

## Setting a DNS Server

### Objective

To practice and understand how DNS works and how a DNS server is configured on a Unix system.

### Background

DNS servers are used on the Internet to provide mapping between domain names (which are meaningful to people) to IP numbers (which are meaningful to routers). Besides making it easier to remember addresses, the DNS naming scheme allows to build a logical hierarchy of addresses, which is independent of the underlying implementation (routing depends on IP and you cannot use whatever IPs you want). Each domain (an example of a domain is **ucf.edu** or **cs.ucf.edu**, but not **eola.cs.ucf.edu**) has at least one associated name server, which is the contact point for resolving DNS-to-IP mappings for that domain. The knowledge about whether **x.ucf.edu** is a domain or a machine is contained in the **ucf.edu** domain server. If it is a machine (like **pegasus.ucf.edu**), the **ucf.edu** domain server will be its IP. If it is a domain (like **cs.ucf.edu**), the **ucf.edu** domain server will tell you what is the IP of the name server, which is responsible for this sub-domain. There is also a scheme for reverse mapping from IP addresses back to DNS records, which is called reverse name resolution. In this scheme, the DNS name of an ip address of the form **A.B.C.D** can be found by looking at the **D.C.B.A.in-addr.arpa** address.

The main configuration file of the name server is `/etc/named.conf`. Here is an example of what it might look like:

```
options {
    directory "/var/named";
};

zone "." {
    type hint;
    file "named.root";
};

zone "0.0.127.in-addr.arpa" {
    type master;
    file "db.127.0.0";
};

zone "netlab.cs.ucf.edu" {
    type master;
    file "db.netlab";
};

zone "100.0.10.in-addr.arpa" {
    type master;
    file "db.10.0.100";
};
```

The directory option describes where the zone description files are located. The options section is followed by a series of dns zones declarations. In the above example, the DNS server is the master server for the **netlab.cs.ucf.edu**, **10.0.10.in-addr.arpa** and **0.0.127.in-addr.arpa** domains. The last two domains are used for reverse DNS resolution. The "." zone contains information about the main DNS servers on the Internet, and is used to resolve all the other address queries.

Here is an example of what a zone file looks like (this is the db.netlab file) the following example. Everything after a ";" character is a command. The numbers following the first line are just times (in seconds) for different parameters of the described data.

```

;
; BIND data file for netlab.cs.ucf.edu
;
@      IN      SOA      netlab.cs.ucf.edu. harper.longwood.cs.ucf.edu. (
                                1      ; Serial
                                604800 ; Refresh
                                86400  ; Retry
                                2419200 ; Expire
                                604800 ) ; Default TTL

      IN      NS       server1.netlab.cs.ucf.edu.

server1.netlab.cs.ucf.edu. IN    A   10.0.100.10
rsm.netlab.cs.ucf.edu.     IN    A   10.0.100.100
router1.netlab.cs.ucf.edu. IN    A   10.0.100.1
router2.netlab.cs.ucf.edu. IN    A   10.0.100.2
pc1.netlab.cs.ucf.edu.     IN    A   10.0.10.10
sun1.netlab.cs.ucf.edu.    IN    A   10.0.10.20

```

Here is an example of how the reverse DNS zone might look like (this is the db.10.0.100 file).

```

@      IN      SOA      100.0.10.in-addr.arpa. harper.longwood.cs.ucf.edu. (
                                1      ; Serial
                                604800 ; Refresh
                                86400  ; Retry
                                2419200 ; Expire
                                604800 ) ; Default TTL

      IN      NS       server1.netlab.cs.ucf.edu.

1.100.10.in-addr.arpa.     IN    PTR  router1.netlab.cs.ucf.edu.
2.100.10.in-addr.arpa.     IN    PTR  router2.netlab.cs.ucf.edu.
10.100.10.in-addr.arpa.    IN    PTR  server1.netlab.cs.ucf.edu.
100.100.10.in-addr.arpa.   IN    PTR  rsm.netlab.cs.ucf.edu.

```

Note that every address definition in both files ends with a period. Another thing to be careful about is that DNS-to-IP records are done with the "IN A" construct, while IP-to-DNS records are done with the "IN PTR" construct.

## Preparation

In this exercise, you will be asked to setup a fictitious DNS server on the Unix workstation in front of you. Pico is recommended as the text editor, and you must type the configuration as exactly as they are required below.

## Procedure

1. Login to your workstation as root.
2. Create a file `/etc/named.conf` with the following contents:

```
options {
    directory "/var/named";
};

zone "netlab.cs.ucf.edu" in {
    type master;
    file "db.netlab";
};

zone "xx.0.10.in-addr.arpa" in {
    type master;
    file "db.10.0.xx";
};
```

Where xx is your segment number followed by 0. For example, if you are using computer SUN1, then xx should be 10.

3. Create the directory `/var/named`, this is the working directory of dns server.
4. Create the file `/var/named/db.netlab` with the following contents:

```
@ IN SOA netlab.cs.ucf.edu. harper.longwood.cs.ucf.edu. (
    1      ; Serial
    604800 ; Refresh
    86400  ; Retry
    2419200 ; Expire
    604800 ) ; Default TTL
IN      NS  fake.netlab.cs.ucf.edu.

fake.netlab.cs.ucf.edu. IN  A  10.0.xx.20
```

Note: SOA is S-number zero-A  
replacing xx with the appropriate numbers for your IP.

5. Create the file `/var/named/db.10.0.xx` with the following contents:

```
@ IN SOA xx.0.10.in-addr.arpa. harper.longwood.cs.ucf.edu. (
                                1 ; Serial
                                604800 ; Refresh
                                86400 ; Retry
                                2419200 ; Expire
                                604800 ) ; Default TTL
IN NS fake.netlab.cs.ucf.edu.
20.xx.10.in-addr.arpa. IN PTR fake.netlab.cs.ucf.edu.
```

replacing `xx` with the appropriate numbers for your IP.

6. Make the `/etc/resolv.conf` file look like this:

```
domain netlab.cs.ucf.edu
search netlab.cs.ucf.edu
nameserver 127.0.0.1
```

7. Open file `/etc/nsswitch.conf`, find the line

```
hosts: files
```

Replace this line with the following:

```
hosts: files dns
```

Save file and exit.

8. Start the name server with the following command:

```
# in.named
```

9. Open another terminal window, try the DNS server by typing:

```
# telnet fake
```

If everything is ok, you should be able to telnet to your own workstation.

10. If there are some problems with our DNS server, you need following the procedure below:

- a. Open a terminal window, run the command

```
ps -A | grep "in.named"
```

This command will give you process number(s) of any running `in.named`

- b. Run following command to terminate all existing `in.named`.

```
kill -9 <all process number(s) you get from step a, seperated by space>
```

- c. Check your configuration files, and make necessary corrections.

- d. Restart `in.named`, and try "telnet fake" again.