CIA Lab Assignment: Domain Name System (1)*

A. Bakker N. Sijm J. van der Ham M. Pouw†

Feedback deadline:
September 29, 2017 10:00 CET

Abstract

The *Domain Name System (DNS)* is a hierarchical, distributed database, which is mostly used for matching IP addresses to hostnames. DNS information is also used for mail routing and other internet applications. The practicum this week is about compiling, installing, configuring and testing DNS on your experimentation machines. You will be running a caching nameserver, but also an authoritative one. To this end, you are provided with your own subdomain in the domain `prac.os3.nl`.

1 Introduction

The best known implementation of a Domain Name Server is called *Berkeley Internet Name Domain (BIND)*. The first version of BIND was written in Berkeley, California by a group of students, and was paid for by DARPA. Currently the development of BIND is supported by the *Internet Systems Consortium (ISC)*. BIND can serve both as a caching and as an authoritative nameserver.

The *Unbound* nameserver is a caching nameserver, created in a modular fashion so that also DNSSEC validation and stub resolvers are easily possible. As Unbound is only a caching nameserver, it is combined with NSD in this practicum.

The *Name Server Daemon (NSD)* is an authoritative name server written from scratch by NLnet Labs (located at Amsterdam Science Park) in cooperation with RIPE NCC. It was written from scratch to avoid all top-level name servers running the same code and become vulnerable to the same flaws.

For this and the DNS assignments that follow, half of you will use BIND and half will use Unbound+NSD. If you have an even table number, you should use BIND, otherwise Unbound+NSD.

*Based on earlier work by E.P. Schatborn and A. van Inge. Version September 19, 2017.

†Arno.Bakker@os3.nl,mick@os3.nl

2 Downloading and Installing a Caching Nameserver

The sources for the latest version of BIND (9.10.6, as the time of writing) can be downloaded from the website `http://www.isc.org`. `Unbound 1.6.6` can be download from `http://unbound.net`. We will install `NSD 4.1.17` later.

2.1 Validating the Download

The `http://www.isc.org` website provides signature files in addition to the BIND tarball. These can be used to check if you have downloaded the version they intended to distribute. The Unbound website uses a different mechanism, in particular, a modification detection code, better known as a cryptographic hash.

1. Why is it wise to use a signature to check your download?

Download the BIND tarball (also if you are doing the Unbound+NSD part) and check its validity using one of the signatures.

2. Which signature is the best one to use? Why?

2.2 Installation Documentation

Apart from the source code, the distributions contain documentation about the servers and DNS. For BIND, the README file contains instructions on installation, and in the `doc/` directory you can find the Administrator Reference Manual (ARM) and other relevant documents. For Unbound, the `doc/` directory contains all information, including a README.

Most things about DNS are described and standardized in so-called "Request For Comments (RFCs)", created and published by the *Internet Engineering Task Force (IETF)*. A good DNS RFC to start with is RFC 1034: "Domain Names - Concepts and Facilities."

2.3 Compiling

Compiling and installing BIND, Unbound and NSD servers is simple and consists of the usual sequence of commands on most systems; `./configure`, `make` and `make install`. First, make sure your installation does not contain a previous version of of the servers, as that can really mess things up.

Next, configure, compile and then install the servers in the directory `/usr/local/`. Make sure each server will look for its configuration files in `/usr/local/etc/bind`, or `/usr/local/etc/unbound` and `/usr/local/etc/nsd`, respectively. Let the server write its state information, such as the `named.pid` file, in `/var/run`. You can find more information in the README files.

3 Configuring and Testing

Compiling and installing a server is relatively simple, but configuring a DNS server is not trivial. To keep things simple, we will start with BIND and Unbound running as a caching-only name server. This type of name server does not control any zone data.

Question

3. Why are caching-only name servers still useful?

3.1 Main Configuration

The main configuration file you will have to create for BIND is called `named.conf` and should be stored in `/usr/local/etc/bind`. For Unbound the main configuration file is `/usr/local/etc/unbound/unbound.conf`. It contains general options for the name server as well as references to other configuration files. BIND does not come with an example configuration, so see Figure 1. For Unbound, the distribution already provides an example file in the desired location.

```
// Define a access list to limit recursion later
acl localnet {
127.0.0.1/32;
};

// Working directory and limit recursion
options {
directory "/usr/local/etc/bind";
allow-recursion {
    localnet;
};
};

// Caching only DNS server
zone "." {
type hint;
file "named.cache";
};

// Provide a reverse mapping for the loopback address 127.0.0.1
zone "0.0.127.in-addr.arpa" {
type master;
file "named.local";
notify no;
};
```

Figure 1: An example configuration file for a BIND caching-only name server.

```
    $TTL 86400
    @ IN SOA <origin> <person> (
1         ; serial
360000    ; refresh every 100 hours
3600      ; retry after 1 hour
3600000   ; expire after 1000 hours
3600      ; negative cache is 1 hour
    )

       IN NS    <server>.
    0          IN PTR   loopback.
    1          IN PTR   localhost.
```

Figure 2: An example `named.local` file for the domain `0.0.127-in-addr.arpa`

3.2 Root Servers

Our server needs a file containing references to the DNS root servers, the root hint file. The root hint file contains a list of root servers that our server uses to retrieve a more recent list of root servers. This file can be downloaded from `ftp://ftp.rs.internic.net/domain`.

3.3 Resolving `localhost`

The BIND server also needs a file in the working directory called `named.local` to resolve the loopback address 127.0.0.1 to the name `localhost`. For Unbound, this is done automatically (see the `local-zone` feature).

This `named.local` file is in the standardized zone file format that you will need further on in the assignment. So take some time to study the example shown in Figure 2. The zone file format is defined in RFC 1033 which explains the `<origin>`, `<person>` and `<server>` fields.

Question

4. Now that you know all the elements of the main configuration, create a simple `named.conf` or `unbound.conf` for a caching-only name server. Show the configuration file in your log.

3.4 Testing

You can check the syntax of your configuration file by using the `named-checkconf` and `unbound-checkconf` programs, respectively. The `named-checkconf` program returns only a result value on success, `unbound-checkconf` prints a line and returns a result value.

5. Why do the programs return a result value?

One way to see the result value is as follows:

```
if named-checkconf; then echo t; else echo f; fi
```

4 Running and Improving the Name Server

You can now start the DNS server by hand using `named -g -d2`, or `unbound -d -vv`, respectively. It will start the daemon with debug level 2 (read the manual page for more information).

It would be better to configure the name server to write debug information to a log file. Configure your DNS server to write debug information to a log file.

It is also better to use a "name server control" tool to start and stop the server. For BIND this tool is called `rndc`, for Unbound it is called `unbound-control`. You'll have to adapt the configuration to enable it. This is described in Chapter 3 of the BIND ARM, or the `unbound-control` manual page, respectively.

Question

Configure the server to use remote control:

6. Show the changes you made to your configuration to allow remote control.
7. Show that remote control works.
8. What other commands/functions does `rndc`/`unbound-control` provide? Make your own list that describes the most important ones.

To use your own name server you will need to adapt `resolv.conf`. Take into account that on recent Ubuntu distributions this file is automatically generated.

9. What do you need to put in `resolv.conf` (and/or other files) to permanently use your own name server?

Now use the tools and scripts provided with your distribution to test your name server.

5 (Installing and) Configuring an Authoritative Nameserver

If you are using Unbound+NSD, you should now install and test the NSD server, in addition to Unbound, before going to the next step. You can get it from `http://www.nlnetlabs.nl/downloads/nsd/`. Follow the same steps as above for Unbound to `configure`, `make` and `install` it, also with the `/usr/local` prefix.

Now that you have checked that your configuration works correctly, you can set it up to serve your own subdomain of `os3.nl`. The zone file format is standardized, so it is the same for both BIND and NSD. Do the following:

- Use the subdomain `<city>.prac.os3.nl`, which is already delegated to your experimentation server (`<city>.studlab.os3.nl`). Now create a forward mapping zone file for your domain. It must contain the following resource records:
  - 2 MX records. Make sure that mail for your domain is delivered to your own computer. We will use the second MX record later on.
  - 4 A or AAAA records. Use your imagination...
  - 2 CNAME records.

  RFC 1178 provides useful tips for choosing names. It is also simply fun to read. Use the mentioned RFCs (1034 and 1033) for information about zone files and examples.
- Add a reference to your zone file to `named.conf` or `nsd.conf`, respectively. Restart or reload the name server and test your configuration using the provided tools.

When you are done, please answer the following questions:

10. Show the forward mapping zone file in your log.

11. If Niels had not yet implemented the delegation, what information would you need to give him so that he can implement it?

12. What important requirement is not yet met for your subdomain?