DT149G — Administration of UNIX-like systems
Laboratory Assignment:
Quem quaeritis?

Lennart Franked*

December 13, 2024

Contents

1 Introduction

In this laboratory assignment, you will run three separate Docker containers, each having its own IP address, and each container will host a domain using Bind9.

2 Aim

After completion of this assignment you will:

- Be able to setup and administer docker containers.
- Be able to correctly set up and administrate your own domain using BIND.
- Have basic understanding of DNSSEC.

*E-post: lennart.franked@miun.se.

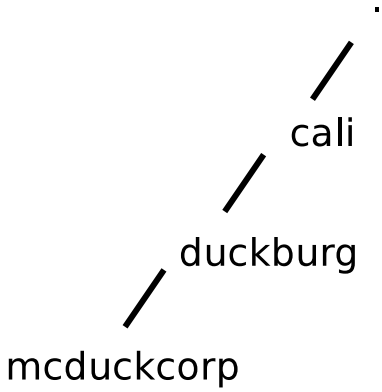

Figure 1: DNS name tree for Duckburg

3 Reading instructions

Before starting this assignment you should have read Nemeth et al., 2017, Chapter 16, 25. During the lab you will also need to consult the follwing documents Koch, 1999, Barr, 1996, Docker, 2024d, Docker, 2024b and Docker, 2024a.

4 Tasks — DNS

Perform the following tasks and document all the steps taken to complete them The lab scenario is that you will set up and administer the domains for

- cali.
- duckburg.cali.
- mcduckcorp.duckburg.cali.

See Figure fig. 1.

4.1 IPTables

Before we start setting up DNS and docker, you must update your firewall so that the ports for DNS are open.
Hence update your firewall policy with the following:

- Allow all incoming packets using transport protocol UDP and port 53 (DNS).
- Allow all incoming packets using transport protocol TCP and port 53 (DNS).

To answer in you report For this task, present the following in your report:

- Include a screenshot showing your new IPTable rules.

4.2 Installing Docker

Using the package manager, install Docker on your linux installation, see Docker, 2024c for installation instructions.
Consult the Docker CLI Cheat Sheet Docker, 2024a for command reference.
Once you have Docker installed, start by creating a new docker network with the help of the `docker network create` command[1]. See Docker, 2024b.

Your new docker network should be configured as follows:

- Name: dt149g<StudentID> (Replace <StudentID> with your student ID)
- Type: bridge
- Subnet: 10.YY.MM.0/28 (YY is the year you are born, MM is the month)

Once you have setup the Docker network, you can download the bind9 docker image from docker hubDocker, 2024d. Simplest way to achieve this is to use `docker pull ubuntu/bind9`
Finally, create a user on your system called `bind`, with a corresponding group `bind`.

To answer in your report For this task, answer the following questions in your report:

- Include a screenshot of your newly created docker network, by issuing the `docker network inspect <your docker network>`
- Include a screenshot showing that you have your docker images installed.

4.3 Setting up container environment

Create the following three folders (if /var/docker doesn't exist, you can create it).

- /var/docker/bind9_cali
- /var/docker/bind9_duckburg
- /var/docker/bind9_mcduckcorp

These folders will be used as the configuration root for each bind9-container.

Each container root should have the following subfolders

[1]It is not necessary to create a new network, however we would like to give static IP-addresses to the containers in our own given subnet.

- etc/bind
- var/lib/bind
- var/cache/bind

The Bind9 service that is run inside the container, will be run as the user `bind` (as defined by those created the image), however the bind-user inside the container might have a different UID than the bind-user you have created. This will cause problems with the file permissions for the mounted volumes. Therefore make a symbolic link of your */etc/passwd* and */etc/group* into each container configuration roots /etc-folder. We will then mount these files into the container, overriding the containers local passwd and group files.

Next, download the files `named.conf`, `named.conf.options` and `named.conf.local` from Moodle and place a copy of them in the `etc/bind` folder for each container-root. These are just the default files that are delivered with bind9 installation. Finally change the permissions on all folders, so that the owner is `bind`, group is `docker` and the permissions are set to 775

Multiple Docker containers are preferable managed through an orchestration tools, such as docker compose, or docker swarm for large scale use. However in this assignment we will focus more on the concept of containers, and therefore run them manually with the `docker(1)` command.
However, since there are numerous parameters that should be added, putting the docker-command in small bash-scripts will make it more manageable. Therefore create three files:

- `run_bind9_cali`
- `run_bind9_duckburg`
- `run_bind9_mcduckcorp`

Given below is a configuration example of how to start a docker-container that will be used for the cali-domain. Modify it accordingly, then create two more for the remaining two domains you will administer.
Consult table 1 for which IP to allocate to the containers.

Table 1: Addressing table for docker containers

Domain	Address
cali	10.YY.MM.2
duckburg	10.YY.MM.3
mcduckcorp	10.YY.MM.4

```
docker run \
  -d \
  --name=bind9_cali \
  --net=dt149g<StudentID> \
  --ip=10.YY.MM.2 \
```

```
  -e TZ=Europe/Stockholm \
  -v /var/docker/bind9_cali/etc/bind:/etc/bind \
  -v /var/docker/bind9_cali/etc/passwd:/etc/passwd \
  -v /var/docker/bind9_cali/etc/group:/etc/group \
  -v /var/docker/bind9_cali/var/lib/bind:/var/lib/bind \
  -v /var/docker/bind9_cali/var/cache/bind:/var/cache/bind \
  ubuntu/bind9
```

To answer in your report For this task, answer the following questions in your report:

- Discuss the advantages and disadvantages of running containers instead of installing Bind9 with apt.
- Include a screenshot of your folder hierarchy by issuing the command `tree /var/docker/`
- Describe the installation process of installing with Docker.
- Explain each line in the configuration example given for starting a bind9 container
- Include a screenshot showing your three bash scripts to start the three docker containers

4.4 Configuring Bind9

Perform the following for all your three domain configuration files.
For each bind9 instance you will need to modify three files in the /etc/bind folder.

- named.conf.local — Includes what zone (domain) this bind9 server will administer
- named.conf.options — Inlcudes configuration for the bind9-server.
- <yourdomain>.db — Zone-file, includes the resource records for that zone (domain).

For each bind container, perform the following.

1. Edit the named.conf.options on the Duckburg and Mcduckcorp containers and make them non-recursive.
2. Edit named.conf.options on the cali container and make it recursive.
3. Edit the <yourdomain>.db and add a SOA resource record. Consult Koch, 1999 and Barr, 1996 before setting the SOA values.
4. Next set up an NS record and a corresponding glue record. Allocate one of the IP-addresses you created in 4.3 to each subdomains NS
5. Edit the named.conf.local file and add the zone and link the zone-file to it.

In the mcduckcorp domain, you should also add the following resource records.

1. An A-record for your host IP (10.YY.MM.1) and point that to the hostname mail
2. A CNAME resource record pointing to mail and name it after your student-id.

Once all the configurations are finished. Start the containers using your script.
HINT: use docker logs <container> to verify that your container started

1. With the help of `dig (1)` test your DNS server by requesting the different resource records. If you get an `ANSWER SECTION` it works.
2. Confirm that the bind9 servers for cali and duckburg are non-recursive.

When your DNS-server is up and running, its time to configure the zones reverse file as well.
You should only create a revese-zone for your mcduckcorp-zone.
Create a zone-file for your IP-address range, for example if your address range is 10.2.3.0/24, create the file 3.2.10.in-addr.arpa.db

1. In this file start by adding the SOA-record
2. Next add the PTR records for each A-record you put in your *yourdomain*.db
3. Finally add this zone to bind in named.conf (or named.conf.local).
4. Restart your mcduckcorp_bind9 container.
5. check with `dig (1)` to make sure that your reverse zone are up and running.

To answer in your report For this task, answer the following questions in your report:

- Include a screenshot of your docker network, by issuing the `docker network inspect <your docker network>`
- Include a screenshot showing that all your three docker containers are running.
- Include screenshots of all your three domain configuration files and the reverse zone configuration file.
- Include a screenshot of the named.conf (or named.conf.local) showing what you had to include so that Bind will respond on DNS requests for the domains you just created.
- Include a screenshot showing that DIG can perform successful lookups on all three of your domains.

- Explain the need for the GLUE-record
- Discuss the SOA-values that you set. What do we need to consider when deciding these values?
- Explain how the reverse zone-file work.
- If you wanted to move your mcduckcorp-domain from your local DNS-tree out to the public tree. What would you have to do?

4.5 Securing DNS

In this section you will further secure your DNS-server by setting up DNSSEC. This security service that is based on an asymmetric encryption scheme and chain of trust, will among other things make your domain less susceptible to DNS cache poisoning attacks.[2]
You will have to setup DNSSEC for all of your containers. Since you will create a chain of trust from *.cali* to *.mcduckcorp*, where the parent domain should sign its child domains key, start by signing mcduckcorp.duckburg.cali and work upwards to cali. This way you will not have to redo the zone-signing for the parent when you create keys for the child.
For each domain, do the following: *HINT: For key generation and signing, open a bash shell into your container to access the commands*

1. Enable the `dnssec-validation` option to make your DNS server validate received signatures from other servers.
2. Generate your zone signing key (ZSK) and key signing keys(KSK) with the help of the `dnssec-keygen(8)`. What key size did you choose for the ZSK and KSK? Motivate your choice.
3. `dnssec-keygen(8)` generated two files per key pair, *.key* and *.private*. Explain the contents of these files.
4. Include your KSK and ZSK public keys in your domain zone file using the `$include <public key>` syntax to ensure that your public keys will be self-signed.
5. Now that you have your ZSK and KSK its time to sign your zone with the help of the `dnssec-signzone(8)` command.
6. Update your named.conf.local file to point to your signed domain zone file, and then restart your container.
7. When you signed your zone, a file named dsset-<yourdomain> was created that includes a DS-record that must be placed in the parent zone to ensure Chain-of-trust.
8. With the help of `dig`, verify that you can get the signed records.

It is sadly not possible to perform a proper DNSSEC validation due to your domain being offline.

[2] An attack that can send false DNS replies about your domain redirecting the traffic to another host.

To answer in you report For this task, present the following in your report:

- Include screenshots of how you generated your KSK and ZSK for all three domains.
- Include screenshots that shows your signed zone-files for all three domains.
- Explain the contents of the DS record and its purpose
- Explain the purpose of chain-of-trust in regards to DNSSEC.
- Include a screenshot showing the output of `dig` for all three domains when you verify that DNSSEC works.
- Finally include in your report the difficulties with key rollover for both ZSK and KSK and how to achieve this in a secure manner.

5 Examination

Hand in a report containing all your solutions to the questions in Section 4 *Remember that you must include references to the given reading instructions, alternatively to the laboratory work you have done*

References

Barr, D. (1996). *Common DNS Operational and Configuration Errors* (rfc No. 1912). IETF. http://tools.ietf.org/rfc/rfc1912.txt

Docker. (2024a). *Docker cli cheat sheet*. Retrieved October 25, 2024, from https://docs.docker.com/get-started/docker_cheatsheet.pdf

Docker. (2024b). *Docker network create*. Retrieved October 25, 2024, from https://docs.docker.com/engine/reference/commandline/network_create/

Docker. (2024c). *Install docker engine*. Retrieved October 25, 2024, from https://docs.docker.com/engine/install/

Docker. (2024d). *Ubuntu/bind9*. Retrieved October 25, 2024, from https://hub.docker.com/r/ubuntu/bind9

Koch, P. (1999). *Recommendations for dns soa values*. Retrieved October 25, 2024, from http://www.ripe.net/ripe/docs/ripe-203

Nemeth, E., Snyder, G., Hein, T. R., Whaley, B., & Mackin, D. (2017). *Unix and linux system administration handbook* (Fifth edition.). Addison-Wesley/Pearson.