1 Detailed DNS configuration

1.1 Introduction

By default Bind (DNS server software) which uses executables with name '`named`' or uses service name '`named`' uses chroot to folder '`/var/named/chroot`' after service has successfully started. Hence all the files that Bind uses must be present inside some sub-folder of `/var/named/chroot` for Bind to work.

Before bind does chroot to `/var/named/chroot` it will use the files directly with respect to actual `/` and later on it will use the same path to access same files inside `/var/named/chroot`. For example, before bind does chroot it will access `/etc/named.conf` which is actual `/etc/named.conf`. After chroot bind will continue to use `/etc/named.conf` but now as per filesystem bind would end up using `/var/named/chroot/etc/named.conf`.

Hence for things to work we need two copies of every file that bind uses, so that bind can find the file both before and after chroot. For this copying file is bad idea, as if after copying we change one file, we would have to change other as well which is very impractical in long run. Also this may lead to bugs which are very hard to find or troubleshoot.

To ensure that bind gets two copies of each files without actually requiring to maintain two different copies we use symbolic links. Hence we create original files in `/var/named/chroot/<some sub-folder>` and create symbolic links in actual `/`. For example we would have actual `named.conf` at `/var/named/chroot/etc/named.conf` and create symbolic link using absolute path at `/etc/named.conf` using command:

```
ln -s /var/named/chroot/etc/named.conf /etc/named.conf
```

Please also note that we cant do the same thing in other order. For example we cannot have actual file in `/etc/named.conf` and symbolic link in `/var/named/chroot/etc` as after chroot that symbolic link would become invalid.

In good server configurations `/var` and `/etc` are on separate partitions. I am avoiding detailed explanation of why that is done here, but that implies we cannot use hard-links to link `/etc/named.conf` and `/var/named/chroot/etc/named.conf`.

Thus the only good way of configuring DNS is to have original files in `/var/named/chroot` and to have symbolic links that using absolute path in /etc/ poiting to same files in `/var/named/chroot/etc`.

Although the above explanation is given for `/etc` configuration files, the same applies to `/var/named/chroot/var/named` zone files. Hence original zone files should go inside folder `/var/named/chroot/var/named` with proper permissions and owner etc. and symbolic link to each of these zone files using absolute paths must be created in `/var/named`.

1.2 File structure

By default after installation of DNS packages (Already done in lab machines) following is the output of running tree command in `/var/named/chroot` folder.

```
.
|-- dev
|   |-- null
|   |-- random
|   `-- zero
|-- etc
|   |-- localtime
|   `-- rndc.key
`-- var
    |-- log
    |-- named
    |   |-- data
    |   `-- slaves
    |-- run
    |   `-- named
    `-- tmp
```

If you want to ensure that previous DNS related configuration on the system does not affect your work / experiments then you can delete all the files/folders inside `/var/named/chroot`, except the ones listed above.

Different types of DNS configuration files can be categorized broadly into two different categories:

1. Config files
2. Zone files

Config files usually go in folder `/etc` and zone files in `/var/named`. We can always specify different path for zone files in /etc/named.conf file by using `directory` configuration option.

1.3 Hello World DNS

In this subsection we will try to create minimalist configuration for DNS using defaults wherever possible. Then we will add more and more configuration options / zones to this minimum DNS server and try to go as close as possible to sample configuration given in actual lab handout.

Minimal `/var/named/chroot/etc/named.conf` can contain

```
options
{
    directory "/var/named";
    //empty, going to use all defaults for rest
};

include "/etc/named.root.hints";    //Need location of
                                    // . or root servers
                                    //for recursion
include "/etc/named.rfc1912.zones"; //A proper DNS
                                    //must have these zones
```

You must do following things after adding above contents to `/var/named/chroot/etc/named.conf` file:

1. Create symbolic link in /etc/ using: `ln -s /var/named/chroot/etc/named.conf /etc/named.conf` command
2. Copy sample `named.root.hints` file that comes with Bind to `/var/named/chroot/etc` and then create symbolic link of this file in /etc. Please note that since the file has been included in `/etc/named.conf` it will also go in `/etc` folders. The files that are not included but are referred like zone files go inside `/var/named` folders.

   You can use linux locate command to search for `named.root.hints` file.
3. Copy sample `named.rfc1912.zones` file that comes with Bind to `/var/named/chroot/etc` and then create symbolic link of this file in /etc. The reasons and method being same as used for `named.root.hints` file.
4. Use command '`named-checkconf /etc/named.conf`' to check the current configuration file and ensure that everything so far is fine. If everything so far is correct then '`named-checkconf`' will not show any output (error messages).

Now if we look inside contents of `named.root.hints` file that we have used above, we find lines

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

Here one zone with named dot ('.') which is parent most zone possible is being defined and it is mentioned that it is of type hint. It is also mentioned that hints related to this zone, which basically means list of dot nameservers is mentioned in file '`named.root`'

So we must have file with name '`named.root`' in '`/var/named/named.root`' files. Note that here `named.root` is being referred and is not included. Hence this file would go in `/var/named` and not in `/etc`.

Fortunately for us default Bind server comes with file `named.root` and we only need to copy it to `/var/named/chroot/var/named` folder. Then we need to create symbolic link of file from `/var/named/chroot/var/named` folder. to `/var/named` folder. The exact command would be:

```
ln -s /var/named/chroot/var/named/named.root /var/named/named.root
```

Similarly if we look inside `named.rfc1912.zones` file we can see that it refers to files:

1. `localdomain.zone`
2. `localhost.zone`
3. `named.local`
4. `named.ip6.local`
5. `named.broadcast`
6. `named.zero`

Now we have to use default files that come with Bind with above names by copying them to folder `/var/named/chroot/var/named` and creating their corresponding symbolic links inside folder `/var/named`.

If you find more than one file with same name on system you can copy any one of them to `/var/named/chroot/var/named` without worrying about which file is correct.

After doing all this if you run `tree` command in folder `/var/named`, you should see output like:

```
.
|-- chroot
|   |-- dev
|   |   |-- null
|   |   |-- random
|   |   `-- zero
|   |-- etc
|   |   |-- localtime
|   |   |-- named.conf
|   |   |-- named.rfc1912.zones
|   |   |-- named.root.hints
|   |   `-- rndc.key
|   `-- var
|       |-- log
|       |-- named
|       |   |-- data
|       |   |-- localdomain.zone
|       |   |-- localhost.zone
|       |   |-- named.broadcast
|       |   |-- named.ip6.local
|       |   |-- named.local
|       |   |-- named.root
|       |   |-- named.zero
|       |   `-- slaves
|       |-- run
|       |   `-- named
|       `-- tmp
|-- data
|-- localdomain.zone -> /var/named/chroot/var/named/localdomain.zone
|-- localhost.zone -> /var/named/chroot/var/named/localhost.zone
|-- named.broadcast -> /var/named/chroot/var/named/named.broadcast
|-- named.ip6.local -> /var/named/chroot/var/named/named.ip6.local
|-- named.local -> /var/named/chroot/var/named/named.local
|-- named.root -> /var/named/chroot/var/named/named.root
|-- named.zero -> /var/named/chroot/var/named/named.zero
`-- slaves
```

Please note that important information that symbolic link to named.conf, named.root.hints and named.rfc1912.zones file exist in /etc cannot be verified by using above output. For verifying that use command
`ls -l /etc/named.* | grep -o '\/.*$'`

which should give following output:

```
/etc/named.conf -> /var/named/chroot/etc/named.conf
/etc/named.rfc1912.zones -> /var/named/chroot/etc/named.rfc1912.zones
/etc/named.root.hints -> /var/named/chroot/etc/named.root.hints
```

As one final step use '`chown named:named *`' in both '`/var/named`' and '`/var/named/chroot/var/named` folder to give named permission to read files that we have copied just now from root user.

After this use '`service named start`' to check DNS configuration done so far and to verify that we can start Bind with this much configuration. If you are not able to start Bind then verify all the above things mentioned in handout very carefully.

It can be noted that this is considerably more than minimal Bind configuration. But if we do minimal configuration then Bind may not be able to handle recursive queries because of absence of . zone or it may not confirm to RFC1912 as it would miss some zones which every DNS must have as per RFC. Hence the current configuration is minimal correct DNS configuration.

Please also note that if you try commands like: `nslookup www.google.com 127.0.0.1` then your DNS will not be able to resolve the query and will give 'connection timed out' error after waiting for few seconds.

This is because all machines in LAN cannot by default send UDP packets outside campus. Hence your DNS server is not able to query dot ('.') servers for address of .com servers to resolve above query and hence nslookup fails. You can verify that your DNS server is trying to query dot('.') servers using wireshark.

A simple resolution of this problem is by adding following option in /etc/named.conf:

```
     forwarders  192.168.36.222; 192.168.36.204; ;
     forward only;
```

This means that DNS server should forward the queries that it receives to 192.168.36.222 or 192.168.36.204. Since port number has not been defined the queries would be sent to other DNS servers on default DNS port. The '`forward only`' option indicates that DNS should always forward query and never try to do contact other DNS directly.

The complete /etc/named.conf config file now has (excluding comments)

```
options
{
    directory "/var/named";
    forwarders { 192.168.36.222; 192.168.36.204; };
    forward only;
};

include "/etc/named.root.hints";
include "/etc/named.rfc1912.zones";
```

Now use '`service named reload`' and again try `nslookup` or `dig` for www.google.co.in etc. and the DNS server should now be able to resolve queries. You can verify that current setup is working because DNS is forwarding queries to 222 or 204 DNS servers.

Prefer to use `dig` command instead of `nslookup` for all your DNS queries. `dig` is way more powerful than `nslookup`

1.4 Our own zone

Now that DNS is up and running including resolution of outside zones, the next best thing to do is to create our own zones. For purpose of this handout we will create test.iiit.ac.in zone. Please note that we can create any zone, even 'google.com' and anyone who uses our DNS server when they query for google.com, the replies would depend on our configuration and not actual google.com servers. Hence it is very important to use only trusted DNS servers to avoid DNS spoofing based attacks.

To create zone we need to append following lines at end of `/etc/named.conf` file:

```
zone "test.iiit.ac.in."
{
        type master;
        file "test.iiit.ac.in.forward";
};
```

This means that we are going to create a new zone with name 'test.iiit.ac.in' and that information of that zone is present in file `test.iiit.ac.in.forward`. As should be clear by now the file `test.iiit.ac.in.forward` must be created / present in `/var/named` folders for DNS to work.

Contents of `test.iiit.ac.in.forward` file can be

```
$TTL 3600
@ SOA ns.test.iiit.ac.in. root.test.iiit.ac.in. (1 15m 5m 30d 1h)
        NS ns.test.iiit.ac.in.
        A 10.3.3.157
ns              IN      A       10.3.3.157
lab320pc1       IN      A       10.3.3.157
```

Here, meaning of various options in SOA field was explained in DNS lecture and is also mentioned in slides on DNS lecture. Basically we have created a new zone with name test.iiit.ac.in. Nameserver for that zone is ns.test.iiit.ac.in. IP address of test.iiit.ac.in zone itself is 10.3.3.157. There are two hosts in this zone ns and lab320pc1. Both of them also have address 10.3.3.157.

To verify whether zone file is correct or not use command:

```
named-checkzone test.iiit.ac.in test.iiit.ac.in.forward
```

If you do not see any error being output by above command then zone file is created correctly.

Now create symbolic link to above zone file in `/var/named` folder and also ownership of file in both places (actual file and link) to named:named. Now try `service named restart` command and the command should be able to stop and start DNS server without any problem.

You can now check working of zone by querying your server using command:

```
dig @127.0.0.1 lab320pc1.test.iiit.ac.in
```

1.5 What is left

The most important parts of DNS configuration have been explained here in detail. To see list all options supported by Bind use '`man named.conf`' command. To actually understand meaning / significance of various things specified in man page refer to bind Administrative manuals hosted at http://www.bind9.net/manuals

In lab handout I have given very advanced configuration which uses views, detailed logging, restrictions number of queries of each type, explains bind how to do load-balancing if multiple addresses are assigned for same name, listen on which IPs and which ports, allow recursive queries or not, what server information should be returned, etc. It takes considerable time and experience to understand all those options and what is the best value for them in given scenario.

As part of this lab you are not required to understand any of those extra options. You can always read about them from bind manual.

In lab handout we have also defined reverse zones. There is nothing special about reverse zones other than that they have PTR records instead of A or AAAA records. Rest everything about reverse records remains same and is easy to understand.

For any further queries you can email me directly or contact me when I am in my office. I hope this document improves your basic understanding of DNS configuration and enables you to learn more advanced topics by yourself.