Week #5
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)

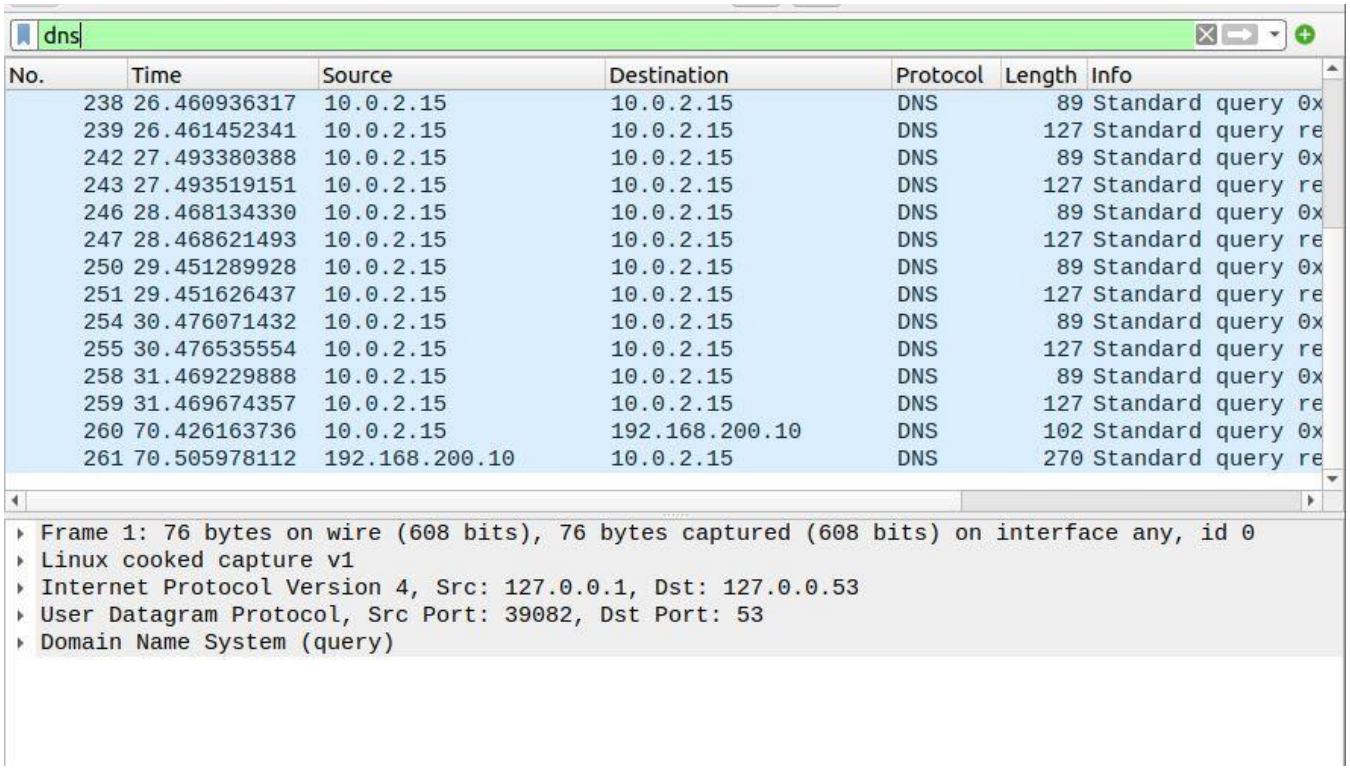
DNS Server: 10.2.22.184 User/Client: 10.2.22.195

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).

```
nagavenigowda@ubuntu-1:~$ ping www.google.com
PING www.google.com (142.250.194.36) 56(84) bytes of data.
64 bytes from del12s02-in-f4.1e100.net (142.250.194.36): icmp_seq=1 ttl=52 time=
849 ms
64 bytes from del12s02-in-f4.1e100.net (142.250.194.36): icmp_seq=2 ttl=52 time=
68.1 ms
64 bytes from del12s02-in-f4.1e100.net (142.250.194.36): icmp_seq=3 ttl=52 time=
110 ms
64 bytes from del12s02-in-f4.1e100.net (142.250.194.36): icmp_seq=4 ttl=52 time=
```



dns

No.	Time	Source	Destination	Protocol	Length	Info
238	26.460936317	10.0.2.15	10.0.2.15	DNS	89	Standard query 0x
239	26.461452341	10.0.2.15	10.0.2.15	DNS	127	Standard query re
242	27.493380388	10.0.2.15	10.0.2.15	DNS	89	Standard query 0x
243	27.493519151	10.0.2.15	10.0.2.15	DNS	127	Standard query re
246	28.468134330	10.0.2.15	10.0.2.15	DNS	89	Standard query 0x
247	28.468621493	10.0.2.15	10.0.2.15	DNS	127	Standard query re
250	29.451289928	10.0.2.15	10.0.2.15	DNS	89	Standard query 0x
251	29.451626437	10.0.2.15	10.0.2.15	DNS	127	Standard query re
254	30.476071432	10.0.2.15	10.0.2.15	DNS	89	Standard query 0x
255	30.476535554	10.0.2.15	10.0.2.15	DNS	127	Standard query re
258	31.469229888	10.0.2.15	10.0.2.15	DNS	89	Standard query 0x
259	31.469674357	10.0.2.15	10.0.2.15	DNS	127	Standard query re
260	70.426163736	10.0.2.15	192.168.200.10	DNS	102	Standard query 0x
261	70.505978112	192.168.200.10	10.0.2.15	DNS	270	Standard query re

```
▸ Frame 1: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface any, id 0
▸ Linux cooked capture v1
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.53
▸ User Datagram Protocol, Src Port: 39082, Dst Port: 53
▸ Domain Name System (query)
```

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

```
  GNU nano 6.2                          /etc/resolv.conf
# 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 10.0.2.15
nameserver 127.0.0.53
```

/etc/resolvconf/resolv.conf.d/head file.

nameserver 10.2.22.184

```
  GNU nano 6.2                  /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 10.0.2.15
```

Run the following command for the change to take effect.

sudo resolvconf -u

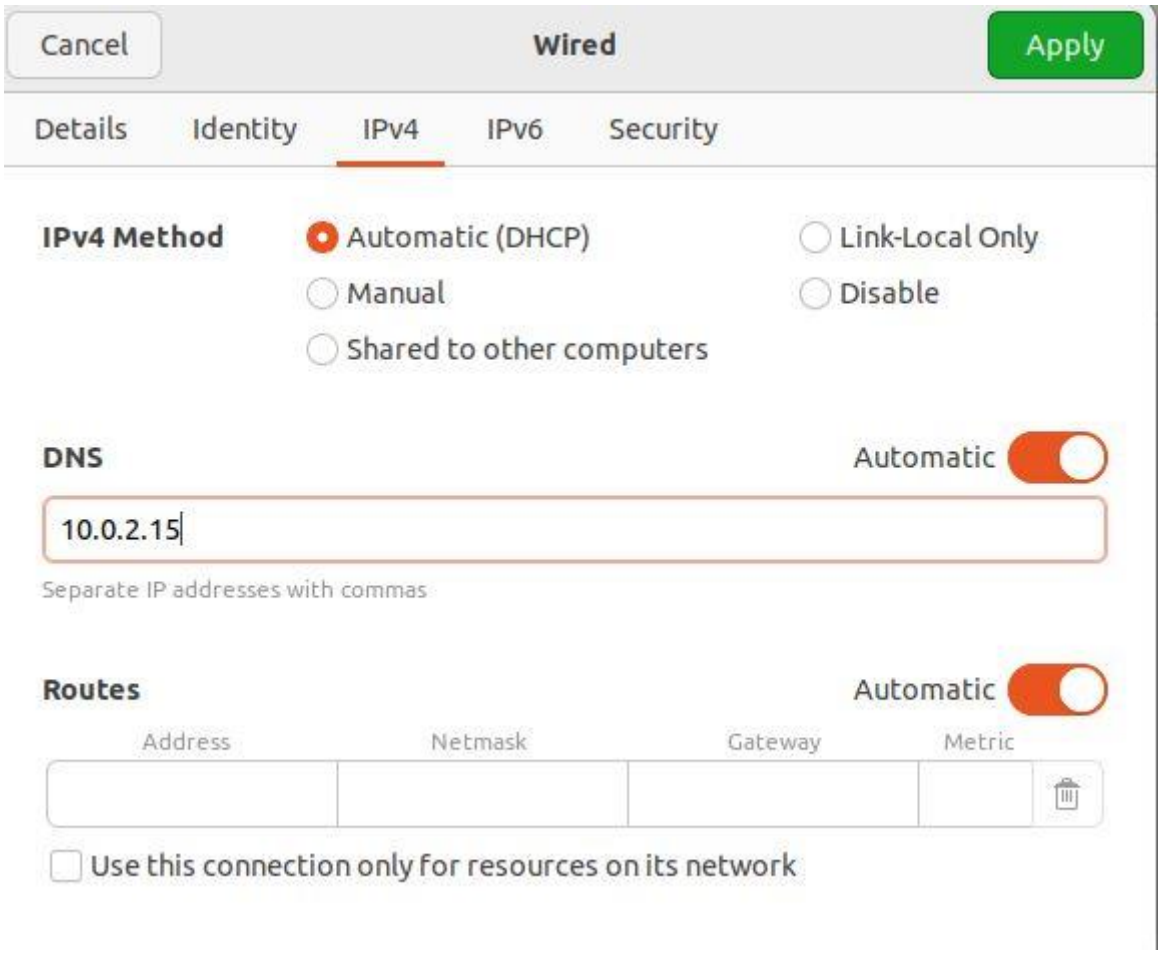
```
root@ubuntu-1: ~
root@ubuntu-1:~# sudo nano /etc/resolvconf/resolv.conf.d/head
root@ubuntu-1:~# sudo nano /etc/resolv.conf
root@ubuntu-1:~# sudo nano /etc/resolvconf/resolv.conf.d/head
root@ubuntu-1:~# sudo nano /etc/resolv.conf
root@ubuntu-1:~# sudo nano /etc/resolvconf/resolv.conf.d/head
root@ubuntu-1:~# sudo resolvconf -u
root@ubuntu-1:~#
```

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

```
root@ubuntu-1:~# 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 10.0.2.15
root@ubuntu-1:~#
```

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).

```
nagavenigowda@ubuntu-1:~$ ping www.google.com
PING www.google.com (142.250.194.36) 56(84) bytes of data.
64 bytes from del12s02-in-f4.1e100.net (142.250.194.36): icmp_seq=1 ttl=52 time=
849 ms
64 bytes from del12s02-in-f4.1e100.net (142.250.194.36): icmp_seq=2 ttl=52 time=
68.1 ms
64 bytes from del12s02-in-f4.1e100.net (142.250.194.36): icmp_seq=3 ttl=52 time=
110 ms
64 bytes from del12s02-in-f4.1e100.net (142.250.194.36): icmp_seq=4 ttl=52 time=
```

dns

No.	Time	Source	Destination	Protocol	Length	Info
238	26.460936317	10.0.2.15	10.0.2.15	DNS	89	Standard query 0x
239	26.461452341	10.0.2.15	10.0.2.15	DNS	127	Standard query re
242	27.493380388	10.0.2.15	10.0.2.15	DNS	89	Standard query 0x
243	27.493519151	10.0.2.15	10.0.2.15	DNS	127	Standard query re
246	28.468134330	10.0.2.15	10.0.2.15	DNS	89	Standard query 0x
247	28.468621493	10.0.2.15	10.0.2.15	DNS	127	Standard query re
250	29.451289928	10.0.2.15	10.0.2.15	DNS	89	Standard query 0x
251	29.451626437	10.0.2.15	10.0.2.15	DNS	127	Standard query re
254	30.476071432	10.0.2.15	10.0.2.15	DNS	89	Standard query 0x
255	30.476535554	10.0.2.15	10.0.2.15	DNS	127	Standard query re
258	31.469229888	10.0.2.15	10.0.2.15	DNS	89	Standard query 0x
259	31.469674357	10.0.2.15	10.0.2.15	DNS	127	Standard query re
260	70.426163736	10.0.2.15	192.168.200.10	DNS	102	Standard query 0x
261	70.505978112	192.168.200.10	10.0.2.15	DNS	270	Standard query re

```
▸ Frame 1: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface any, id 0
▸ Linux cooked capture v1
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.53
▸ User Datagram Protocol, Src Port: 39082, Dst Port: 53
▸ Domain Name System (query)
```

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

```
root@ubuntu-1: ~
root@ubuntu-1:~# sudo apt-get install bind9
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
bind9 is already the newest version (1:9.18.1-1ubuntu1.3).
The following package was automatically installed and is no longer required:
  systemd-hwe-hwdb
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 265 not upgraded.
root@ubuntu-1:~# sudo apt-get update
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:2 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [107 kB]
Fetched 336 kB in 7s (45.2 kB/s)
Reading package lists... Done
root@ubuntu-1:~#
```

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.

```
root@ubuntu-1: ~
  GNU nano 6.2                    /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
        //========================================================================
        dnssec-validation auto;
```

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**.

```
root@ubuntu-1:~# cat /var/cache/bind/dump.db
;
; Start view _default
;
;
; Cache dump of view '_default' (cache _default)
;
; using a 0 second stale ttl
$DATE 20230223094509
; secure
.                       518391  IN NS    a.root-servers.net.
                        518391  IN NS    b.root-servers.net.
                        518391  IN NS    c.root-servers.net.
                        518391  IN NS    d.root-servers.net.
                        518391  IN NS    e.root-servers.net.
                        518391  IN NS    f.root-servers.net.
                        518391  IN NS    g.root-servers.net.
                        518391  IN NS    h.root-servers.net.
                        518391  IN NS    i.root-servers.net.
                        518391  IN NS    j.root-servers.net.
                        518391  IN NS    k.root-servers.net.
                        518391  IN NS    l.root-servers.net.
                        518391  IN NS    m.root-servers.net.
; secure
                        518391  RRSIG    NS 8 0 518400 (
                                         20230308050000 20230223040000 951 .
                                         gYpGnWJim88jzPX9TbTk1x0+bDUWg6FI4Qcb
                                         fzl4wO/NeyF7Sbayjm7OoV2wQzlacKQm8F+8
                                         GChw1Tj/mkAbxCcR75zFrD+a7GBLVlj7X2vo
                                         RvrYWqxZEUGe37bfcncWvgWjzknfxMw5d3Fj
                                         V7jr8fpU5kWT6mcMlYpPKz+TN7B76550Vv0H
                                         IN3N8QnpYSy4z9M1WctCWe95XGOW8Oy2NUAN
                                         1BjjeJWJmjAwZmsePOAOAmpATzIXd1cXJ1Wm
                                         b1P1kQP7ewDRpfctA+zFEyofhoAfeWsV7xTs
                                         /7UFFTstKNiLjcYb2l9zEG79JOGMl3bGQ1/Q
                                         aq4nFCuOU6Ut2XyZKg== )
; secure
                        172791  DNSKEY   256 3 8 (
```

Step 2: Start DNS server

We start the DNS server using the command:

$ sudo service bind9 restart

```
root@ubuntu-1: ~
root@ubuntu-1:~# sudo nano /etc/bind/named.conf.options
root@ubuntu-1:~# sudo nano /etc/bind/named.conf.options
root@ubuntu-1:~# sudo service bind9 restart
root@ubuntu-1:~#
```

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.

```
root@ubuntu-1:~# sudo nano /etc/bind/named.conf.options
root@ubuntu-1:~# sudo service bind9 start
root@ubuntu-1:~# sudo rndc dumpdb -cache
root@ubuntu-1:~# sudo service bind9 start
root@ubuntu-1:~# sudo service bind9 restart
root@ubuntu-1:~# sudo rndc dumpdb -cache
root@ubuntu-1:~# sudo rndc flush
root@ubuntu-1:~#
root@ubuntu-1:~#
```

dns

No.	Time	Source	Destination	Protocol	Length	Info
402	71.503026301	10.0.2.15	10.0.2.15	DNS	101	Standard query 0x
403	71.503452018	10.0.2.15	10.0.2.15	DNS	139	Standard query re
406	72.103310102	10.0.2.15	10.0.2.15	DNS	101	Standard query 0x
407	72.103624124	10.0.2.15	10.0.2.15	DNS	139	Standard query re
410	73.073727315	10.0.2.15	10.0.2.15	DNS	101	Standard query 0x
411	73.074289175	10.0.2.15	10.0.2.15	DNS	139	Standard query re
414	74.101601473	10.0.2.15	10.0.2.15	DNS	101	Standard query 0x
415	74.101732009	10.0.2.15	10.0.2.15	DNS	139	Standard query re
418	75.079065796	10.0.2.15	10.0.2.15	DNS	101	Standard query 0x
419	75.079438766	10.0.2.15	10.0.2.15	DNS	139	Standard query re
422	76.101592406	10.0.2.15	10.0.2.15	DNS	101	Standard query 0x
423	76.101818897	10.0.2.15	10.0.2.15	DNS	139	Standard query re
426	77.113525130	10.0.2.15	10.0.2.15	DNS	101	Standard query 0x
427	77.113652374	10.0.2.15	10.0.2.15	DNS	139	Standard query re

```
▸ Frame 166: 101 bytes on wire (808 bits), 101 bytes captured (808 bits) on interface any, id 0
▸ Linux cooked capture v1
▸ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.2.15
▸ User Datagram Protocol, Src Port: 43081, Dst Port: 53
▸ Domain Name System (query)
```

```
0000  00 00 03 04 00 06 00 00  00 00 00 00 a1 6d 08 00   ........ .....m..
0010  45 00 00 55 ff 00 40 00  40 11 23 7a 0a 00 02 0f   E..U..@. @.#z....
0020  0a 00 02 0f a8 49 00 35  00 41 18 70 df 0c 01 20   .....I.5 .A.p...
0030  00 01 00 00 00 00 00 01  03 31 30 30 03 31 38 32   ........ .100.182
0040  03 32 35 30 03 31 34 32  07 69 6e 2d 61 64 64 72   .250.142 .in-addr
0050  04 61 72 70 61 00 00 0c  00 01 00 00 29 04 b0 00   .arpa... ....)...
0060  00 00 00 00 00                                     .....
```

```
dns
▸ Frame 506: 101 bytes on wire (808 bits), 101 bytes captured (808 bits) on interface any, id 0
▸ Linux cooked capture v1
▾ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.2.15
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 85
    Identification: 0xd17a (53626)
  ▸ Flags: 0x40, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: UDP (17)
    Header Checksum: 0x5100 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.0.2.15
    Destination Address: 10.0.2.15
▾ User Datagram Protocol, Src Port: 53886, Dst Port: 53
    Source Port: 53886
    Destination Port: 53
    Length: 65
    Checksum: 0x1870 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 147]
  ▸ [Timestamps]
    UDP payload (57 bytes)
▾ Domain Name System (query)
    Transaction ID: 0xdf11
  ▸ Flags: 0x0120 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 1
  ▸ Queries
  ▸ Additional records
    [Response In: 507]
```

```
root@ubuntu-1:~# sudo rndc dumpdb -cache
root@ubuntu-1:~# cat /var/cache/bind/dump.db | grep *google *
grep: snap: Is a directory
root@ubuntu-1:~# cat /var/cache/bind/dump.db | grep "google"
250.142.in-addr.arpa.   85986   NS      ns1.google.com.
                        85986   NS      ns2.google.com.
                        85986   NS      ns3.google.com.
                        85986   NS      ns4.google.com.
google.com.             172380  NS      ns1.google.com.
                        172380  NS      ns2.google.com.
                        172380  NS      ns3.google.com.
                        172380  NS      ns4.google.com.
ns1.google.com.         172380  A       216.239.32.10
ns2.google.com.         172380  A       216.239.34.10
ns3.google.com.         172380  A       216.239.36.10
ns4.google.com.         172380  A       216.239.38.10
root@ubuntu-1:~#
```

```
; glue
ns1.google.com.          172380  A       216.239.32.10
; glue
                         172380  AAAA    2001:4860:4802:32::a
; glue
ns2.google.com.          172380  A       216.239.34.10
; glue
                         172380  AAAA    2001:4860:4802:34::a
; glue
ns3.google.com.          172380  A       216.239.36.10
; glue
                         172380  AAAA    2001:4860:4802:36::a
; glue
ns4.google.com.          172380  A       216.239.38.10
; glue
                         172380  AAAA    2001:4860:4802:38::a
```

dns

dns
dnsserver

No.		Source	Destination	Protocol	Length	Info
	749968	10.0.2.15	10.0.2.15	DNS	139	Standard query re
157	11.776733206	10.0.2.15	10.0.2.15	DNS	139	Standard query re
158	11.777045467	10.0.2.15	10.0.2.15	ICMP	167	Destination unrea
159	11.777090581	10.0.2.15	10.0.2.15	DNS	139	Standard query re
162	11.880375278	10.0.2.15	10.0.2.15	DNS	101	Standard query 0x
163	11.881070087	10.0.2.15	10.0.2.15	DNS	139	Standard query re
166	12.915984789	10.0.2.15	10.0.2.15	DNS	101	Standard query 0x
167	12.916511382	10.0.2.15	10.0.2.15	DNS	139	Standard query re
	70674	10.0.2.15	10.0.2.15	DNS	101	Standard query 0x
	00936	10.0.2.15	10.0.2.15	DNS	139	Standard query re
174	14.848253350	10.0.2.15	10.0.2.15	DNS	101	Standard query 0x
175	14.848610710	10.0.2.15	10.0.2.15	DNS	139	Standard query re
178	16.029052173	10.0.2.15	10.0.2.15	DNS	101	Standard query 0x
179	16.029638150	10.0.2.15	10.0.2.15	DNS	139	Standard query re
182	17.000888852	10.0.2.15	10.0.2.15	DNS	101	Standard query 0x
183	17.001400090	10.0.2.15	10.0.2.15	DNS	139	Standard query re
186	18.062708111	10.0.2.15	10.0.2.15	DNS	101	Standard query 0x
187	18.063659845	10.0.2.15	10.0.2.15	DNS	139	Standard query re
190	19.049732746	10.0.2.15	10.0.2.15	DNS	101	Standard query 0x
191	19.050141740	10.0.2.15	10.0.2.15	DNS	139	Standard query re
194	20.060300824	10.0.2.15	10.0.2.15	DNS	101	Standard query 0x
195	20.060942956	10.0.2.15	10.0.2.15	DNS	139	Standard query re
198	20.861862225	10.0.2.15	10.0.2.15	DNS	101	Standard query 0x
199	20.862212127	10.0.2.15	10.0.2.15	DNS	139	Standard query re
202	21.850337448	10.0.2.15	10.0.2.15	DNS	101	Standard query 0x
203	21.850630984	10.0.2.15	10.0.2.15	DNS	139	Standard query re
206	23.084470323	10.0.2.15	10.0.2.15	DNS	101	Standard query 0x
207	23.084884636	10.0.2.15	10.0.2.15	DNS	139	Standard query re
210	23.872560171	10.0.2.15	10.0.2.15	DNS	101	Standard query 0x
211	23.873002883	10.0.2.15	10.0.2.15	DNS	139	Standard query re
214	24.848687892	10.0.2.15	10.0.2.15	DNS	101	Standard query 0x

LibreOffice Writer

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).

```
  GNU nano 6.2                          /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 "2.0.10.in-addr.arpa" {
type master;
file "etc/bind/10.0.2.db";
};
```

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.

```
root@ubuntu-1:~# cat /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       10.0.2.101
mail    IN      A       10.0.2.102
ns      IN      A       10.0.2.10
*.example.com.  IN      A 10.0.2.100
root@ubuntu-1:~#
```

```
root@ubuntu-1:~# cat  /etc/bind/10.0.2.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.
root@ubuntu-1:~#
root@ubuntu-1:~#
root@ubuntu-1:~#
```

```
$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.

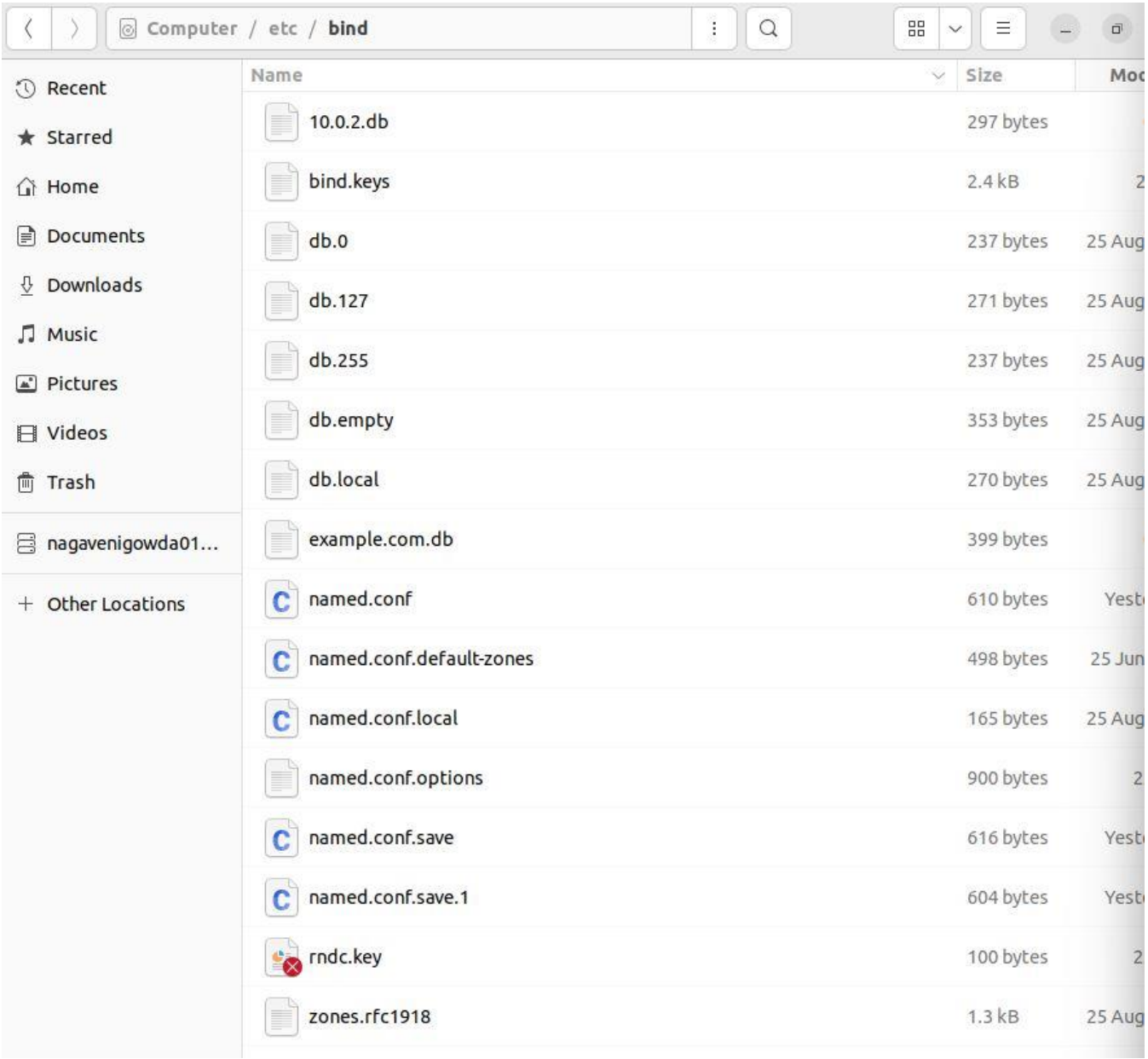
```
$TTL 3D
@     IN    SOA   ns.example.com. admin.example.com. (
            2008111001
            8H
            2H
            4W
            1D)
@     IN    NS    ns.example.com.

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.

```
root@ubuntu-1:~# sudo cp 10.0.2.db /etc/bind
root@ubuntu-1:~# sudo cp example.com.db /etc/bind
root@ubuntu-1:~#
```



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.

```
root@ubuntu-1:~# sudo service bind9 restart

root@ubuntu-1:~# sudo service bind9 restart
root@ubuntu-1:~# dig www.example.com

; <<>> DiG 9.18.1-1ubuntu1.3-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 755
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 71f12181bbbe1f4d0100000006400215482cbf55e0788c6c5 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        259200  IN      A       10.0.2.101

;; Query time: 0 msec
;; SERVER: 10.0.2.15#53(10.0.2.15) (UDP)
;; WHEN: Thu Mar 02 09:38:52 IST 2023
;; MSG SIZE  rcvd: 88

root@ubuntu-1:~#
```

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.

```
root@ubuntu-1:~# cat /var/cache/bind/dump.db | grep "example"
root@ubuntu-1:~# sudo service bind9 restart
root@ubuntu-1:~# sudo rndc dumpdb -cache
root@ubuntu-1:~# dig www.example.com
```

Step 3: Observe the results in Wireshark capture.

dns

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	192.168.211.195	DNS	102	Standard query 0x2a6e A connectivity-check.ubuntu.com…
2	0.405908082	192.168.211.195	10.0.2.15	DNS	246	Standard query response 0x2a6e A connectivity-check.u…
13	79.810464103	10.0.2.15	10.0.2.15	DNS	101	Standard query 0x2bf4 A mail.example.com OPT
14	79.812269122	10.0.2.15	10.0.2.15	DNS	133	Standard query response 0x2bf4 A mail.example.com A 1…
15	105.398005830	10.0.2.15	10.0.2.15	DNS	100	Standard query 0xf335 A www.example.com OPT
16	105.398434350	10.0.2.15	10.0.2.15	DNS	132	Standard query response 0xf335 A www.example.com A 10…
20	200.751306784	10.0.2.15	192.168.211.195	DNS	102	Standard query 0x1a1f AAAA connectivity-check.ubuntu.…
21	201.014544975	192.168.211.195	10.0.2.15	DNS	270	Standard query response 0x1a1f AAAA connectivity-chec…
24	300.083293769	10.0.2.15	192.168.211.195	DNS	102	Standard query 0x898b A connectivity-check.ubuntu.com…
25	300.145636182	192.168.211.195	10.0.2.15	DNS	246	Standard query response 0x898b A connectivity-check.u…

```
    [Checksum Status: Unverified]
    [Stream index: 2]
  ▸ [Timestamps]
    UDP payload (56 bytes)
▾ Domain Name System (query)
    Transaction ID: 0xf335
  ▾ Flags: 0x0120 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ..1. .... = AD bit: Set
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 1
  ▾ Queries
    ▾ www.example.com: type A, class IN
        Name: www.example.com
        [Name Length: 15]
        [Label Count: 3]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
  ▾ Additional records
    ▸ <Root>: type OPT
    [Response In: 16]
```

MAIL

```
root@ubuntu-1:~# dig mail.example.com

; <<>> DiG 9.18.1-1ubuntu1.3-Ubuntu <<>> mail.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61108
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 56964c6176f4ea5e01000000640027a694e3750ecdc3855b (good)
;; QUESTION SECTION:
;mail.example.com.              IN      A

;; ANSWER SECTION:
mail.example.com.       259200  IN      A       10.0.2.102

;; Query time: 0 msec
;; SERVER: 10.0.2.15#53(10.0.2.15) (UDP)
;; WHEN: Thu Mar 02 10:05:50 IST 2023
;; MSG SIZE  rcvd: 89

root@ubuntu-1:~#
```

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	10.0.2.15	DNS	100	Standard query 0x9d8c A www.example.com OPT
2	0.004124894	10.0.2.15	10.0.2.15	DNS	132	Standard query response 0x9d8c A www.example.com A 10…
3	56.856710640	10.0.2.15	192.168.211.195	DNS	102	Standard query 0x6721 AAAA connectivity-check.ubuntu.…
4	57.043760682	192.168.211.195	10.0.2.15	DNS	270	Standard query response 0x6721 AAAA connectivity-chec…
5	57.765005669	10.0.2.15	10.0.2.15	DNS	101	Standard query 0xeeb4 A mail.example.com OPT
6	57.765571443	10.0.2.15	10.0.2.15	DNS	133	Standard query response 0xeeb4 A mail.example.com A 1…
7	61.868371851	PcsCompu_64:26:6f		ARP	44	Who has 10.0.2.2? Tell 10.0.2.15
8	61.868732271	RealtekU_12:35:02		ARP	62	10.0.2.2 is at 52:54:00:12:35:02

```
▸ Frame 5: 101 bytes on wire (808 bits), 101 bytes captured (808 bits) on interface any, id 0
▸ Linux cooked capture v1
▸ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.2.15
▾ User Datagram Protocol, Src Port: 42120, Dst Port: 53
    Source Port: 42120
    Destination Port: 53
    Length: 65
    Checksum: 0x1870 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 2]
  ▸ [Timestamps]
    UDP payload (57 bytes)
▾ Domain Name System (query)
    Transaction ID: 0xeeb4
  ▸ Flags: 0x0120 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 1
  ▾ Queries
    ▸ mail.example.com: type A, class IN
  ▾ Additional records
    ▸ <Root>: type OPT
    [Response In: 6]
```

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

```
root@ubuntu-1:~# sudo rndc dumpdb -cache
root@ubuntu-1:~# sudo rndc flush
root@ubuntu-1:~#
```

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?

   ---- UDP

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

   ---53—dst port of query message and 53 is the source port of response msg

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?

   ---10.0.2.15

   Yes IP ADDRESSES ARE SAME

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

   Standard query

   NO

   Answers RR =0

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

```
▾ Domain Name System (response)
    Transaction ID: 0x3562
  ▸ Flags: 0x8000 Standard query response, No error
    Questions: 1
    Answer RRs: 0
    Authority RRs: 8
    Additional RRs: 3
  ▾ Queries
    ▸ _.flipkart.com: type A, class IN
  ▾ Authoritative nameservers
    ▸ flipkart.com: type NS, class IN, ns sdns14.ultradns.com
    ▸ flipkart.com: type NS, class IN, ns sdns14.ultradns.net
    ▸ flipkart.com: type NS, class IN, ns sdns14.ultradns.biz
    ▸ flipkart.com: type NS, class IN, ns sdns14.ultradns.org
    ▸ CK0POJMG874LJREF7EFN8430QVIT8BSM.com: type NSEC3, class IN
    ▸ CK0POJMG874LJREF7EFN8430QVIT8BSM.com: type RRSIG, class IN
    ▸ 9DA2I5Q698NJIM2MTFM0Q3GHAN5HKA22.com: type NSEC3, class IN
    ▸ 9DA2I5Q698NJIM2MTFM0Q3GHAN5HKA22.com: type RRSIG, class IN
  ▾ Additional records
    ▸ sdns14.ultradns.com: type A, class IN, addr 156.154.140.14
    ▸ sdns14.ultradns.com: type AAAA, class IN, addr 2610:a1:1001::e
    ▸ <Root>: type OPT
    [Request In: 3]
```

FOLLOWING SCREENSHOTS FOR www.flipkart.com query

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	10.0.2.15	DNS	78	Standard query 0x7574 A www.flipkart.com
2	0.000091630	10.0.2.15	10.0.2.15	DNS	78	Standard query 0xf477 AAAA www.flipkart.com
3	0.001450601	10.0.2.15	192.54.112.30	DNS	99	Standard query 0x3562 A _.flipkart.com OPT
4	0.240475295	192.54.112.30	10.0.2.15	DNS	809	Standard query response 0x3562 A _.flipkart.com NS s…
5	0.243384864	10.0.2.15	199.9.14.201	DNS	104	Standard query 0x1670 A sdns14.ultradns.biz OPT
6	0.243660944	10.0.2.15	199.9.14.201	DNS	104	Standard query 0xe3bf AAAA sdns14.ultradns.biz OPT
7	0.243949888	10.0.2.15	156.154.140.14	DNS	101	Standard query 0x2be1 A www.flipkart.com OPT
8	0.244227122	10.0.2.15	156.154.140.14	DNS	101	Standard query 0xf242 AAAA www.flipkart.com OPT
9	0.244484971	10.0.2.15	199.9.14.201	DNS	104	Standard query 0x69e2 A sdns14.ultradns.net OPT
10	0.244707097	10.0.2.15	199.9.14.201	DNS	104	Standard query 0xcd8f AAAA sdns14.ultradns.net OPT
11	0.244888234	10.0.2.15	199.9.14.201	DNS	104	Standard query 0xd0ef A sdns14.ultradns.org OPT

```
▸ Frame 1: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface any, id 0
▸ Linux cooked capture v1
▸ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.2.15
▾ User Datagram Protocol, Src Port: 49073, Dst Port: 53
    Source Port: 49073
    Destination Port: 53
    Length: 42
    Checksum: 0x1859 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
  ▸ [Timestamps]
    UDP payload (34 bytes)
▾ Domain Name System (query)
    Transaction ID: 0x7574
  ▸ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▸ Queries
```

```
▾ Domain Name System (query)
    Transaction ID: 0x7574
  ▾ Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▾ Queries
    ▸ www.flipkart.com: type A, class IN
    [Response In: 81]
```

```
▸ Frame 4: 809 bytes on wire (6472 bits), 809 bytes captured (6472 bits) on interface any, id 0
▸ Linux cooked capture v1
▸ Internet Protocol Version 4, Src: 192.54.112.30, Dst: 10.0.2.15
▾ User Datagram Protocol, Src Port: 53, Dst Port: 48995
    Source Port: 53
    Destination Port: 48995
    Length: 773
    Checksum: 0x0cab [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1]
  ▸ [Timestamps]
    UDP payload (765 bytes)
▾ Domain Name System (response)
    Transaction ID: 0x3562
  ▸ Flags: 0x8000 Standard query response, No error
    Questions: 1
    Answer RRs: 0
    Authority RRs: 8
    Additional RRs: 3
  ▸ Queries
  ▸ Authoritative nameservers
  ▸ Additional records
    [Request In: 3]
    [Time: 0.239024694 seconds]
```

```
▾ Domain Name System (response)
    Transaction ID: 0x3562
  ▸ Flags: 0x8000 Standard query response, No error
    Questions: 1
    Answer RRs: 0
    Authority RRs: 8
    Additional RRs: 3
  ▾ Queries
    ▸ _.flipkart.com: type A, class IN
  ▾ Authoritative nameservers
    ▸ flipkart.com: type NS, class IN, ns sdns14.ultradns.com
    ▸ flipkart.com: type NS, class IN, ns sdns14.ultradns.net
    ▸ flipkart.com: type NS, class IN, ns sdns14.ultradns.biz
    ▸ flipkart.com: type NS, class IN, ns sdns14.ultradns.org
    ▸ CK0POJMG874LJREF7EFN8430QVIT8BSM.com: type NSEC3, class IN
    ▸ CK0POJMG874LJREF7EFN8430QVIT8BSM.com: type RRSIG, class IN
    ▸ 9DA2I5Q698NJIM2MTFM0Q3GHAN5HKA22.com: type NSEC3, class IN
    ▸ 9DA2I5Q698NJIM2MTFM0Q3GHAN5HKA22.com: type RRSIG, class IN
  ▾ Additional records
    ▸ sdns14.ultradns.com: type A, class IN, addr 156.154.140.14
    ▸ sdns14.ultradns.com: type AAAA, class IN, addr 2610:a1:1001::e
    ▸ <Root>: type OPT
    [Request In: 3]
```