

## DNS

*This lab will help you learn about DNS servers by guiding you through the steps required to setup the BIND software. Your system will be a DNS master server for a fictitious zone, providing DNS information for fictitious machines that reside on a fictitious network. Perform the indicated steps, and write the answers to questions in the appropriate spaces on this hand-out. Turn-in one filled in handout per team by the beginning of class next week. Be sure to include the names of all team members.*

### Configure the System

*Bring the system up single-user and perform the initial network configuration*

- Step 1. Boot your system into single user mode.
- Step 2. Configure your system with the IP address 10.0.0.N/24, where N is your system number.
- Step 3. Bring up the **lo** and **eth0** interfaces. Try to ping a host in the internet (if the internet gateway is running).
- Step 4. Remove the DNS resolver configuration file **/etc/resolv.conf**. We will set this up again later.

### Install the BIND DNS software

*To run a name server, you need the BIND software which includes the named daemon, utilities and some other files. In this section, you will obtain and install the BIND package, and copy some sample template files into place to further customize.*

- Step 5. The first step in setting up your system to act as a DNS server is to obtain the BIND software if it is not available on your system. If you do not have the **named** daemon **/usr/sbin/named**, then your system does not have BIND installed. You will have to obtain and install the BIND package; it can be found on a Red Hat installation CD, or at any of the Red Hat software mirrors. There is a copy available via ftp on the instructor's system, along with some template files that will assist you in setting up your zone files. If you do not need BIND nor want the configuration files, then skip to the next section.
- Step 6. Use the **ftp** command to connect to 10.0.0.200. Once connected, login in anonymously (user: **ftp** or **anonymous**). Get the file **bind.tgz**, which is a **gzip**-compressed, **tar** archive file. It contains the BIND software, and some configuration and template files you will use. You will see some help after you login to the server.
- Step 7. Once **bind.tgz** is downloaded, exit ftp, and run the command below to decompress and un-archive the contents:

```
# tar -xvzf bind.tgz
```

You now have a new directory named BIND, which contains the BIND software in Red Hat Package Manager format (RPM), and also some template files I have created for you.

- Step 8. Change directories into the BIND directory, and use **rpm** to install the RPMs located in the BIND directory

```
# cd BIND && rpm -ivh *.rpm
```

- Step 9. The **named** daemon needs a configuration file. I have provided a basic one for you. Move the named configuration file into its proper place:

```
# mv etc/named.conf /etc
```

- Step 10. The **named** daemon also needs a set of zone files and the hints file. Since you will be setting up a master name server, you need to construct the zone files for your zone. The hints file is standard and can generally be copied from almost any system (or well known web sites). Sample templates and a hints file **named.ca** are included for you. Create the standard directory where the zone files live and move the entire **zone-templates** directory, and the **named.ca** file into the **/var/named** directory. You will customize these shortly.

```
# mkdir -p /var/named
# mv zone-templates named.ca /var/named
```

- Step 11. Now you are ready to configure **named**. There are two parts to configuring BIND: the **named** configuration file, and the zone files themselves. The named configuration file is **/etc/named.conf**. The zone files live in the directory that

is specified in the **options** section of the file **/etc/named.conf**. Examine the **named.conf** configuration file (which you installed earlier) to validate the name of the directory used to store the zone files.

**Q1.** What is the name of the directory used to store your zone files? \_\_\_\_\_

**Q2.** What section in the **named.conf** file is this information located in? \_\_\_\_\_

**Q3.** What keyword is used to specify this section? \_\_\_\_\_

Step 12. Change directories now into the zone file directory you discovered above, and look at the contents of the directory. You should see a file named **named.ca** (the root-servers hints file) and a sub-directory **zone-templates** that contains the class template files you will soon copy and modify. If the files **named.local** or **localhost.zone** exists, ignore them. These are Red Hat's default zone files for caching-only nameserver configuration. You will not be using them.

### Setting up the zone files

*It is now time to setup the zone files for the zone.*

Step 13. It is now time to create the zone files for your zone. Before you can do this, you need a domain name assigned to you, and need to select a sub-domain name. The table at the end of this lab specifies the domain information you will use for your system. Look up the information that corresponds to your system. It will include your sub-domain, domain, and network address. You will use the imaginary 192.168.N.0/24 network where there are imaginary hosts: **host1**, **host2**, and **host3**, with host numbers 1, 2, and 3 respectively. While you obviously will not be able to connect to or **ping** these fictitious hosts, the name server does not care, and will function perfectly well. And you will be able to use **nslookup** or **dig** to lookup DNS information being provided by your DNS server. Make sure you know the answers to the questions below – incorrect information will give you trouble configuring your DNS server.

**Q4.** What is the name of your sub-domain? \_\_\_\_\_

**Q5.** What is the name of your domain? \_\_\_\_\_

**Q6.** What are the FQDN's for your three imaginary hosts? \_\_\_\_\_

Step 14. Each zone needs a forward file and a reverse file. Create your zone's forward zone file by copying the template file **zone-templates/forward** to **/var/named/db.subdomain**, where *subdomain* is the sub-domain you identified above in question Q4 (i.e. not including the domain part; it does not matter, but makes your file names shorter). The zone files can be named anything you want – but using a standard convention makes it easier for you and others to comprehend.

Step 15. Edit your **db.subdomain** forward zone file, replacing the items as instructed at the top of the file. Remember, the forward zone file must minimally contain the **SOA**, **NS**, and **A** records for the zone. The RRs in the template zone file are generally not using any of the shortcut notations (i.e. @, default domains, etc.) so that you can follow the fields more easily. Later, you can change the zone files to use the standard shortcuts. When you are done editing the file, save it. Here is some more help:

1. Change all **xxZONExx** placeholders to the FQDN you determined above (don't erase the trailing dot). Hint: in **vi**, type **/xxZONExx** (search for pattern), and hit Enter. Then type **cw** (change word), type in the zone name and hit Escape. To change the next placeholder, type **n** (next) followed by a **.** (period) to repeat the change. Do the same for the remaining **xxZONExx** placeholders). The **vi** commands above will leave the trailing **.** in the names – make sure you did not delete it!
2. The master nameserver for this zone will be **ns1.xxZONExx**. This is just the name you will give to your server - other systems would use this name query your zone information.
3. The email address should be the email address of the zone's administrator – the **@** symbol in the email address is replaced by a dot.
4. The **N** is the number on your CPU – replace all **xxNxx** placeholders with this number.

Step 16. Now a reverse zone file must be created to map IP addresses back to hostnames. Copy the template file **zone-templates/reverse** to **/var/named/db.netaddr** where *netaddr* is only the network part of the IP address of the imaginary 192.168.N.0/24 network. Edit the reverse zone file just created, again replacing the placeholder items.

Step 17. Now the reverse zone file for the 127.0.0.0/8 loopback network needs to be created. Copy the template file **zone-templates/reverse-loopback** to **/var/named/db.127.0.0**. Edit the file and make the required changes.

**Q7.** Why do we not need a forward zone file for the loopback interface? \_\_\_\_\_

Step 18. You should already have a root server hints file – it is named **named.ca**. Examine the file and look at the root servers for the entire world. You do not need to change this file.

**Q8.** What is the name and IP address of the fourth root server? \_\_\_\_\_

### Configure the named configuration file

*The named.conf configuration file needs to be configured with options and configurations parameters for named, and needs to tell named where to find the zone files and which zones it serves*

Step 19. Now that the zone files are ready, edit **/etc/named.conf** and make the changes mentioned at the top of the file. You will end up with four zones sections: one for the root servers (a hints zone), and three configured as master servers. Each zone section lists the name of the zone file (relative to the directory specified at the top of the **named.conf** file) that contains the zone data for that particular zone. Make sure the zone names in double quotes do not contain trailing dot. The **file** keyword specifies the name of your zone file in double quotes – the name must exactly match the name of the zone file.

### Start the named daemon

*The named daemon needs to be started to enable DNS services.*

Step 20. Normally **named** runs in the background as a daemon, and is controlled once started using the **rndc** program. But for debugging purposes, we will start **named** in the foreground. This will save a lot of time and grief. To do this, you will have to start **named** on one TTY while you use another TTY for commands. You can either use X Windows and two **xterm** sessions, or you can use the virtual consoles (shift between them with Ctrl-Alt-FN, where FN is a function key such as F1-F7). I would suggest for now using the virtual consoles, as X Windows relies on networking and changing the networking infrastructure from under X Windows is sometimes problematic. Either way, use one terminal to start and leave **named** running, and another to run various commands. This allows you to go back and forth between the two to observe the results of your actions. Make sure you have two terminals available.

Step 21. Start **named** with the command below. The **-u** is necessary so that **named** is run as user **named** because it will change its UID when it starts, lowering its privileges; it will then not have permission to write its lock file into the directory **/var/run/named**.

```
# named -g -u named
```

**Q9.** What does the **-g** option specify? \_\_\_\_\_

Step 22. Examine the output and look for any error messages. You can ignore the message about logging being ignored because **named** was started with **-g**. Any other messages between that error and the last line indicating that **named** is running are errors that need to be corrected. Start with the first one, and resolve the trouble. Each error will contain the offending file name and the line number. Kill **named** with Ctrl-C, resolve the problem, and then restart **named** again. Rinse, lather, and repeat until all errors are resolved.

### Test your name server

*Use the DNS query tools to test the DNS server.*

Step 23. With **named** now running, it is time test your DNS server! Switch terminals and use either **nslookup** or **dig** to resolve the host **host1.FQDN** where **FQDN** is your fully qualified domain name. Try to also resolve the other hosts **host2**, **host3**, and **ns1** using fully qualified names.

Step 24. Now try to resolve a relative host name (i.e. one that is not fully qualified). Try the command:

```
# nslookup ns1
```

**Q10.** What happens? \_\_\_\_\_

Step 25. To allow simple host names to resolve, the resolver needs a default domain to append names that are not fully qualified. Create **/etc/resolv.conf** and add your domain name after a **domain** keyword. Do not add a trailing dot to the name. The entry in **/etc/resolv.conf** should look like:

```
domain FQDN
```

Step 26. Retry the **nslookup** command above. This time it should succeed, since the resolver is appending the default domain name (listed after the **domain** keyword in **/etc/resolv.conf**) to each simple host name (those that are not fully qualified). Try resolving all of the following names: **ns1**, **host1**, **host2**, and **host3**, using the simple names and the fully qualified names for each.

Step 27. Use **nslookup** or **dig** to resolve IP addresses into hostnames. Try each of the IP addresses you have configured for your fictitious 192.168.N.0/24 network

**Q11.** Explain why your name server cannot resolve a host name given the IP address of your eth0 interface. \_\_\_\_\_

\_\_\_\_\_

Step 28. Resolve the IP addresses for **localhost.FQDN**. (trailing dot is included, since you want a FQDN).

**Q12.** What is the IP address resolved? \_\_\_\_\_

Step 29. Now try for **localhost**. (trailing dot included). You will find that resolution of **localhost** fails – it times out. This occurs because your DNS server does not know about a zone called **localhost.**, so it goes to a root-server to resolve. There is no **localhost** TLD under the root (you can validate this with **nslookup** / **dig**). Once DNS is running, all names used by the system are translated into, and must be, fully qualified names. And **localhost** is not a valid fully qualified name. To reduce the burden on the root-servers, your system should resolve the zone **localhost** locally.

Step 30. Copy the file **zone-templates/db.localhost** to **/var/named/db.localhost**. This is a zone file to locally manage the **localhost** zone. Examine the file and try to understand its contents. Notice in particular how the **\$ORIGIN** statement defines the zone name and how the **@** symbol is used as its shorthand. This file is ready to use. You just need to edit **/etc/named.conf** and add the zone statement that tells **named** to use it.

Step 31. Edit **/etc/named.conf** and copy/paste one of your existing zone sections. Replace the zone name with **localhost** (no trailing dot in the double quoted names) and the file name with **db.localhost**. Save the file, kill **named** and restart it. Try to resolve **localhost** again – this time, it should work. Try to resolve the loopback IP address as well.

Step 32. If you can ping a host by IP address on the internet, your name server should now be able to respond to all of your queries. It will forward any non-cached queries to a root name server for resolution. Try to resolve an external host using **nslookup**.

Step 33. Finally, test that another system can use your name server to resolve names in your domain. Ask someone to use **nslookup** / **dig** to resolve a name in your zone. They will need to set the default name server for **nslookup** / **dig** to use the IP address of your name server.

**Q13.** Can they resolve your **host1.FQDN**? \_\_\_\_\_

**Q14.** What is required for them to resolve short names (i.e. host2)? \_\_\_\_\_

\_\_\_\_\_

Step 34. Add two **CNAME** RRs for one or more hosts in your zone that fit in with the theme of your zone (i.e. worms in the slimy.bugs.org zone, or slingblade in the drama.flix.org zone). Recall that CNAMEs are aliases for the real name of a host. Kill and restart the nameserver to have the names take affect.

Step 35. Use your favorite DNS tool to resolve the CNAME just added.

Step 36. Add one or more **TXT** RRs that includes your name and other info you want. Be sure the syntax is correct, or it will not work. Restart **named**, and use your favorite DNS tool to find these newly added records.

## Domain Name Table

CPU Number	Sub-domain	Domain	Network Address
1	creepy	bugs.org	192.168.1.0/24
2	thriller	flix.org	192.168.2.0/24
3	crawly	bugs.org	192.168.3.0/24
4	comedy	flix.org	192.168.4.0/24
5	icky	bugs.org	192.168.5.0/24
6	musical	flix.org	192.168.6.0/24
7	slimy	bugs.org	192.168.7.0/24
8	drama	flix.org	192.168.8.0/24
9	wiggly	bugs.org	192.168.9.0/24
10	romance	flix.org	192.168.10.0/24
11	big	bugs.org	192.168.11.0/24
12	adventure	flix.org	192.168.12.0/24
13	water	bugs.org	192.168.13.0/24
14	action	flix.org	192.168.14.0/24
15	biting	bugs.org	192.168.15.0/24
16	crime	flix.org	192.168.16.0/24
17	stinging	bugs.org	192.168.17.0/24
18	western	flix.org	192.168.18.0/24
19	flying	bugs.org	192.168.19.0/24
20	war	flix.org	192.168.20.0/24
200	horror	flix.org	192.168.200.0/24