server and the general and basic configuration of Bind9.

Step 2: Configuration for authoritative 'only'

If we want Bind9 to '**only**' be authoritative and not allow recursion, we must configure the options below in the '/etc/bind/named.conf' file:

```
options {
    directory "/var/cache/bind";
    forwarders {
        2001:db8:db8::8888;
        2001:db8:db8::8844;
    };
    listen-on-v6 { any; };
    query-source-v6 address 2001:db8:cafe::2;
    listen-on-v6 { ::1; 2001:db8:cafe::2; };
    recursion no;
  …;
};
```

Step 3:

Because the DNS server is authoritative, we need to set the 'example.com' zone configuration in the '/etc/bind/named.conf' file. This is how we do it:

```
options {
        …;
};
zone "example.com" {
    type master;
    file "/etc/bind/example.com.db";
};
```

Important note: A DNS server can be both authoritative and recursive at the same time, but this type of configuration requires a more in-depth analysis of the security aspects involved.

Zone transfer configuration

Configuring the primary DNS server (master)

When configuring the primary DNS server's '/etc/bind/named.conf' file, we must specify that it is a master server and define the IPv6 addressing of who can request zone transfer.

The corresponding configuration is shown below:

```
options {
        ...;
};
zone "example.com" {
   type master;
   file "/etc/bind/zones/example.com.db";
   allow-transfer { 2001:db8:cafe::3; };
};
```

Configuring the secondary DNS server (slave)

When configuring the secondary DNS server's '/etc/bind/named.conf' file, we must specify that it is a slave server and define the IPv6 addressing of the master servers.

The corresponding configuration is shown below:

```
options {
        ...;
};
zone "example.com" {
   type slave;
   file "example.com.db";
   masters { 2001:db8:cafe::2; };
};
```

Summary of the Bind9 server configuration for IPv6 only

```
;;;Authoritative, no recursion
;;;Master
;;;File: /etc/bind/named.conf

acl my_ipv6_net {
   2001:db8:abcd::/48;
};
options {
      directory "/var/cache/bind";
      //forwarders {
      //   2001:db8:db8::8888;
      //   2001:db8:db8::8844;
      //};
      query-source-v6 address 2001:db8:cafe::2;
      listen-on-v6 { ::1; 2001:db8:cafe::2; };
      recursion no;
    …;
};
zone "example.com" {
   type master;
   file "/etc/bind/zones/example.com.db";
   allow-transfer { 2001:db8:cafe::3; };
};
```

```
;;;Authoritative, no recursion
;;;Slave
;;;File: /etc/bind/named.conf

acl my_ipv6_net {
   2001:db8:abcd::/48;
};
options {
      directory "/var/cache/bind";
      //forwarders {
      //   2001:db8:db8::8888;
      //   2001:db8:db8::8844;
      //};
      query-source-v6 address 2001:db8:cafe::3;
      listen-on-v6 { ::1; 2001:db8:cafe::3; };
      recursion no;
    …;
};
zone "example.com" {
   type slave;
   file "example.com.db";
   masters { 2001:db8:cafe::2; };
};
```

Summary of commands for Linux and Bind9 server operation

Managing interfaces and IPv6 addressing

ip link show
ifconfig -a
ip -6 address show
ip -6 route show

Managing Bind9

systemctl status named
systemctl reload named
systemctl restart named

Checking Bind9 processes and binding

netstat -puan | grep named

Testing IPv6 DNS locally

nslookup www.google.com ::1
nslookup www.google.com 2001:db8:cafe::2
nslookup -type=AAAA www.google.com ::1
dig @::1 AAAA www.google.com

Common errors when implementing an IPv6 DNS server

- Typos in the configuration of the '/etc/bind/named.conf'' file, especially in the options that use or end in the ';' symbol. To avoid this, we recommend being particularly careful when editing or creating configurations.
- Incorrect use of Bind9's authoritative server and recursion options, making it an 'Open Resolver' with all the security implications that this implies.
- Failure to properly validate the server's IPv6 addresses and misalignments between the server's and Bind9's configurations.
- Making the Bind9 server listen on IPv6 addresses configured on flapping interfaces. This causes the DNS service to be intermittent.
- Incorrectly editing the name of the zone files.

References and links of interest

- https://kb.isc.org/docs/aa-00269
- https://www.isc.org/bind/
- https://labs.lacnic.net/RRL-en-BIND9/