Building a DNS cache with BIND

1. Check the version of BIND which is installed

```
$ named -v
BIND 9.9.2-P1
```

2. Configure your RESOLV host to accept queries from neighbors

Log in to your RESOLV host if you haven't already done so (resolv.grpX.dns.nsrc.org).

Edit the file /etc/namedb/named.conf

```
$ sudo vi /etc/namedb/named.conf
```

If you prefer another editor (ee or jed, for example), use that instead of vi

If it is there, find the line:

```
      listen-on       { 127.0.0.1; };
```

... and REMOVE it.

Replace it with the following line:

```
      allow-recursion { 127.0.0.1; 10.20.0.0/16; };
```

Double check to see that there aren't any zones configured in your DNS. For instance, if you see a line like follows:

```
      zone "10.in-addr.arpa"     { type master; file "/etc/namedb/master/empty.db";
};
```

... remove it, and save the file.

NOTE: Be careful about the semicolons ';' and braces { } - BIND will complain if they are not placed correctly

By removing the line "listen-on ..." and adding the line "allow-recursion", we are telling BIND:

- please listen to the network for queries, not only on the local interface "127.0.0.1";

- please allow clients in the 10.20.0.0/16 to send queries to me, as well as myself;

3. Restart the cache and check it is running

If you haven't done so earlier, edit `/etc/rc.conf` and add two lines saying:

```
      named_chrootdir=""
```

```
      named_enable="YES"

NOTE: We would normally not turn off chroot, which is a security
      mechanism, but we need to do this here in the lab, because of
      restrictions from the virtualization environment. In a production
      environment, we wouldn't do this.

Then run these commands:

    $ sudo service named stop
    $ sudo service named start
    # ps auxwww | grep named
    # tail /var/log/messages

Check for successful startup with no error messages (you can ignore errors
about missing `master/localhost.rev` and `master/localhost-v6.rev`, as well
as messages regarding managed-keys-zone)


4. Reconfigure your resolver to use your own cache only
-------------------------------------------------------

You will do this on all your hosts (AUTH and RESOLV) - you will need to ssh
to each machine to make the following changes.

If you haven't done so earlier, edit `/etc/resolv.conf` as follows
(remember to use sudo !)

Remove any existing 'nameserver' lines, or comment them out by inserting '#'
at the front. 127.0.0.1 is the loopback address; that is, an IP address
which means 'send the packet to myself', and we'll use it as our nameserver:

    search dns.nsrc.org
    nameserver 10.20.X.3

... where X is the number of your group, i.e.: group 7, replace X with 7, etc.

Now save and exit.

5. Test resolution
------------------

Issue a query, for instance:

      $ dig google.com NS
      $ dig noc.dns.nsrc.org A

For each query:

1. Is the server responding ?
2. How do you know that you are talking to your OWN server ?
3. What do you notice ?

If your neighbour has got their cache working, then try sending some queries
to their cache:

    $ dig @10.20.X.1 somedomain.name

... where XXX is the IP of the machine in the class you want to send the
```

query to, and "somedomain.name" is the query you would like to perform.

Try and make some of the same queries you did before. Do the nameservers of the other machines answer you ?

Are you getting answers ? What about for dns.nsrc.org ?

Why ?

Help your neighbours to get their cache working if required.

6. Make sure you can resolve hostnames in the class

Ping other PCs in the room, where X is 1-32:

```
$ ping auth1.grpX.dns.nsrc.org
$ ping resolv.grpX.dns.nsrc.org
$ ping auth2.grpX.dns.nsrc.org
```

7. Watch the cache in operation

You can take a snapshot of the cache contents like this:

```
$ sudo ln -s /var/named/var/dump /var/dump
$ sudo /usr/sbin/rndc dumpdb
$ sudo less /var/named/var/dump/named_dump.db
```

(Don't do this on a busy cache - you will generate a huge dump file!)

You can watch the cache making queries to the outside world using `tcpdump` in a different window (log in again via SSH):

```
# tcpdump -n -s1500 -i eth0 udp port 53
```

Note that your ethernet interface may not necessarily be named `eth0`.

To find out the name if your ethernet interface - e.g. `em0` or `bge0` - run "ifconfig"

While tcpdump is running, in the first window flush your cache (so it forgets all existing data) and then issue some queries.

```
# rndc flush
# dig noc.dns.nsrc.org.   -- and watch tcpdump output. What do you see?

# dig noc.dns.nsrc.org.   -- watch tcpdump again. This time?
```

NOTE: we now have enabled BIND to be recursive!

Remember that it is not a good idea to run recursive and authoritative service on the same server.