**Week #4**

**Implementation of a Local DNS Server and Authoritative NameServer**

DNS (Domain Name System) is the Internet’s phone book; it translates hostnames to IP addresses (and vice versa). This translation is through DNS resolution, which happens behind the scene.

The objectives of this lab are to understand:

* Install, set up and deploy a local DNS server
* Deploy authoritative nameserver for example.com domain

**Lab Setup (with Internet Connection)**

DNS Server: 10.2.22.184 User/Client: 10.2.22.195 ***Note:*** Use the default IP address provided by PESU LAN.

**Observation 1:**

Ping a computer such as www.google.com (any domain). Please use Wireshark to show the DNS query triggered by your ping command and DNS response. Describe your observation.

(Take a screenshot).

**Part 1: Setting Up a Local DNS Server**

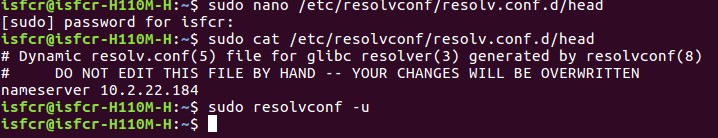
# Task 1: Configure the User/Client Machine

On the client machine 10.2.22.195, we need to use 10.2.22.184 as the local DNS server. This is achieved by changing the resolver configuration file **(/etc/resolv.conf**) of the user machine, so the server 10.2.22.184 is added as the first nameserver entry in the file, i.e., this server will be used as the primary DNS server. Add the following entry to the **/etc/resolvconf/resolv.conf.d/head** file.

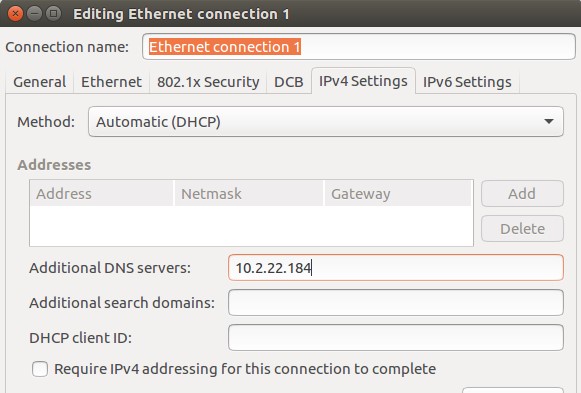
**nameserver 10.2.22.184**

Run the following command for the change to take effect. **sudo resolvconf -u**

The following screenshot shows how to set DNS server on the client machine.



Also, add 10.2.22.184 in ‘Additional DNS servers’ field in IPv4 settings of client machine.



**Observation 2:**

Ping a computer such as [www.google.com.](http://www.google.com/) Please use Wireshark to show the DNS query triggered by your ping command and DNS response. Describe your observation. (Take a screenshot).

# Task 2: Set Up a Local DNS Server

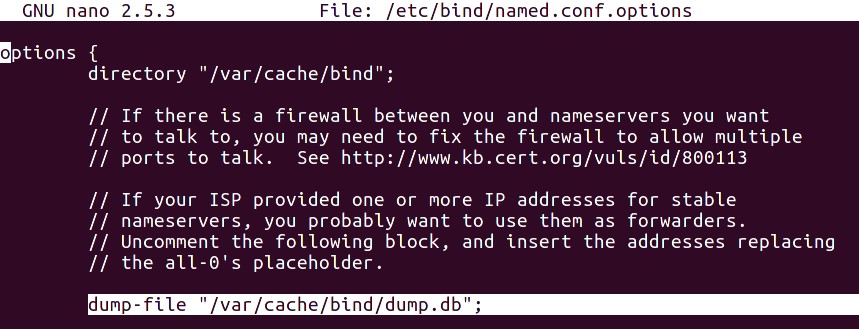
Note: If bind9 server is not already installed, install using the command

**$ sudo apt-get update**

**$** **sudo apt-get install bind9**

**Step 1: Configure the BIND9 Server.**

BIND9 gets its configuration from a file called **/etc/bind/named.conf**. This file is the primary configuration file, and it usually contains several “include” entries. One of the included files is called **/etc/bind/named.conf.options**. This is where we typically set up the configuration options. Let us first set up an option related to DNS cache by adding a dump-file entry to the options block. The above option specifies where the cache content should be dumped to if BIND is asked to dump its cache.



The above option specifies where the cache content should be dumped to if BIND is asked to dump its cache. If this option is not specified, BIND dumps the cache to a default file called **/var/cache/bind/named\_dump.db**.

**Step 2: Start DNS server**

We start the DNS server using the command:

**$ sudo service bind9 restart**



**Observation 3:**

Now, go back to your user machine (10.2.22.195), and ping a computer such as [www.google.com](http://www.google.com/) and describe your observation. Please use Wireshark to show the DNS query triggered by your ping command. Please also indicate when the DNS cache is used. (Take a screenshot).

**Observation 4:**

The two commands shown below are related to DNS cache. The first command dumps the content of the cache to the file specified above, and the second command clears the cache. You need extract the DNS cache using ‘grep’ command and take screenshot of [www.google.com](http://www.google.com/) DNS cache.



**Note:** Compare the above three Wireshark DNS packet capture screenshots taken above.

# Part 2: Setting Up an Authoritative Nameserver for example.com domain

**Task 3: Host a Zone in the Local DNS server.**

Assume that we own a domain, we will be responsible for providing the definitive answer regarding this domain. We will use our local DNS server as the authoritative nameserver for the domain. In this lab, we will set up an authoritative server for the **example.com** domain. This domain name is reserved for use in documentation, and is not owned by anybody, so it is safe to use it.

**Step 1: Create Zones**

We had two zone entries in the DNS server by adding the following contents to **/etc/bind/named.conf** as shown in the below screenshot. The first zone is for forward lookup (from hostname to IP), and the second zone is for reverse lookup (from IP to hostname).



Note: In above screenshot, 10.2.22.0 is the subnet mask of your IP address. This applies to all part of the experiment.

**Step 2: Setup the forward lookup zone file**

We create **example.com.db** zone file with the following contents in the **/etc/bind/** directory where the actual DNS resolution is stored.



The symbol ‘@’ is a special notation representing the origin specified in **named.conf** (the string after "**zone**"). Therefore, ‘@’ here stands for **example.com**. This zone file contains 7 resource records (RRs), including a SOA (Start Of Authority) RR, a NS (Name Server) RR, a MX (Mail eXchanger) RR, and 4 A (host Address) RRs.

**Step 3: Setup the reverse lookup zone file**

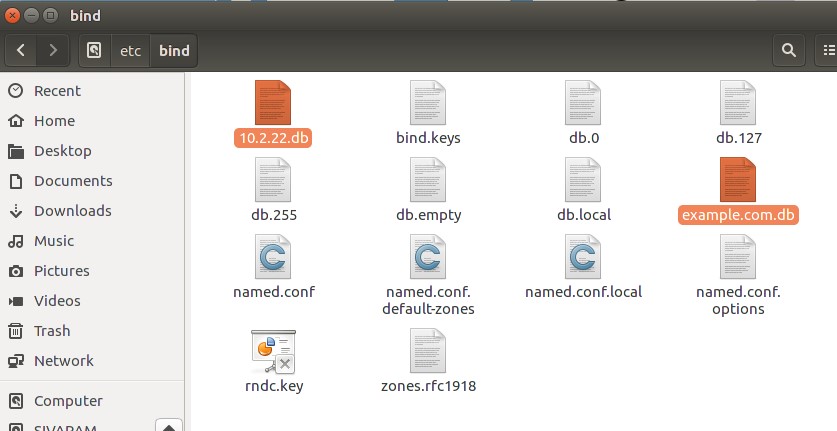
We create a reverse DNS lookup file called **10.2.22.db** for the example.net domain to support DNS reverse lookup, i.e., from IP address to hostname in the **/etc/bind/** directory with the following contents.



**Note:** You can download the above two db files from Edmodo. Indent spacing is essential.

**Step 4:** Copy the above files into **/etc/bind** location.





**Task 4: Restart the BIND server and test**

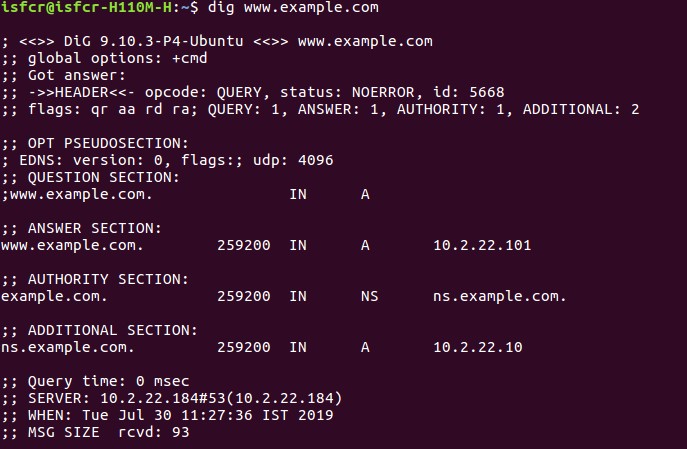
**Step 1:** When all the changes are made, remember to restart the BIND server. Now we will restart the DNS server using the following command:

**$ sudo service bind9 restart**



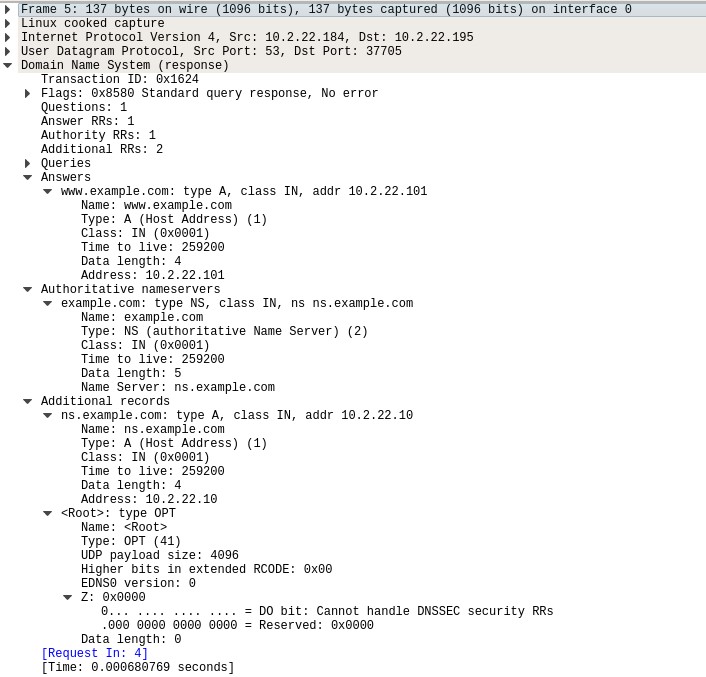
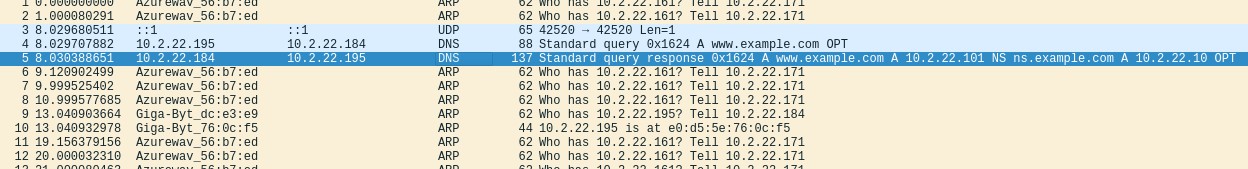
**Step 2:** Now, go back to the client machine and ask the local DNS server for the IP address of www.example.com using the dig command.

**Dig** stands for (Domain Information Groper) is a network administration command-line tool for querying DNS name servers. It is useful for verifying and troubleshooting DNS problems and also to perform DNS lookups and displays the answers that are returned from the name server that were queried. dig is part of the BIND domain name server software suite.



We can see that the ANSWER SECTION contains the DNS mapping. We can see that the IP address of [www.example.com](http://www.example.com/) is now 10.2.22.101, which is what we have setup in the DNS server.

**Step 3: Observe the results in Wireshark capture.**



To load and clear DNS cache, use the below commands.



**Edmodo Requirements:**

1. Wireshark packet capture screenshots (Observations 1-3)
2. DNS cache for [www.google.com](http://www.google.com/) (Observation 4)
3. **dig www.example.com** command (in Terminal)
4. Wireshark packet capture – **dig www.example.com** command
5. DNS cache on server machine after dig command

**Observation Notebook Requirements:**

For **‘ping www.flipkart.com’,** answer the following questions

1. Locate the DNS query and response messages. Are then sent over UDP or TCP?
2. What is the destination port for the DNS query message? What is the source port of DNS response message?
3. To what IP address is the DNS query message sent? Use ipconfig to determine the IP address of your local DNS server. Are these two IP addresses the same?
4. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?
5. Examine the DNS response message. How many “answers” are provided? What do each of these answers contain?
6. Consider the subsequent TCP SYN packet sent by your host. Does the destination IP address of the SYN packet correspond to any of the IP addresses provided in the DNS response message?