

PES UNIVERSITY
EC CAMPUS BANGALORE

NAME: R SHARMILA

SRN: PES2UG19CS309

DATE: 21-02-2021

SUBJECT: Computer Network Laboratory

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.

CLIENT:

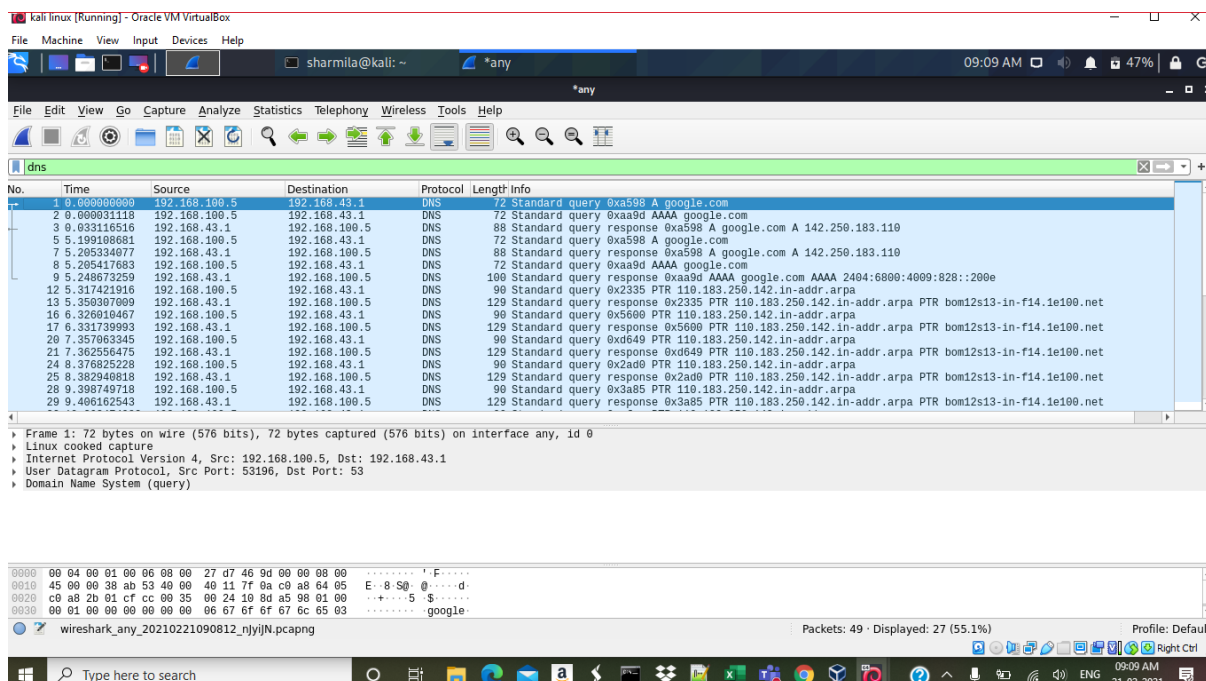
```
sharmila@kali: ~  
File Actions Edit View Help  
(sharmila@kali)-[~]  
$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.100.5 netmask 255.255.255.0 broadcast 192.168.100.255  
    ether 08:00:27:d7:46:9d txqueuelen 1000 (Ethernet)  
    RX packets 9 bytes 2660 (2.5 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 17 bytes 2294 (2.2 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 12 bytes 600 (600.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 12 bytes 600 (600.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
(sharmila@kali)-[~]  
$
```

SERVER:

```
ubuntu@ubuntu-VirtualBox: ~  
ubuntu@ubuntu-VirtualBox:~$ ip addr  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:ae:91:37 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.100.4/24 brd 192.168.100.255 scope global dynamic noprefixroute enp0s3  
        valid_lft 552sec preferred_lft 552sec  
    inet6 fe80::e8d0:38eb:4580:462e/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
ubuntu@ubuntu-VirtualBox:~$
```

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.

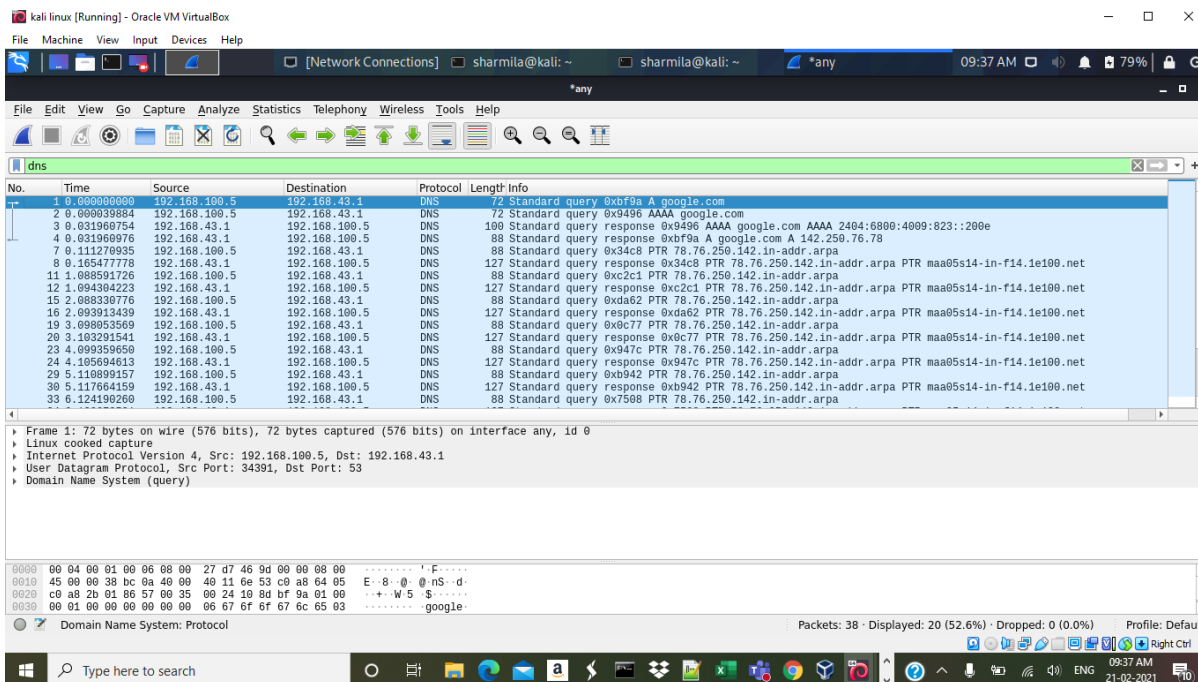
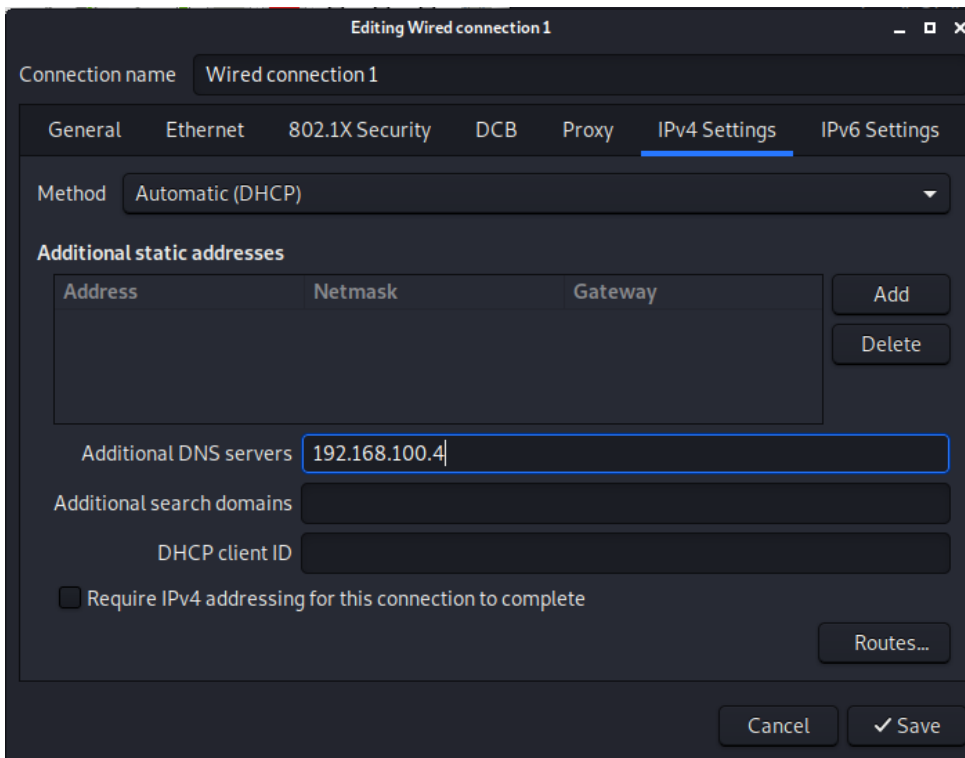


OBSERVATION 2 : Configuring Client Machine

- The IP Address of the client machine is 192.168.100.5 and the IP Address of the server machine is 192.168.100.4
- We need to add the IP Address of the custom DNS server (192.168.100.4) to the client machine.
- This is done by adding the IP address of the server to the file ***/etc/resolvconf/resolv.conf.d/head*** which stores the order of DNS server resolution. This ensures that the custom DNS server will be used to resolve names.
- The IP Address of the custom DNS server is also added to the DNS menu under the IPv4 Network Settings.
- The changes are applied by using the command ***sudo resolvconf -u***

```
sharmila@kali: ~  
File Actions Edit View Help  
GNU nano 5.3 /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  
# 127.0.0.53 is the systemd-resolved stub resolver.  
# run "resolvectl status" to see details about the actual nameservers.  
nameserver 192.168.100.4  
19d AAAA google.com  
ponse 0xaa9d AAAA google.com AAAA 2404:6800:4009:828::200e  
335 PTR 110.183.250.142.in-addr.arpa  
ponse 0x2335 PTR 110.183.250.142.in-addr.arpa PTR bom12s13-in-f14.1e100.net  
399 PTR 110.183.250.142.in-addr.arpa  
ponse 0x5600 PTR 110.183.250.142.in-addr.arpa PTR bom12s13-in-f14.1e100.net  
349 PTR 110.183.250.142.in-addr.arpa  
ponse 0xd649 PTR 110.183.250.142.in-addr.arpa PTR bom12s13-in-f14.1e100.net  
4d0 PTR 110.183.250.142.in-addr.arpa  
ponse 0x2add PTR 110.183.250.142.in-addr.arpa PTR bom12s13-in-f14.1e100.net
```

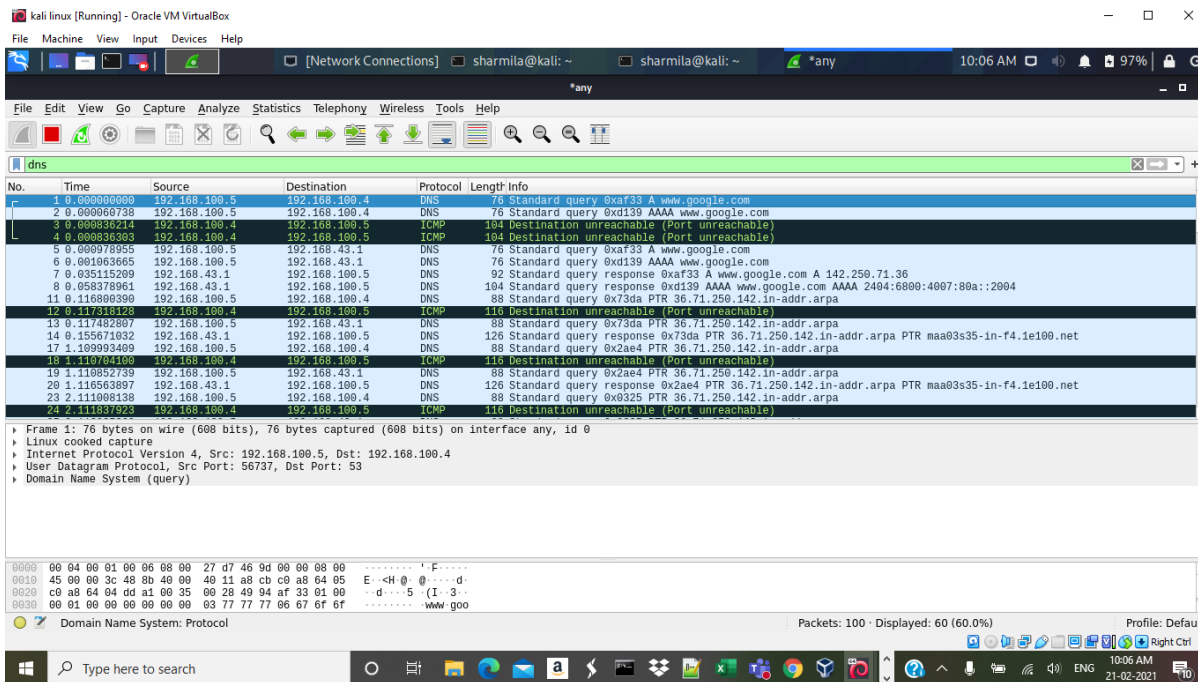
```
sharmila@kali: ~  
File Actions Edit View Help  
GNU nano 5.3 /etc/resolvconf/resolv.conf.d/head  
(sharmila@kali)-[~]  
$ sudo cat /etc/resolvconf/resolv.conf.d/head  
[sudo] password for sharmila:  
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf  
(8).  
# DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN  
# 127.0.0.53 is the systemd-resolved stub resolver.  
# run "resolvectl status" to see details about the actual nameservers.  
nameserver 192.168.100.4  
  
(sharmila@kali)-[~]  
$
```



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.

- The google website is pinged again, and Wireshark is used to capture packets.
- We obtain a destination unreachable error in Wireshark as the server machine does not have a DNS server associated with it
- The client tries to obtain the DNS record from 192.168.100.4 but it does not receive any



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 : CONFIGURING 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.
- BIND dumps the cache to a default file called ***/var/cache/bind/dump.db***.

```

ubuntu@ubuntu-VirtualBox: ~
GNU nano 4.8 /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";
    // forwarders {
    //     0.0.0.0;
    // };

    //=====  

    // If BIND logs error messages about the root key being expired,  

    // you will need to update your keys.  See https://www.isc.org/bind-keys  

    //=====
[ Read 24 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos  

^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line

```

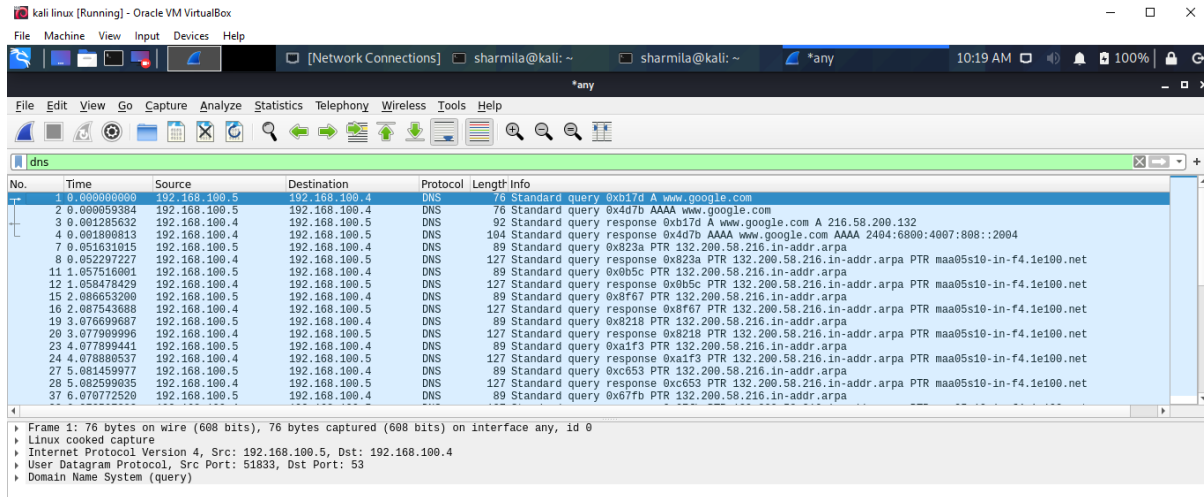

Step 2: Start DNS server:

\$ sudo service bind9 restart

```
ubuntu@ubuntu-VirtualBox:~$ sudo service bind9 restart
ubuntu@ubuntu-VirtualBox:~$
```

Observation 3:

Now, go back to your user machine (192.168.100.4), and ping a computer such as www.google.com



OBSERVATION 5 : Viewing the cache file

- The cache can be dumped into the file using `sudo rndc dumpdb -cache` and can be cleared or flushed out using `sudo rndc flush`.
- The linux command to extract cache for www.google.com from dump.db file,
- `sudo cat /var/cache/bind/dump.db | grep "google"`

```
ubuntu@ubuntu-VirtualBox:~$ sudo service bind9 restart
[sudo] password for ubuntu:
ubuntu@ubuntu-VirtualBox:~$ sudo rndc dumpdb -cache
ubuntu@ubuntu-VirtualBox:~$ sudo cat /var/cache/bind/dump.db | grep "flipkart"
ubuntu@ubuntu-VirtualBox:~$ sudo rndc flush
ubuntu@ubuntu-VirtualBox:~$
```

*any						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
dns						
No.	Time	Source	Destination	Protocol	Length	Info
6	0.680490752	192.168.100.4	192.168.43.1	DNS	91	Standard query 0xb1a AAAA connectivity-check.ubuntu.com
7	0.708284933	192.168.43.1	192.168.100.4	DNS	91	Standard query response 0xb1a AAAA connectivity-check.ubuntu.com
8	0.710072729	127.0.0.1	127.0.0.53	DNS	102	Standard query 0xd3e AAAA connectivity-check.ubuntu.com OPT
9	0.710446703	192.168.100.4	192.168.43.1	DNS	91	Standard query 0xd6c AAAA connectivity-check.ubuntu.com
10	0.71855294	192.168.43.1	192.168.100.4	DNS	91	Standard query response 0xd6c AAAA connectivity-check.ubuntu.com
11	0.718276241	127.0.0.53	127.0.0.1	DNS	102	Standard query response 0xd3e AAAA connectivity-check.ubuntu.com
18	46.594795815	192.168.100.4	192.168.43.1	DNS	91	Standard query 0xd8 A connectivity-check.ubuntu.com
19	46.635226342	192.168.43.1	192.168.100.4	DNS	203	Standard query response 0xd8 A connectivity-check.ubuntu.com
46	51.442928606	192.168.100.4	192.5.5.241	DNS	84	Standard query 0xbe2 DNSKEY <Root> OPT
47	51.443039516	192.168.100.4	192.5.5.241	DNS	84	Standard query 0xba1d NS <Root> OPT

▶ Frame 6: 91 bytes on wire (728 bits), 91 bytes captured (728 bits) on interface any, id 0
 ▶ Linux cooked capture
 ▶ Internet Protocol Version 4, Src: 192.168.100.4, Dst: 192.168.43.1
 ▶ User Datagram Protocol, Src Port: 49139, Dst Port: 53
 ▶ Domain Name System (query)

OBSERVATION 6 : Host a Zone in the Local DNS server

- 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).

```

ubuntu@ubuntu-VirtualBox: ~ ×      ubuntu@ubuntu-VirtualBox: ~ ×      ubuntu@ubuntu-VirtualBox: ~ ×
GNU nano 4.8                                                                /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 "100.168.192.in-addr.arpa" {
type master;
file "/etc/bind/192.168.100.db";
};

```

```
ubuntu@ubuntu-VirtualBox: ~  
ubuntu@ubuntu-VirtualBox:~$ sudo cat /etc/bind/named.conf  
[sudo] password for ubuntu:  
// 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 "100.168.192.in-addr.arpa" {  
    type master;  
    file "/etc/bind/192.168.100.db";  
};  
ubuntu@ubuntu-VirtualBox:~$
```

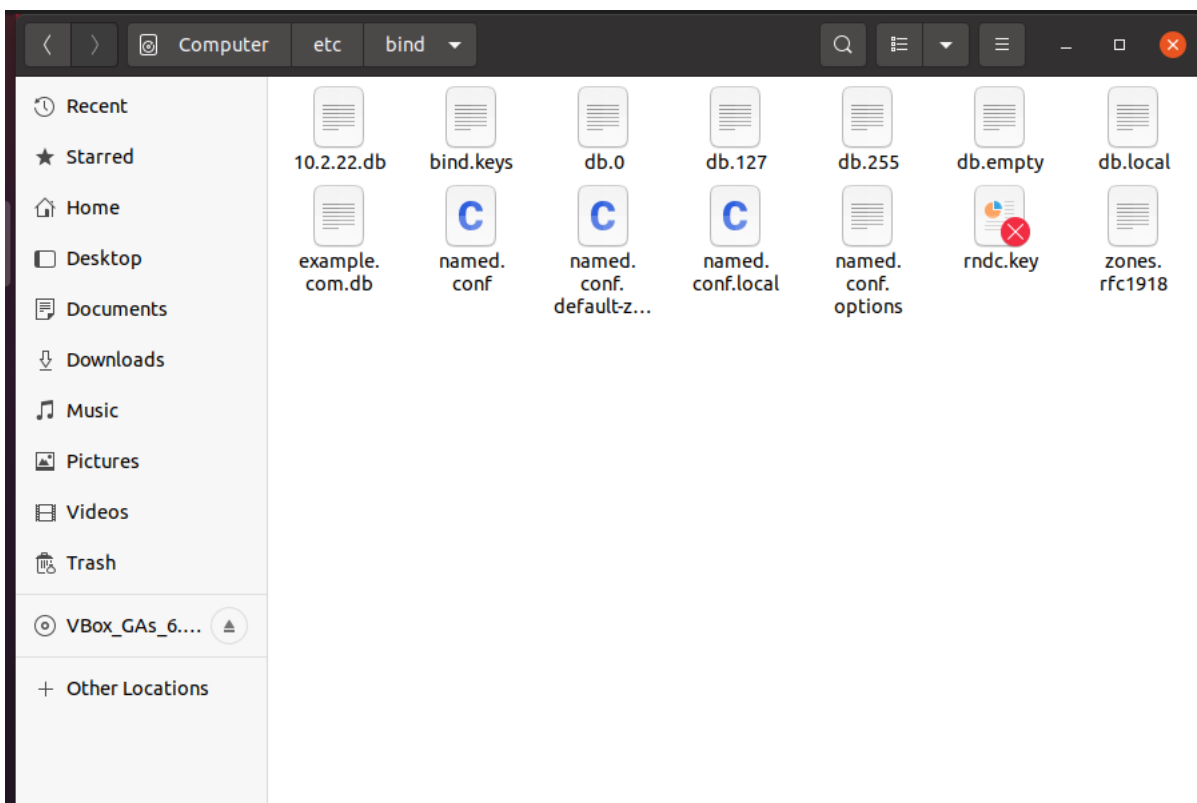
Step 2: Setup the forward lookup zone file

- We create example.com.db zone file with the following contents in the /etc/bind/ directory where the actual DNS resolution is stored.
- We create 10.2.22.db zone file with the following contents in the /etc/bind/ directory where the actual DNS resolution is stored.

```
ubuntu@ubuntu-VirtualBox: ~  
GNU nano 4.8 /etc/bind/example.com.db  
$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        192.168.100.101  
mail    IN      A        192.168.100.102  
ns      IN      A        192.168.100.10  
*.example.com. IN      A 192.168.100.100  
  
[ Wrote 16 lines ]  
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  
^X Exit      ^R Read File ^\ Replace  ^U Paste Text ^T To Spell  ^_ Go To Line
```



```
ubuntu@ubuntu-VirtualBox: ~  
GNU nano 4.8 /etc/bind/10.2.22.db Modified  
$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.  
|
```



OBSERVATION 7 : 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.

```
ubuntu@ubuntu-VirtualBox:~$ sudo service bind9 restart  
ubuntu@ubuntu-VirtualBox:~$ |
```

```
sharmila@kali: ~  
File Actions Edit View Help  
ter), use +noidn  
  
(sharmila@kali)-[~]  
$ dig www.example.com 10 x  
  
; <<>> DiG 9.16.11-Debian <<>> www.example.com  
;; global options: +cmd  
;; Got answer:  
;; -->HEADER<-- opcode: QUERY, status: NOERROR, id: 55749  
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1  
  
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags;; udp: 1280  
;; QUESTION SECTION:  
;www.example.com. IN A  
  
;; ANSWER SECTION:  
www.example.com. 40054 IN A 93.184.216.34  
  
;; Query time: 55 msec  
;; SERVER: 192.168.43.1#53(192.168.43.1)  
;; WHEN: Sun Feb 21 13:18:35 IST 2021  
;; MSG SIZE rcvd: 60  
  
(sharmila@kali)-[~]  
$
```

kali linux [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

sharmila@kali: ~ *any 02:09 PM 49%

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

dns

No.	Time	Source	Destination	Protocol	Length	Info
3	0.000307923	192.168.100.5	192.168.100.4	DNS	100	Standard query 0x8670 A www.example.com OPT
4	0.001386012	192.168.100.4	192.168.100.5	ICMP	128	Destination unreachable (Port unreachable)
5	1.002725778	192.168.100.5	192.168.43.1	DNS	100	Standard query 0x8670 A www.example.com OPT
8	1.009632888	192.168.43.1	192.168.100.5	DNS	93	Standard query response 0x8670 A www.example.com A 93.184.216.34

Frame 8: 93 bytes on wire (744 bits), 93 bytes captured (744 bits) on interface any, id 0

- Linux cooked capture
- Internet Protocol Version 4, Src: 192.168.43.1, Dst: 192.168.100.5
- User Datagram Protocol, Src Port: 53, Dst Port: 47133
- Domain Name System (response)
 - Transaction ID: 0x8670
 - Flags: 0x81a0 Standard query response, No error
 - Questions: 1
 - Answer RRs: 1
 - Authority RRs: 0
 - Additional RRs: 0
- Queries
 - www.example.com: type A, class IN
- Answers
 - www.example.com: type A, class IN, addr 93.184.216.34
 - Name: www.example.com
 - Type: A (Host Address) (1)
 - Class: IN (0x0001)
 - Time to live: 37206 (10 hours, 20 minutes, 6 seconds)
 - Data length: 4
 - Address: 93.184.216.34

[Request in: 5]
[Time: 0.000907110 seconds]

0020 c0 a8 64 05 00 35 b8 1d 00 39 00 dc 86 70 81 a0 . . d . 5 . . . 9 . . p . .
0030 00 01 00 01 00 00 00 00 03 77 77 77 07 65 78 61 www . exa
0040 6d 70 6c 65 03 63 6f 6d 00 00 01 00 01 c0 0c 00 mple . com
0050 01 00 01 00 00 91 56 00 04 5d 08 d8 22 V j . . *

Number of additional records in packet (dns.count.add_rr), 2 bytes

Packets: 18 · Displayed: 4 (22.2%) Profile: Default

02:09 PM 21-02-2021

QUESTIONS

1. Locate the DNS query and response messages. Are then sent over UDP or TCP?

The DNS Query and Response messages are visible in the screenshots. They are sent over UDP(User Datagram Protocol).

2. What is the destination port for the DNS query message? What is the source port of DNS response message?

The destination and source ports of the DNS query and response messages are the same. The port number for DNS protocol is 53.

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?

The DNS query is made to server at the IP Address 192.168.100.4. This is the same as the local DNS server configured.

4. Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"?

The DNS Query is of type A since it requests for an authoritative record. The answer section is empty since it does not have any answer.

5. Examine the DNS response message. How many "answers" are provided? What do each of these answers contain?

The answer section of the DNS response message contains one Resource Record. A type RR: This provides the IP Address of the hostname

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?

The destination IP Address of the SYN packet corresponds to the IP Address of hostname (www.google.com) retrieved from the response message.