



COMPUTER NETWORKS LAB

COURSE CODE:UE19CS255

NAME:PRIYA MOHATA

SRN:PES2UG19CS301

SECTION:E

DATE:21/02/2021

EXPERIMENT: 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.

DNS Client : 10.0.5.36

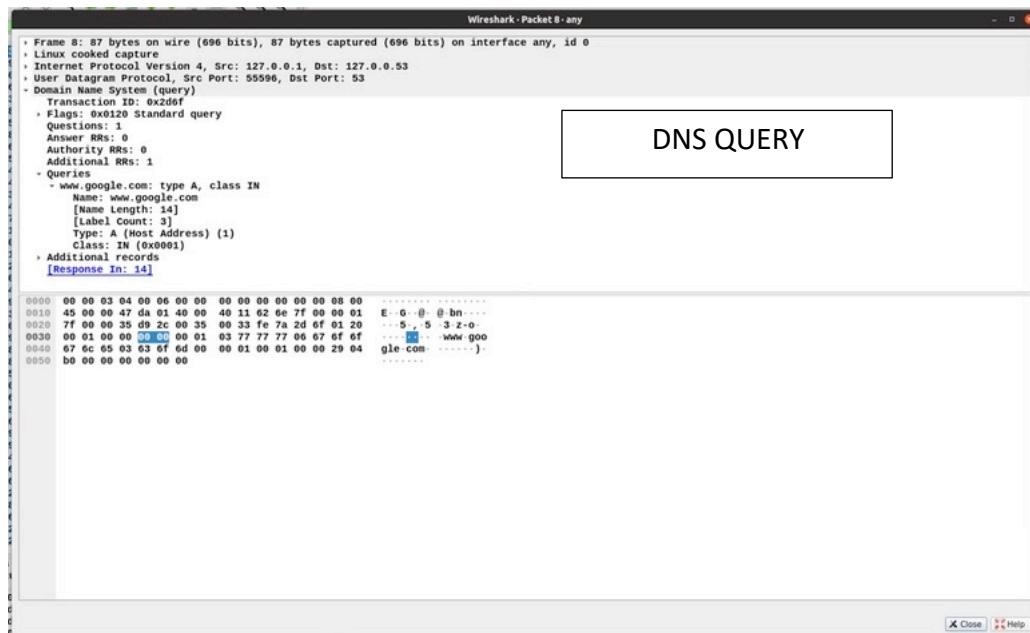
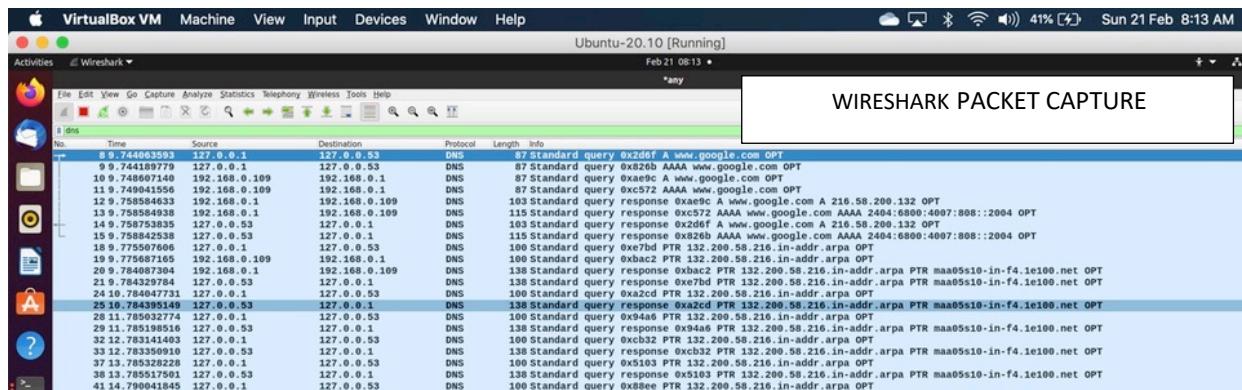
```
 priyamohata -- bash -- 80x24
[Password:
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    inet 127.0.0.1/8 brd 0.0.0.0 scopeid 0x1
        inet6 ::1/128
            inet6 fe80::1/64 scopeid 0x1
en3: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether ac:de:80:00:11:22
        inet6 fe80::aede:4bff:fe00:1122/64 scopeid 0x4
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether a4:83:07:69:e1:4b
        inet 10.0.5.36/24 brd 10.0.5.255 en0
        inet6 fe80::100a:8be4:3796:a4d8/64 secured scopeid 0x6
        inet 192.168.0.140/24 brd 192.168.0.255 en0
awdl0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1484
    ether 36:48:16:9b:41:96
        inet6 fe80::3448:16ff:fe9b:4196/64 scopeid 0xb
llw0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 36:48:16:9b:41:96
        inet6 fe80::3448:16ff:fe9b:4196/64 scopeid 0xc
utun0: flags=8851<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1380
    inet6 fe80::2215:8aee:35cc:c7c/64 scopeid 0xd
utun1: flags=8851<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 2000
    inet6 fe80::4bbd:1f33:e62f:ff22/64 scopeid 0xe
(base) priyas-MacBook-Air:- priyamohata$
```

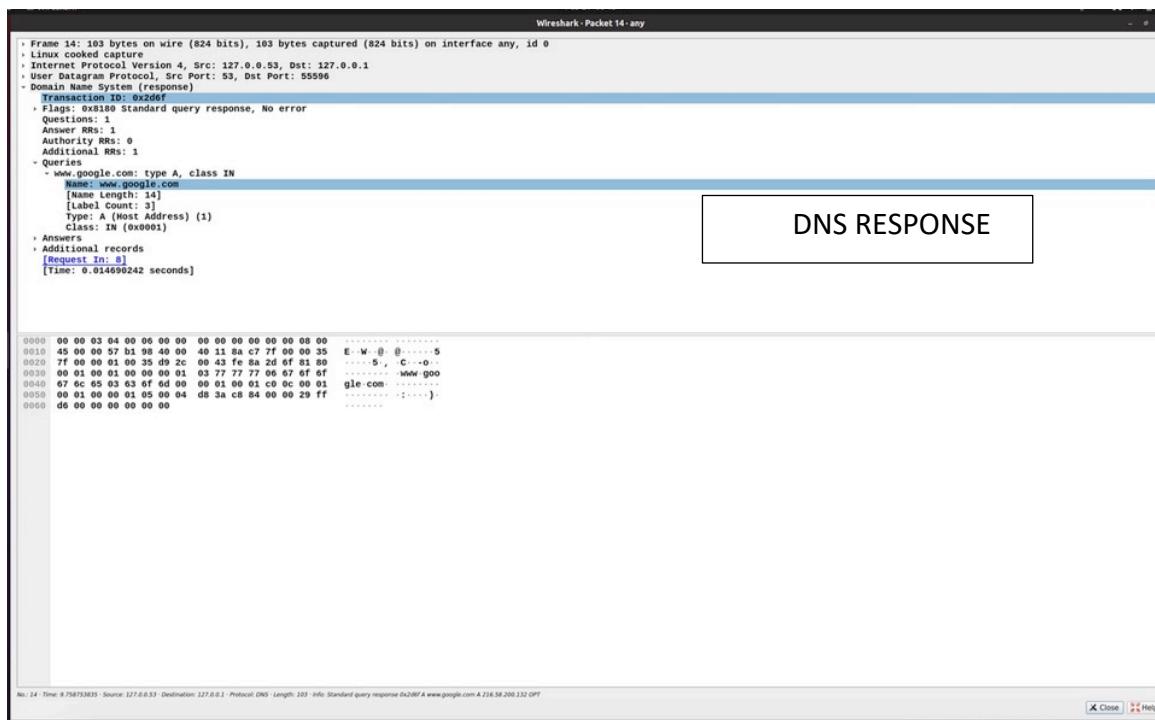
DNS Server : 10.0.5.35

```
 priya@priya-VirtualBox: $ sudo ip addr show
[sudo] password for priya:
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
000
    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:a0:2c:49 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.109/24 brd 192.168.0.255 scope global dynamic noprefixroute enp0s3
        valid lft 6953sec preferred_lft 6953sec
        inet 10.0.5.35/24 scope global enp0s3
            valid lft forever preferred_lft forever
            inet6 fe80::8182:ddc7:13:8f31/64 scope link noprefixroute
                valid lft forever preferred_lft forever
priya@priya-VirtualBox: $
```

OBSERVATION 1 : First Test - Pinging using default DNS

- Wireshark is used to capture the packets in the background while pinging www.google.com
- The IP Address of the Local DNS server is observed to be 127.0.0.53.
- The query is of type A which stands for authoritative. The answer contains the A type record along with the IP address of the website 142.250.76.36
- The first query and authoritative response are shown below





OBSERVATION 2 : Configuring Client Machine

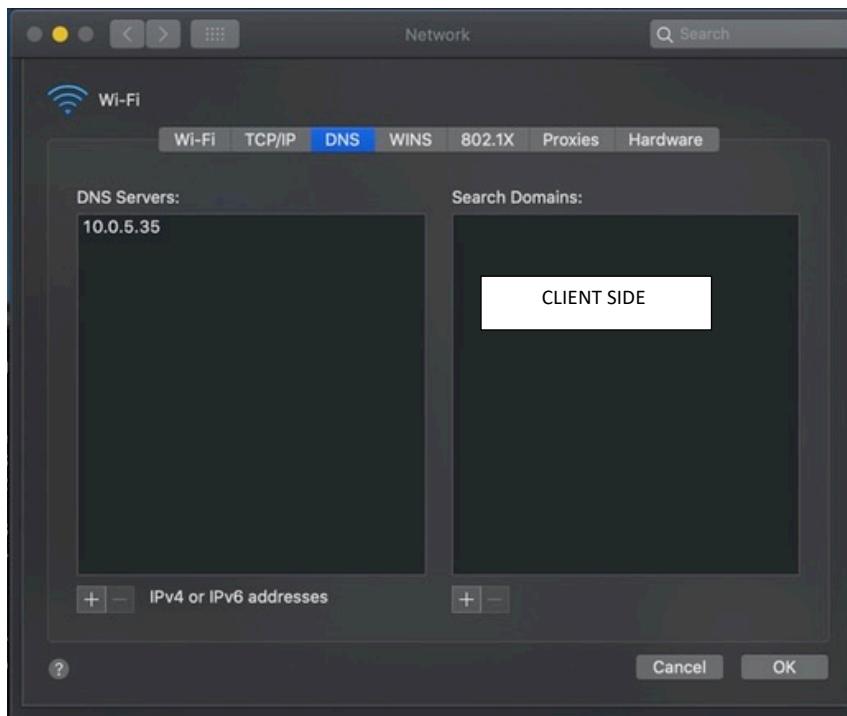
- The IP Address of the client machine is **10.0.5.36** and the IP Address of the server machine is **10.0.5.35**
- We need to add the IP Address of the custom DNS server (10.0.5.35) to the client machine.
- This is done by adding the IP address of the server to the file **/etc/resolvconf/resolv.conf.d/head** 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 theIPv4 Network Settings.
- The changes are applied by using the command **sudo resolvconf -u**



```

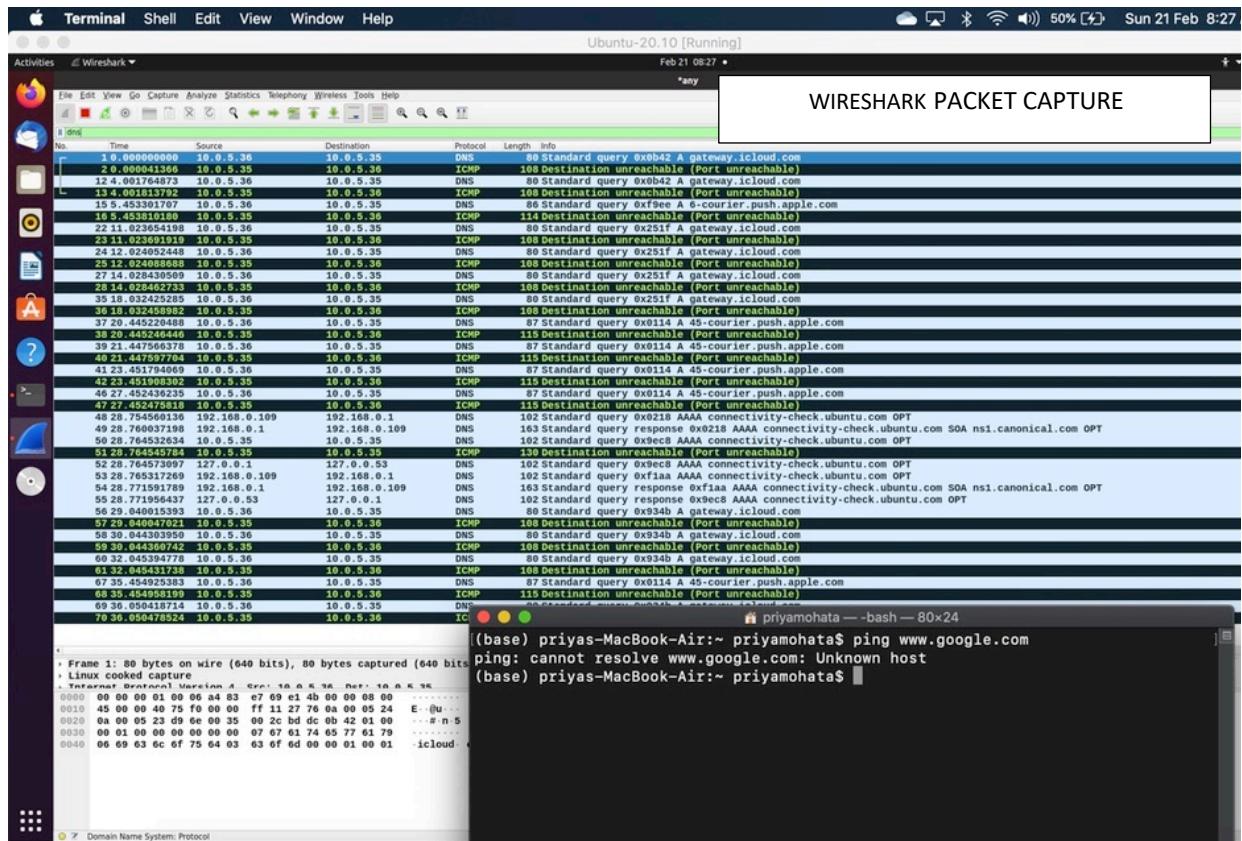
priya@priya-VirtualBox:~$ sudo nano /etc/resolvconf/resolv.conf.d/head
priya@priya-VirtualBox:~$ sudo cat /etc/resolvconf/resolv.conf.d/head
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
# DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
# 127.0.0.53 is the systemd-resolved stub resolver.
# run "systemd-resolve --status" to see details about the actual nameservers.
nameserver 10.0.5.35
priya@priya-VirtualBox:~$
```

SERVER SIDE



OBSERVATION 3 : TESTING PING [WWW.GOOGLE.COM AGAIN](http://www.google.com)

- 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 10.0.5.35 but it does not receive any



OBSERVATION 4 : Setting Up Local DNS Server

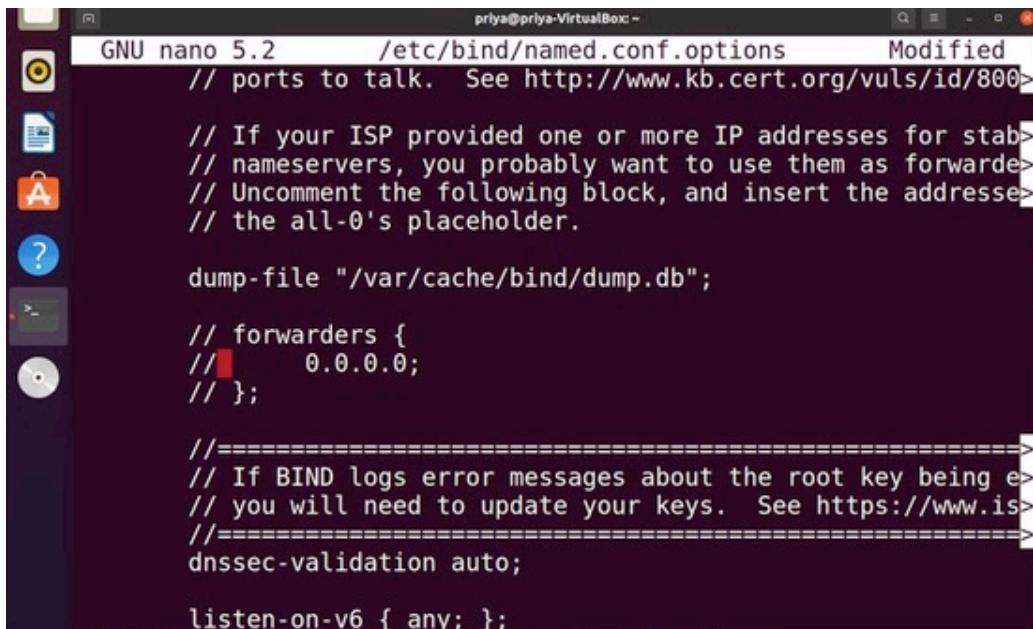
COMMAND USED TO INSTALL BIND9 SERVER

```
$ sudo apt-get update
$ sudo apt-get install bind9
```

```
priya@priya-VirtualBox:~$ sudo apt-get install bind9
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
bind9-dnsutils bind9-libs bind9-utils python3-ply
Suggested packages:
bind-doc python3-ply-doc
The following NEW packages will be installed:
bind9 bind9-utils python3-ply
```

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**.



```
GNU nano 5.2      /etc/bind/named.conf.options      Modified
// ports to talk. See http://www.kb.cert.org/vuls/id/800>
// If your ISP provided one or more IP addresses for stab>
// nameservers, you probably want to use them as forwarder>
// Uncomment the following block, and insert the addressese>
// the all-0's placeholder.

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

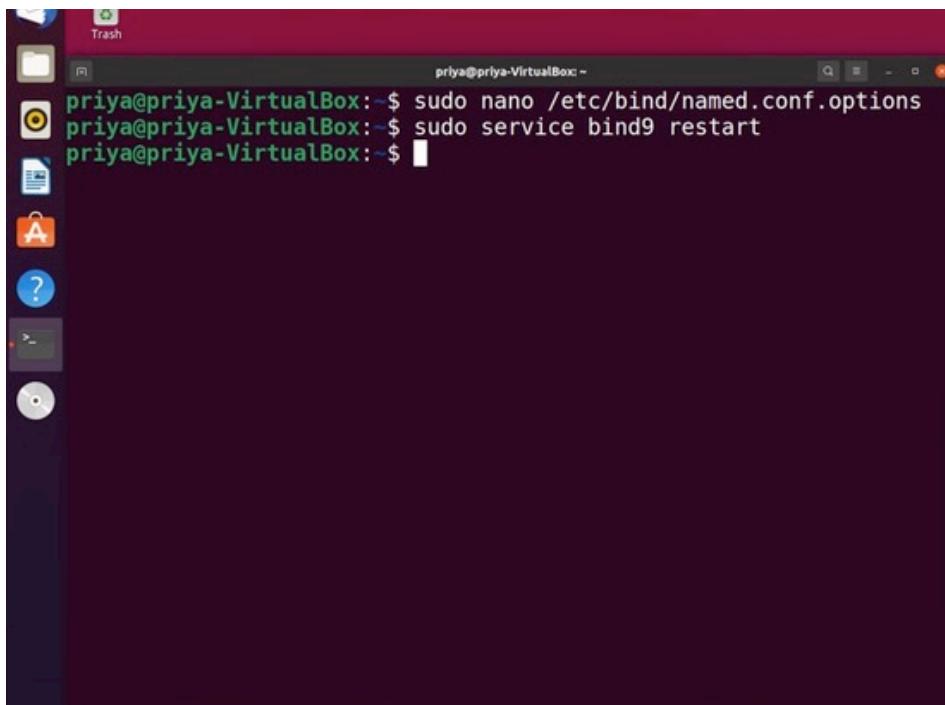
// forwarders {
//   0.0.0.0;
// };

//=====
// If BIND logs error messages about the root key being e>
// you will need to update your keys. See https://www.is>
// =====
dnssec-validation auto;

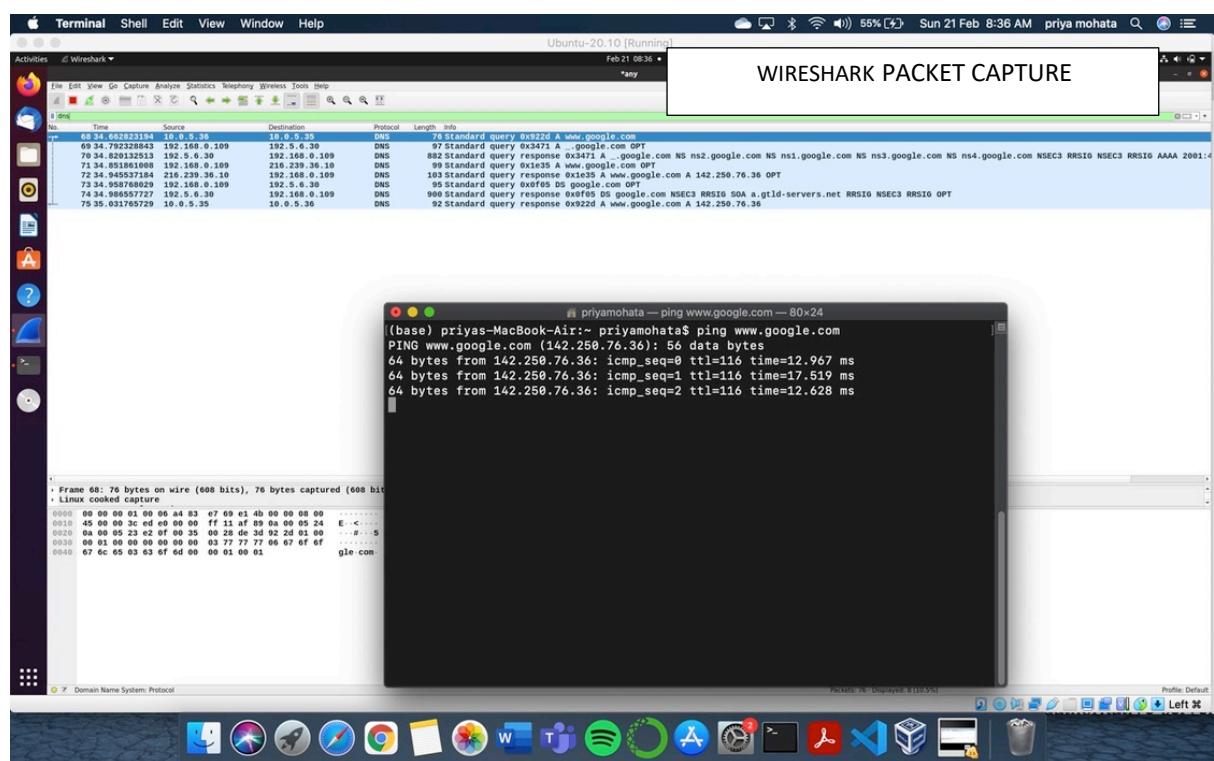
listen-on-v6 { any; };
```

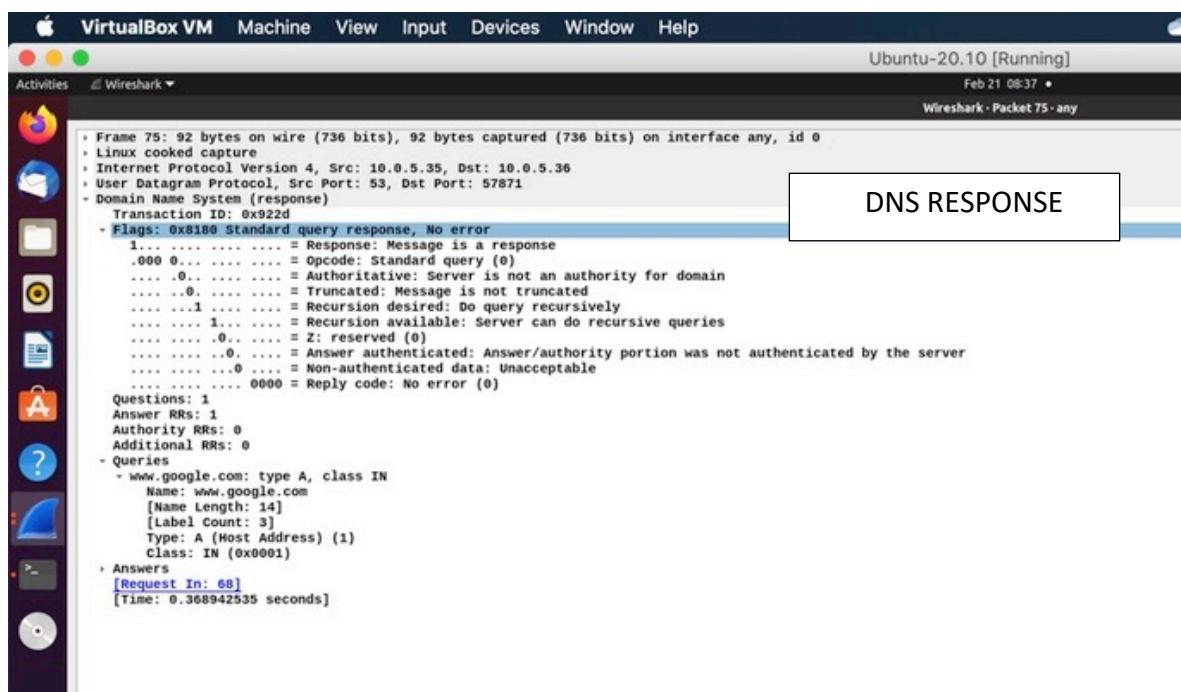
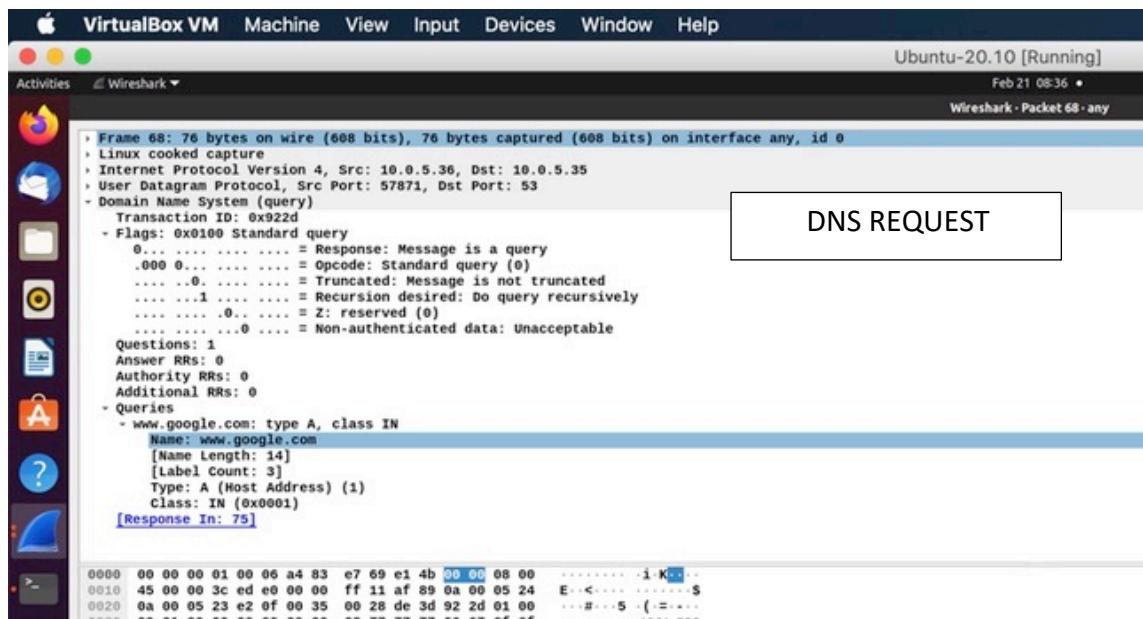
We start the DNS server using the command:

```
$ sudo service bind9 restart
```



We go back to your user machine (10.0.5.36), 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`.

```

VirtualBox VM Machine View Input Devices Window Help
Activities Terminal priya@priya-VirtualBox:~$ sudo rndc dumpdb -cache
priya@priya-VirtualBox:~$ sudo cat /var/cache/bind/dump.db
; Start view _default

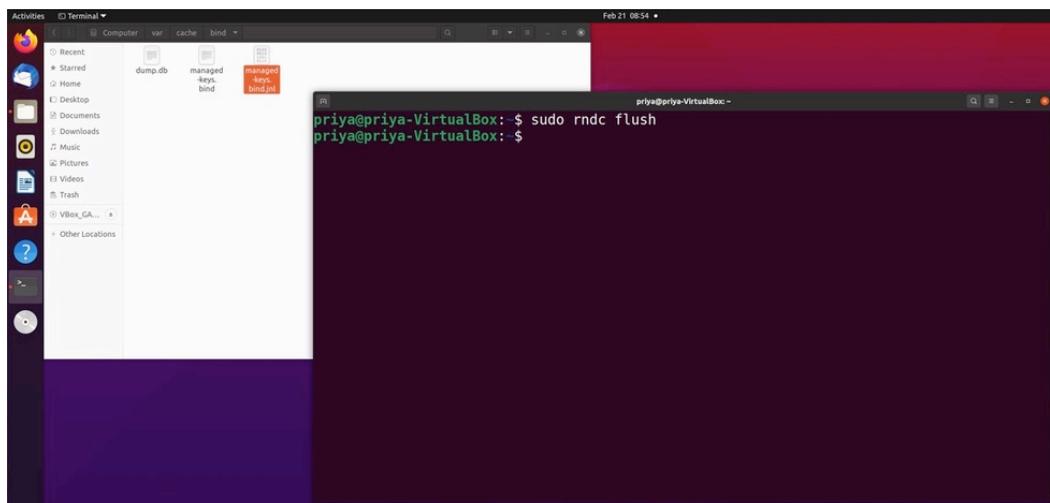
; Cache dump of view '_default' (cache _default)
; using a 43200 second stale ttl
$DATE 20210220151200
; secure
129477 IN SOA a.root-servers.net. ns1d.verisign-grs.com. (
    2021022001 ; serial
    1800        ; refresh (30 minutes)
    900         ; retry (15 minutes)
    604800      ; expire (1 week)
    86400       ; minimum (1 day)
)
; secure
129477 RRSIG SOA 8 0 86400 (
    20210305170000 20210220160000 42351 .
    RNjvPW/4WjR+uvZXUoQSCBC3yI8uS0ULy3q
    axPi5u2Kjpl3de2yiCoXAGoeqKV0/wCvDZB
    4vh1+bTLMJuuvlRYjdnkMyKmJiweBb/6agDJ

Activities Terminal priya@priya-VirtualBox:~$
```

- The linux command to extract cache for www.google.com from dump.db file,
- `sudo cat /var/cache/bind/dump.db | grep "google"`

```

VirtualBox VM Machine View Input Devices Window Help
Activities Terminal priya@priya-VirtualBox:~$ sudo cat /var/cache/bind/dump.db | grep "google"
[sudo] password for priya:
.google.com.          215645  NS      ns1.google.com.
                      215645  NS      ns2.google.com.
                      215645  NS      ns3.google.com.
                      215645  NS      ns4.google.com.
ns1.google.com.        215645  A       216.239.32.10
ns2.google.com.        215645  A       216.239.34.10
ns3.google.com.        215645  A       216.239.36.10
ns4.google.com.        215645  A       216.239.38.10
www.google.com.        43146   A       142.250.76.36
.googleapis.com.       215834   NS     ns1.google.com.
                      215834   NS     ns2.google.com.
                      215834   NS     ns3.google.com.
                      215834   NS     ns4.google.com.
safebrowsing.googleapis.com. 43334 A 216.58.200.138
priya@priya-VirtualBox:~$
```

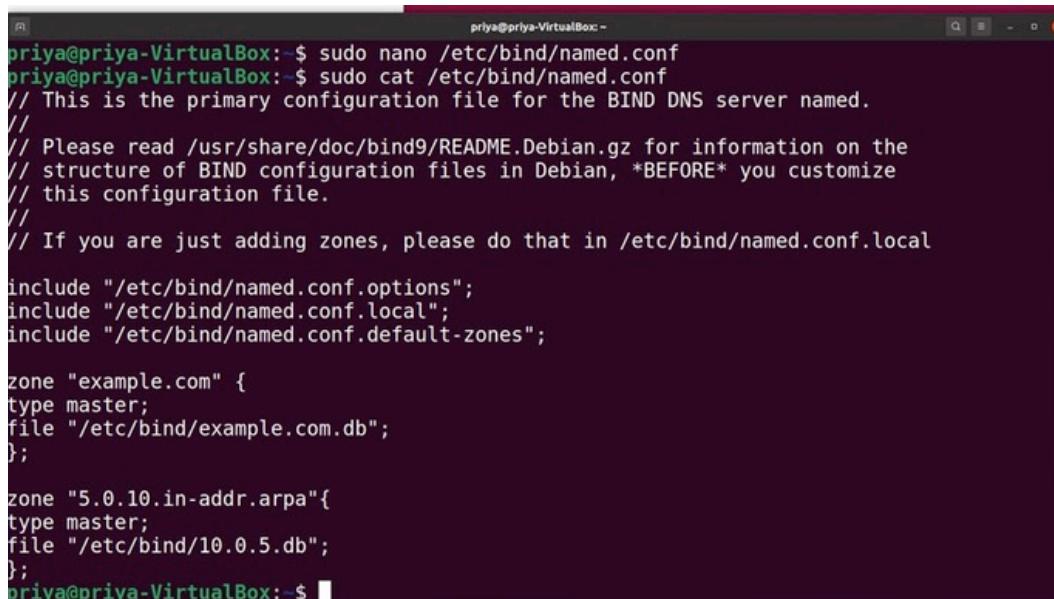


OBSERVATION 6 : Host a Zone in the Local DNS server

- 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).



```

priya@priya-VirtualBox:~$ sudo nano /etc/bind/named.conf
priya@priya-VirtualBox:~$ 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 "5.0.10.in-addr.arpa"{
    type master;
    file "/etc/bind/10.0.5.db";
};
priya@priya-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 **10.0.5.db** zone file with the following contents in the **/etc/bind/** directory where the actual DNS resolution is stored.

```
priya@priya-VirtualBox:~$ sudo cat /etc/bind/example.com.db
[sudo] password for priya:
$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 10.0.5.101
mail IN A 10.0.5.102
ns IN A 10.0.5.10
*.example.com. IN A 10.0.5.100

priya@priya-VirtualBox:~$
```

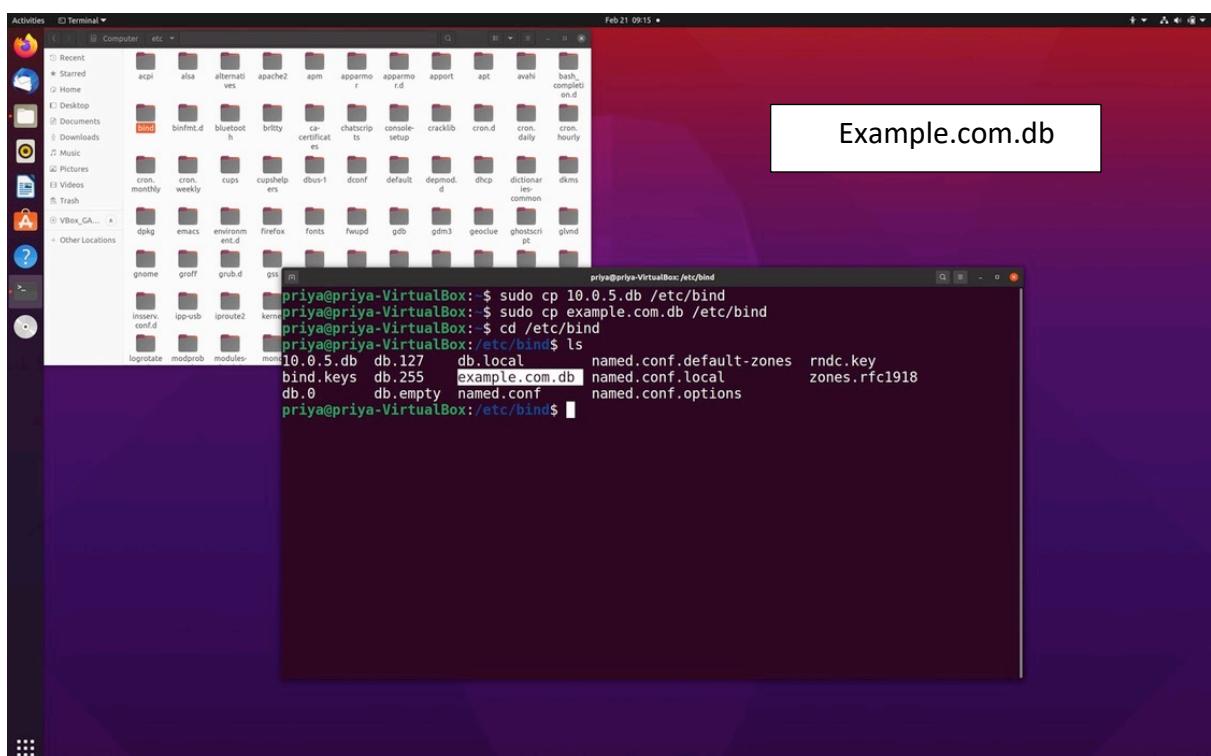
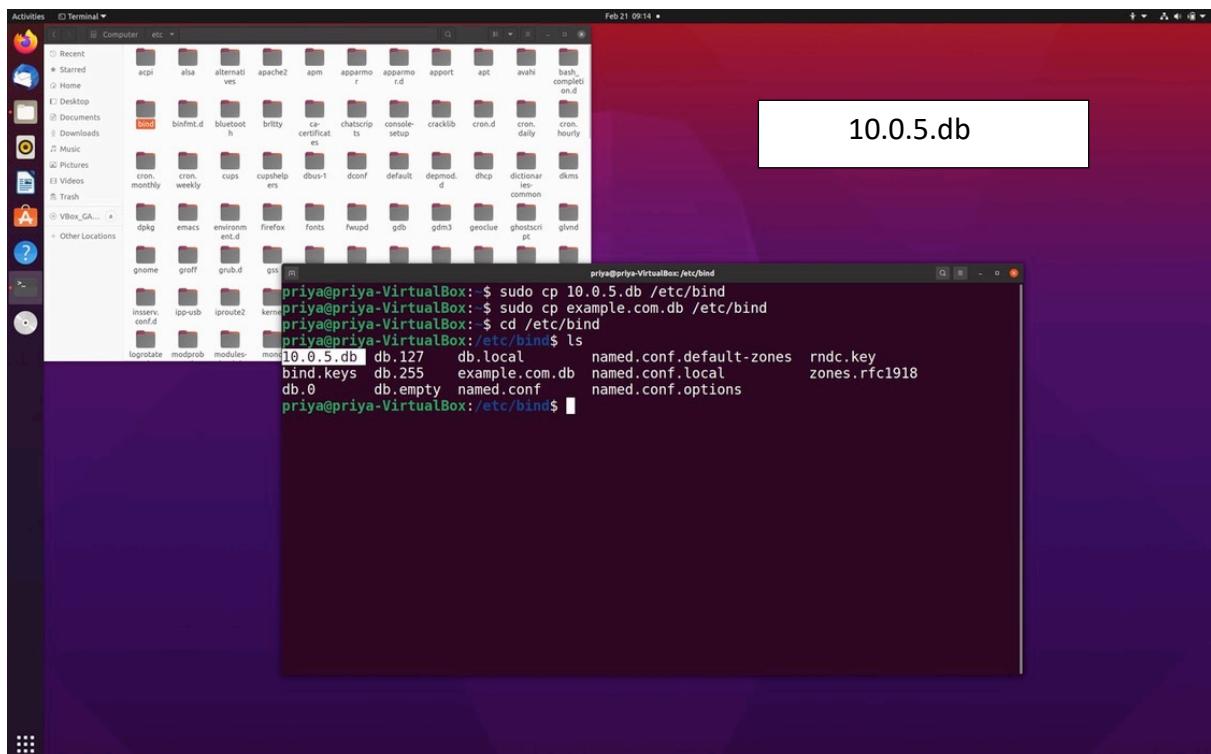
```
priya@priya-VirtualBox:~$ sudo cat /etc/bind/10.0.5.db
$TTL 3D

@ 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.

priya@priya-VirtualBox:~$
```

Step 3: Copy the above files into `/etc/bind` location.



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.

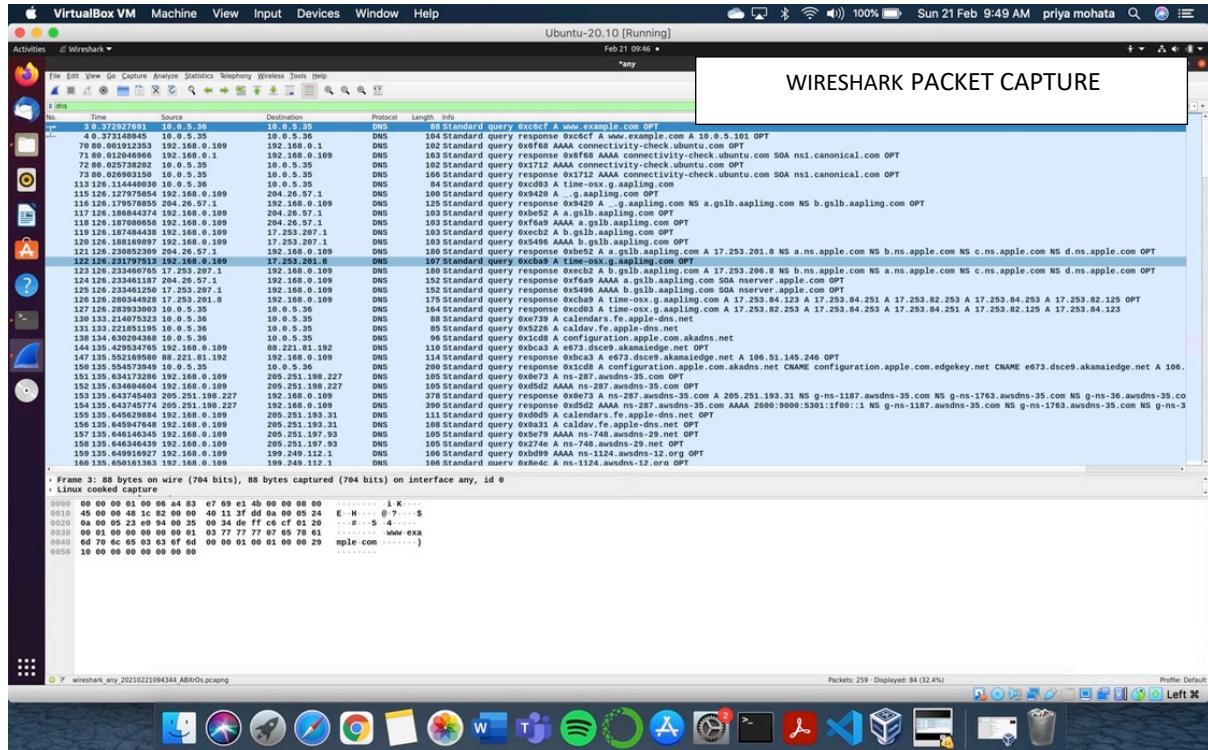
```
(base) priyas-MacBook-Air:~ priyamohata$ dig www.example.com
; <>> Dig 9.10.6 <>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 50895
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

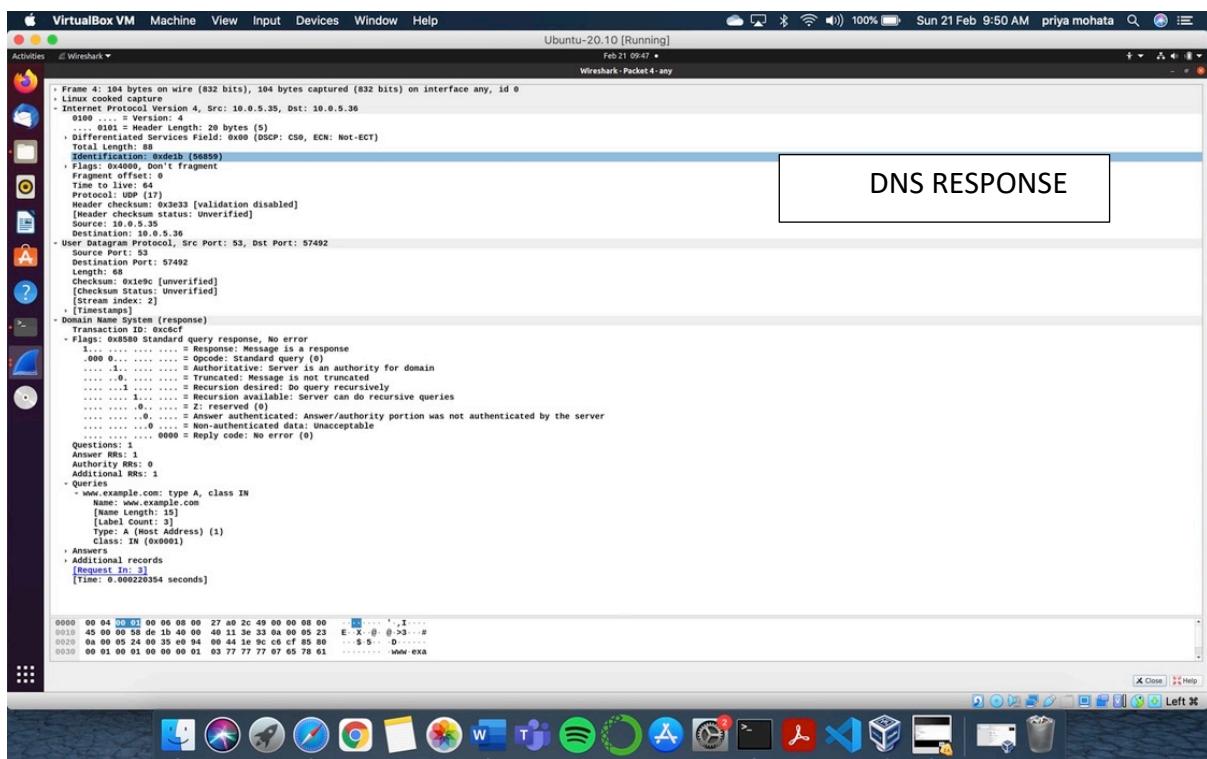
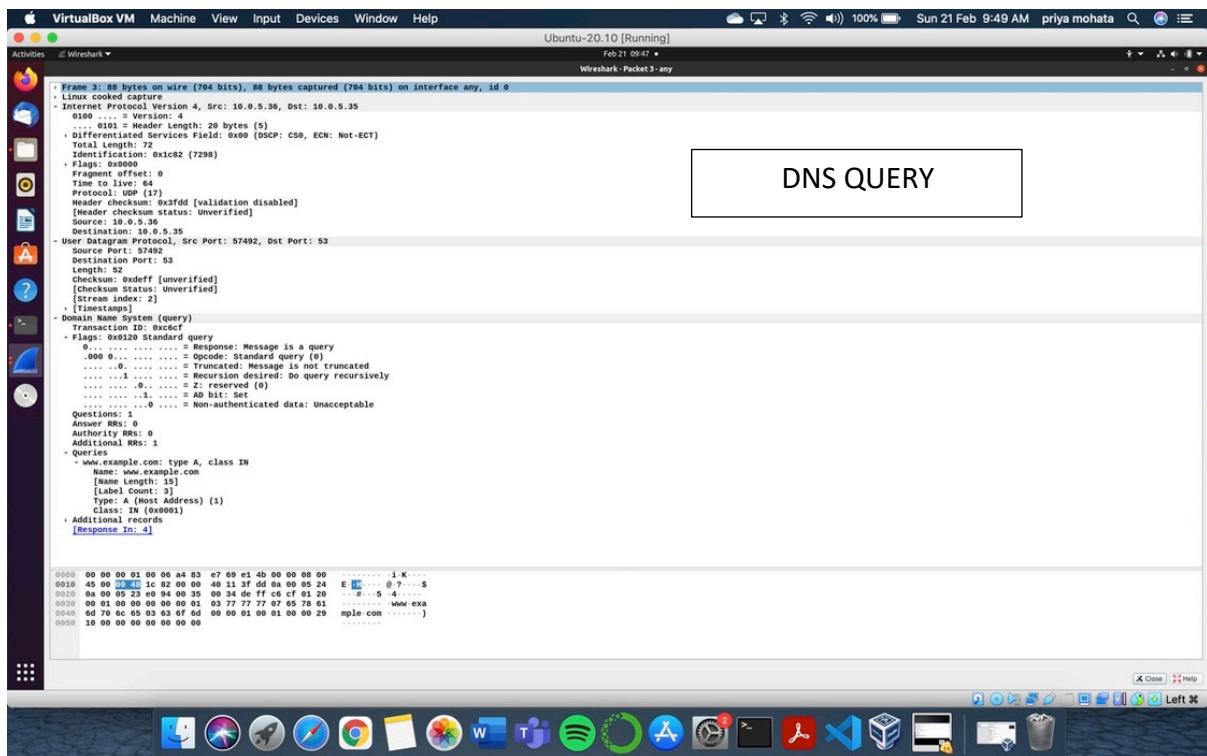
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.example.com.      IN      A

;; ANSWER SECTION:
www.example.com.    259200  IN      A      10.0.5.101

;; Query time: 54 msec
;; SERVER: 10.0.5.35#53(10.0.5.35)
;; WHEN: Sun Feb 21 09:43:48 IST 2021
;; MSG SIZE rcvd: 60

(base) priyas-MacBook-Air:~ priyamohata$
```





QUESTIONS

Q1. Locate the DNS query and response messages. Are they sent over UDP or TCP?

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

Q2. What is the destination port for the DNS query message? What is the source port of DNS response message?

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

Q3. 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?

Answer : The DNS query is made to server at the IP Address 10.0.5.35. This is the same as the local DNS server configured.

Q4. Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"?

Answer :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.

Q5. Examine the DNS response message. How many "answers" are provided? What do each of these answers contain?

Answer :The answer section of the DNS response message contains one Resource Record.

A type RR: This provides the IP Address of the hostname

	RRNAME	TTL	RRTYPE	RRDATA
	; answer			
	www.google.com.	43146	A	142.250.76.36
	; glue			

Q6. 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?

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

