

CSIS0234B Computer and Communication Networks (Class B)

Tutorial 6

Setting up local DNS servers

In this tutorial, we build a set of DNS servers that serve 4 computers within the lab. The appendix of the reading should provide you with examples. Work in group of 4–5 to complete the task. Login as `root`, and stop the firewall using `service iptables stop` so that every computer can use the DNS server you are building. If you encounter problems, the system log at `/var/log/messages` can usually help you identify it. `ethereal` might also be helpful.

1. Running `bind` as a caching server

- a. We have installed `bind` and some configuration files, so that the computers act as a caching name server knowing only `localhost`. All other queries are forwarded to root servers. **Start** `bind` with `service named start`, and modify `/etc/resolv.conf` to use itself (`127.0.0.1`) as a name server (remember the original name servers, we'll need them in the next step), while removing the `search` line there. Use `dig` (e.g., `dig localhost` or `dig -x 127.0.0.1`) to check which domain names it can resolve.
- b. The above doesn't work well: the department firewall blocks outgoing DNS requests to root servers. Add **global forwarders** in the `bind` configuration file `/etc/named.conf` so that queries are forwarded to the servers you recorded in (a), and reload the configuration using `service named reload`. Check that all domain names can now be resolved with `dig`. It shows the answer to your question, which name server is authoritative, and additional information that are usually needed, like the address of the name server.
- c. Use `ethereal` to **capture port 53 traffic** of "any" interface when using `dig`. Note that `dig` simply print whatever received from the name server in a nice way.

2. Creating domain names in a basic zone

Let's do some basic `bind` configurations. We call the 4 computers `father`, `mother`, `son`, `child` and `daughter`, and setup a top-level zone `netlab` to contain them. Each of them is an independent name server: a computer using them as DNS server will see those names. In principle there should be only one master server for a zone, but we set up all 4 computers anyway, so that each of you have a chance do the configuration. We have IP addresses for domain names like `father.netlab`, but `netlab` and `child.netlab` would not have IP addresses.

- a. **Create a new zone:** Edit `/etc/named.conf` to add a master zone `netlab`, with zone file at `/var/named/netlab.zone`—by copying and modifying the entry for `localhost` and the zone file `localhost.zone`. Add `A` records to the zone file so that the server can map the 4 domain names to IP addresses. Then reload the configuration. You should then be able to find the added resources with `dig`.
- b. **Create a reverse zone:** Edit `/etc/named.conf` to add a master zone `9.16.172.in-addr.arpa`, with zone file at `/var/named/named.netlab`—by copying and modifying the entry for `0.0.127.in-addr.arpa` and the zone file `named.local`. Add `PTR` records to the zone file so that the server can map the 4 IP addresses to domain names. Then reload the configuration. You should then be able to do reverse lookup with `dig`.

3. Providing redundancy

Now we change the role of `mother.netlab` to be a slave server for `father.netlab`, and `son.netlab` to be a slave server for `daughter.netlab`.

- a. In the master server, **add NS records of the slave** for **both** the zone files you've created in `/var/named`, and set the `notify` option in both zone entries in the configuration file `/etc/named.conf` so that slaves are notified about changes.
- b. In the slave server, delete both zone files in `/var/named` that you have created. **Modify the master zone entries you've added in `/etc/named.conf` to slave entries.** The slave entries should have a `masters` option set to a list containing the IP address of the master server (the format is the same as that of the forwarders), and should not have `allow-update` option.
- c. Reload configuration for the master server and then the slave. Check that **the slave server automatically get a zone file**, which is similar to (but not entirely the same as) that of the master. Try to modify the zone file in the master, change the serial number of the SOA record and reload the configuration. Check that the slave automatically updates its zone file.

4. Delegating the child domain

In this section the `father` and `mother` server let go of (delegate) the `child.netlab` domain, allowing `son` and `daughter` to control it (add records to it at their will).

- a. In `/var/named/netlab.zone` of `father`, add NS records for `child.netlab` domain, with values `son.child` and `daughter.child`. This makes `father` **non-authoritative** for `child.netlab`. Then modify `/etc/named.conf` in both `father` and `mother` to set the `forwarders` in the `netlab` zone entry to be an empty list—otherwise the queries would be forwarded to the global forwarder (which fails). Reload the configuration.
- b. In `son` and `daughter`, modify the zone entries in `/etc/named.conf` to serve the zone `child.netlab` instead of `netlab`, and change to use a new zone file `child.zone`. Change the global `forwarders` to use the IP addresses of `father` and `mother` instead. In `daughter`, rename `netlab.zone` to `child.zone`, and delete all references to `father` and `mother` there. Reload the configuration file in both computers. Now `son` and `daughter` only serve the `child.zone` domain. Check whether all the 4 servers correctly map each domain name to an IP address. You should also add some dummy resources to `daughter` and see that it is used by all the four servers.
- c. We now delegate the reverse zone, using CNAME records. In `father`, modify the reverse zone file so that `x.9.16.172...` for the computers in the child domain has a canonical name `x.child.9.16.172...`. Add NS records for the `child.9.16.172...` domain to delegate it. Modify the configuration file in both `father` and `mother` to prevent forwarding, and reload the configuration.
- d. In the `daughter` server, move the `named.netlab` reverse zone file to `named.child`, and modify the corresponding configuration file entry. In the reverse zone file, change the A records to provide the information about `x.child.9.16.172...` instead of `x.9.16.172...`. Again, it shouldn't hold the reverse mapping of the `father` and `mother` because it is not authoritative for them. Now verify that reverse DNS lookup works correctly on all four DNS servers, for all IP addresses of the 4 computers.