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

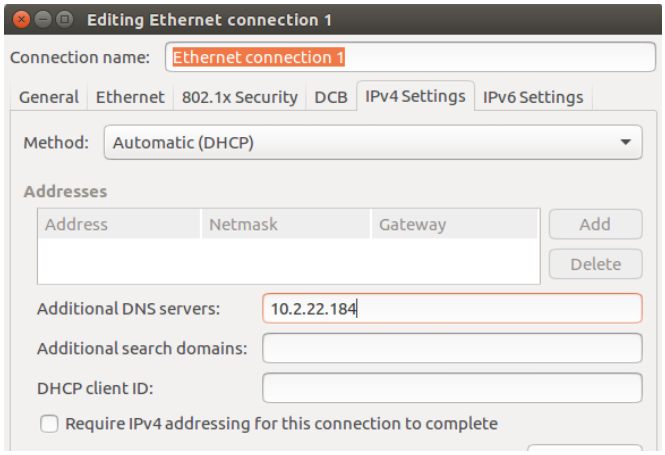
Run the following command for the change to take effect.

sudo resolvconf -u

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

```
isfcr@isfcr-H110M-H:~$ sudo nano /etc/resolvconf/resolv.conf.d/head
[sudo] password for isfcr:
isfcr@isfcr-H110M-H:~$ sudo cat /etc/resolvconf/resolv.conf.d/head
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 10.2.22.184
isfcr@isfcr-H110M-H:~$ sudo resolvconf -u
isfcr@isfcr-H110M-H:~$
```

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. 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.

```
isfcr@isfcr-H110M-H:~$ sudo nano /etc/bind/named.conf.options
[sudo] password for isfcr:
```

```
  GNU nano 2.5.3            File: /etc/bind/named.conf.options

options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

        dump-file "/var/cache/bind/dump.db";
```

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

```
isfcr@isfcr-H110M-H:~$ sudo service bind9 restart
isfcr@isfcr-H110M-H:~$
```

Observation 3:

Now, go back to your user machine (10.2.22.195), and ping a computer such as 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 DNS cache.

```
isfcr@isfcr-H110M-H:~$ sudo rndc dumpdb -cache
isfcr@isfcr-H110M-H:~$ sudo rndc flush
```

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).

```
isfcr@isfcr-H110M-H:~$ sudo nano /etc/bind/named.conf
isfcr@isfcr-H110M-H:~$ sudo cat /etc/bind/named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";

zone "example.com" {
type master;
file "/etc/bind/example.com.db";
};

zone "22.2.10.in-addr.arpa" {
type master;
file "/etc/bind/10.2.22.db";
};
isfcr@isfcr-H110M-H:~$
```

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.

```
$TTL 3D
@     IN    SOA   ns.example.com. admin.example.com. (
            2008111001
            8H
            2H
            4W
            1D)

@     IN    NS    ns.example.com.
@     IN    MX    10 mail.example.com.

www   IN    A     10.2.22.101
mail  IN    A     10.2.22.102
ns    IN    A     10.2.22.10
*.example.com.   IN    A 10.2.22.100
```

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

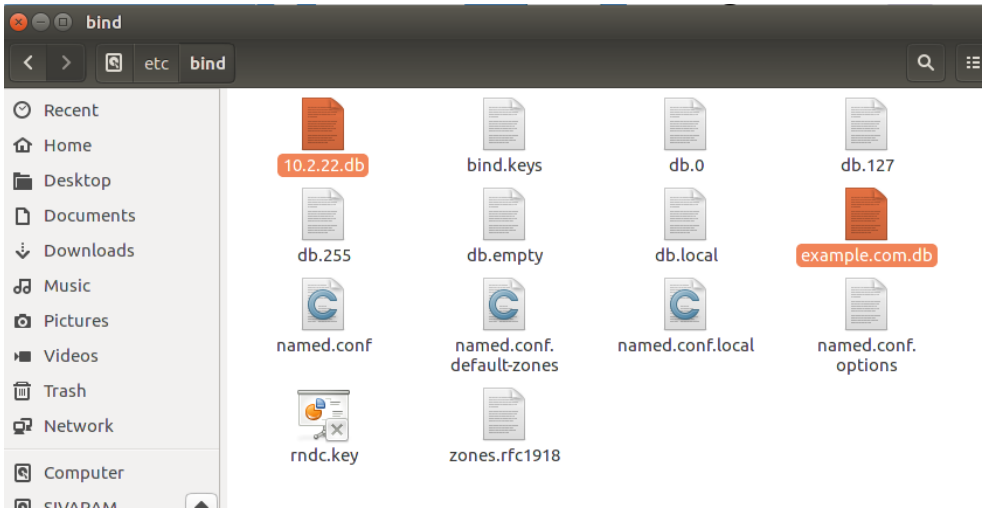
We create a reverse DNS lookup file called **10.2.22.db** for the example.net domain to support DNS reverse lookup, i.e., from IP address to hostname in the **/etc/bind/** directory with the following contents.

```
$TTL 3D
@     IN    SOA   ns.example.com. admin.example.com. (
            2008111001
            8H
            2H
            4W
            1D)
@     IN    NS    ns.example.com.

101   IN    PTR   www.example.com.
102   IN    PTR   mail.example.com.
10    IN    PTR   ns.example.com.
```

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.

```
isfcr@isfcr-H110M-H:~$ sudo cp 10.2.22.db /etc/bind
isfcr@isfcr-H110M-H:~$ sudo cp example.com.db /etc/bind
```



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

```
isfcr@isfcr-H110M-H:~$ sudo service bind9 restart
isfcr@isfcr-H110M-H:~$
```

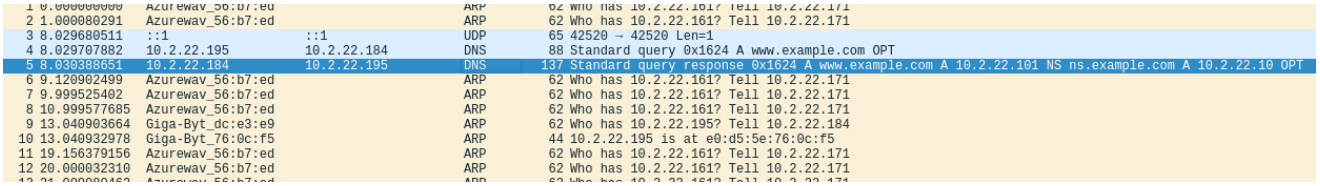
Step 2: Now, go back to the client machine and ask the local DNS server for the IP address of www.example.com using the dig command.

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.

```
isfcr@isfcr-H110M-H:~$ dig www.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5668
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        259200  IN      A       10.2.22.101

;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns.example.com.

;; ADDITIONAL SECTION:
ns.example.com.         259200  IN      A       10.2.22.10

;; Query time: 0 msec
;; SERVER: 10.2.22.184#53(10.2.22.184)
;; WHEN: Tue Jul 30 11:27:36 IST 2019
;; MSG SIZE  rcvd: 93
```

We can see that the ANSWER SECTION contains the DNS mapping. We can see that the IP address of 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.



2	1.000080291	Azurewav_56:b7:ed		ARP	62 Who has 10.2.22.161? Tell 10.2.22.171
3	8.029680511	::1	::1	UDP	65 42520 → 42520 Len=1
4	8.029707882	10.2.22.195	10.2.22.184	DNS	88 Standard query 0x1624 A www.example.com OPT
5	8.030388651	10.2.22.184	10.2.22.195	DNS	137 Standard query response 0x1624 A www.example.com A 10.2.22.101 NS ns.example.com A 10.2.22.10 OPT
6	9.120902499	Azurewav_56:b7:ed		ARP	62 Who has 10.2.22.161? Tell 10.2.22.171
7	9.999525402	Azurewav_56:b7:ed		ARP	62 Who has 10.2.22.161? Tell 10.2.22.171
8	10.999577685	Azurewav_56:b7:ed		ARP	62 Who has 10.2.22.161? Tell 10.2.22.171
9	13.040903664	Giga-Byt_dc:e3:e9		ARP	62 Who has 10.2.22.195? Tell 10.2.22.184
10	13.040932978	Giga-Byt_76:0c:f5		ARP	44 10.2.22.195 is at e0:d5:5e:76:0c:f5
11	19.156379156	Azurewav_56:b7:ed		ARP	62 Who has 10.2.22.161? Tell 10.2.22.171
12	20.000032310	Azurewav_56:b7:ed		ARP	62 Who has 10.2.22.161? Tell 10.2.22.171

```
▸ Frame 5: 137 bytes on wire (1096 bits), 137 bytes captured (1096 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 10.2.22.184, Dst: 10.2.22.195
▸ User Datagram Protocol, Src Port: 53, Dst Port: 37705
▾ Domain Name System (response)
    Transaction ID: 0x1624
  ▸ Flags: 0x8580 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 1
    Additional RRs: 2
  ▸ Queries
  ▾ Answers
    ▾ www.example.com: type A, class IN, addr 10.2.22.101
        Name: www.example.com
        Type: A (Host Address) (1)
        Class: IN (0x0001)
        Time to live: 259200
        Data length: 4
        Address: 10.2.22.101
  ▾ Authoritative nameservers
    ▾ example.com: type NS, class IN, ns ns.example.com
        Name: example.com
        Type: NS (authoritative Name Server) (2)
        Class: IN (0x0001)
        Time to live: 259200
        Data length: 5
        Name Server: ns.example.com
  ▾ Additional records
    ▾ ns.example.com: type A, class IN, addr 10.2.22.10
        Name: ns.example.com
        Type: A (Host Address) (1)
        Class: IN (0x0001)
        Time to live: 259200
        Data length: 4
        Address: 10.2.22.10
    ▾ <Root>: type OPT
        Name: <Root>
        Type: OPT (41)
        UDP payload size: 4096
        Higher bits in extended RCODE: 0x00
        EDNS0 version: 0
      ▾ Z: 0x0000
          0... .... .... .... = DO bit: Cannot handle DNSSEC security RRs
          .000 0000 0000 0000 = Reserved: 0x0000
        Data length: 0
    [Request In: 4]
    [Time: 0.000680769 seconds]
```

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

```
isfcr@isfcr-H110M-H:~$ sudo rndc dumpdb -cache
isfcr@isfcr-H110M-H:~$ sudo rndc flush
```

Edmodo Requirements:

1) Wireshark packet capture screenshots (Observations 1-3)
2) DNS cache for 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?