

CSIS0234B Computer and Data Communication (Class B)

Tutorial 5

Setting up local DNS servers

In this tutorial, you will setup DNS servers for a few computers, in order to gain a deep understanding of the operation of DNS.

Tasks

The computer has a copy of bind. The name of the program is called named, and can be started or restarted by typing `/etc/init.d/named restart` on the command prompt. Currently it is configured as a “caching server”, i.e., it knows nothing more than `localhost` and `127.0.0.1`, and rely on the root servers to find the DNS address (and then cache it). You will modify it to serve several computers.

To complete all the tasks, you need 4 computers. So work in a group of 4. You must also know the IP addresses of the computers, which is `172.16.9.x` where `x` is the computer number as displayed on the board. You also need to use the root account to set it up. The password will be announced at the beginning of the tutorial. You will also find the appendix of the tutorial reading material useful.

1. (For the first 3 computers) Login root, and create a shell. Change to the `/etc` directory. Find the file `resolv.conf`. You will notice that a set of name (DNS) servers are listed there. Copy them down, and change it so that it contains only 1 nameserver line, for `127.0.0.1`, and no `search` line. The first thing that you should notice is that your computer can resolve only “localhost”, and can do the reverse query of it. All external accesses are blocked. It is because our lab has a firewall which blocks all DNS requests, except from the nameservers you’ve just copied.
2. (For the first 3 computers) Edit the file `named.conf` in the `/etc` directory by adding a global `forwarders` line, so that it forwards to the nameserver you have copied. Restart the DNS server, and check that you can reach the rest of the world (e.g., try to resolve `www.csis.hku.hk` in the new setting). Now you have a fully functional caching server. Now turn off the firewall (using `/etc/init.d/ipchains stop`), and try to use the fourth computer to access it (using, e.g., `dig @172.16.9.x www.csis.hku.hk`).
3. Now is time to get some real work done. For the first computer, continue to edit the `named.conf` file to add a new master zone `netlab.` in it. The domain will be visible only from the 4 computers we use, so it is okay to use a top-level domain (without getting approval from the root domain). By copying and modifying the `localhost.zone` file, create another zone file `netlab.zone` in the `/var/named/` directory. Add A entries to it so that it will translate the IP addresses of the 4 computers as `father`, `mother`, `son` and `daughter` respectively. Restart the DNS server. Modify the `resolv.conf` file of the computer 2 (`mother`), to check that it works to translate names for the 4 computers.
4. Now get the reverse lookup right: add a new zone `9.16.172.in-addr.arpa` in the `named.conf` configuration file. By copying and modifying the `named.local` file, create another zone file `named.netlab` in the `/var/named/` directory. Using `dig -x 172.16.9.x` on the computer to see whether reverse DNS lookup worked (you should get the names that you have defined).
5. Setup the second computer `mother` as a slave of `father`. All you need to do is to create two

slave zone sections in its `named.conf` file. It should be similar to the `father` computer, except that it is of type `slave`, have a `masters` option being a list containing only the IP address of `father`, and don't have an `allow-update` option. In the `father` zone file for `netlab.`, add a `NS` entry for `mother`, and set the `notify` option in the zone of `named.conf` to `yes` for automatic updating. You should not need to create a zone file: it should be copied by itself. Everytime when `father` notice that the serial number of the `SOA` resource changes, it tries to tell `mother` to update its list as well. After the changes, restart the name servers (using `/etc/init.d/named restart`). Check that it works correctly, by using the remaining 2 computers to do DNS queries (e.g., using `dig`). Also go to the `/var/named` directory of `mother` to see that it gets its own copy of the zone files.

6. Now the complicated parts: setup `son` as a sub-domain `child.netlab`, which contains the two computers `son` and `daughter`. To do this, setup `son` in exactly the same way as `father`, except that its zone name is `child.netlab.` instead of `netlab.`. The forwarders should point to the `father` computer instead of the computer outside. Don't create a reverse lookup zone for it yet. We suggest that you call the new zone file `child.zone`. Point the resolvers of both `son` and `daughter` to `son`. Check that `son` and `daughter` can refer to each other using the new name, in the `child` domain.
7. Up to this point, `son` and `daughter` should know their new names, but `father` and `mother` don't. Delete `son` and `daughter` from both zone files of `father`. Replace it by a `NS` line telling that `son.child.netlab.` is the DNS server for `child.netlab.`, and an `A` entry telling the IP address of `son.child.netlab.`. In the `/etc/named.conf` file, add an empty `forwarders` option for the `netlab` zone and its reverse zone to disable forwarding.
8. Now the real complicated part: to get reverse lookup correct. To do this, **part** of the domain `9.16.172.in-addr.arpa` must be delegated to `child`, but how? We will create aliases in the reverse zone file of `father` using `CNAME` entries. In particular, the names `x.9.16.172.in-addr.arpa.` will be made aliases of `x.child.9.16.172.in-addr.arpa.` for both the IP addresses of `son` and `daughter`. Now create an `NS` entry there, delegating the domain `child.9.16.172.in-addr.arpa.` to `son.child.netlab.`
9. Setup the reverse DNS lookup for `son`, using the same method as the setup of reverse DNS lookup for `father`. Use the domain name `child.9.16.172.in-addr.arpa.`. Check that reverse lookup works for all the 4 computers. Also, verify that `mother` receives updated zone files.

Troubleshooting

- If you find the server cannot start, you can find the error messages in the file `/var/log/messages`.
- When you find that the server is not replying recursive replies correctly, you might want to ask it to do a non-recursive request. That can be done using `dig` with the `+nonrecursive` option.
- Documentation of `bind` can be viewed using `Konquerer`, in the file `/usr/share/doc/bind-9.2.0/arm/Bv9ARM.html`.