## Week #4
## Implementation of a Local DNS Server

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:

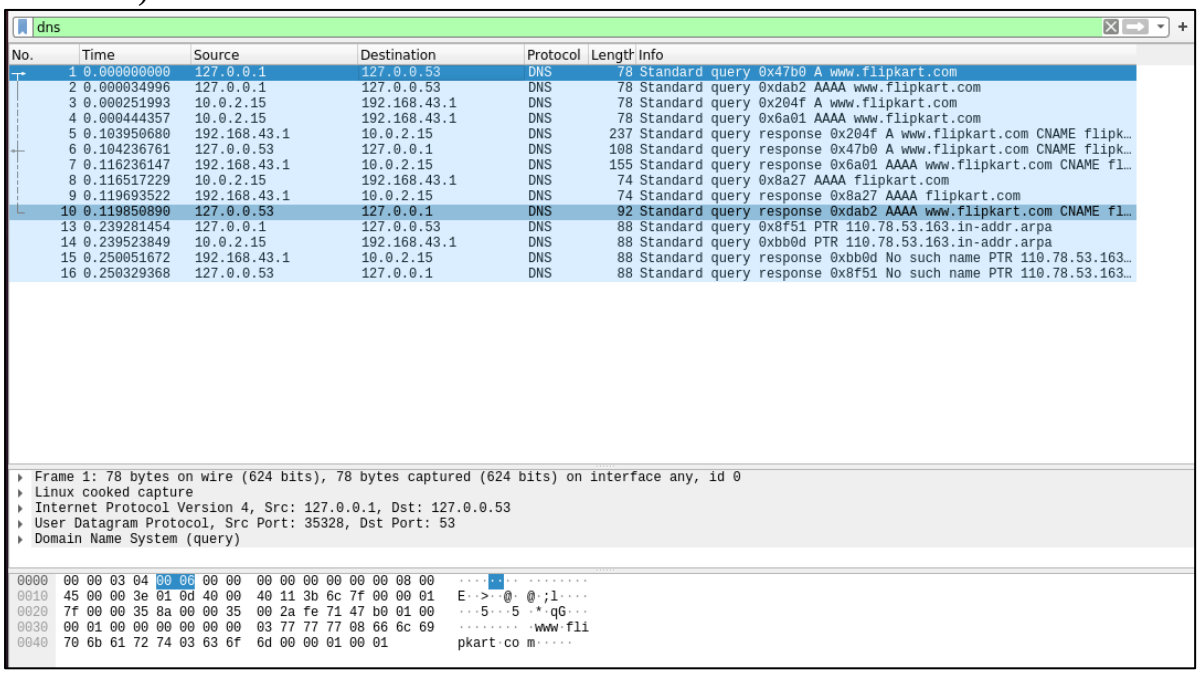
- DNS and how it works
- Install and set up a DNS server
- Functionality and operations

### Lab Setup

DNS Server: 10.0.2.15
User/Client: 10.0.2.4

**First Test:**

Ping a computer such as www.flipkart.com. 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 Machine

On the client machine 10.0.2.4, we need to use 10.0.2.15 as the local DNS server. This is achieved by changing the resolver configuration file **(/etc/resolv.conf)** of the user machine, so the server 10.0.2.15 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.

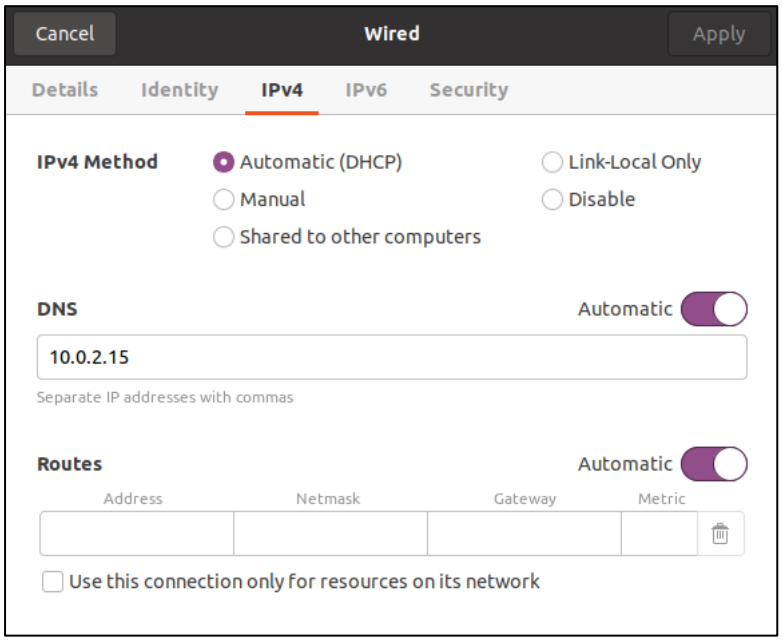
**nameserver 10.0.2.15**

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.

```
itsatul@Ruby:~$ sudo nano /etc/resolvconf/resolv.conf.d/head
[sudo] password for itsatul:
itsatul@Ruby:~$ sudo resolvconf -u
itsatul@Ruby:~$ 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
```

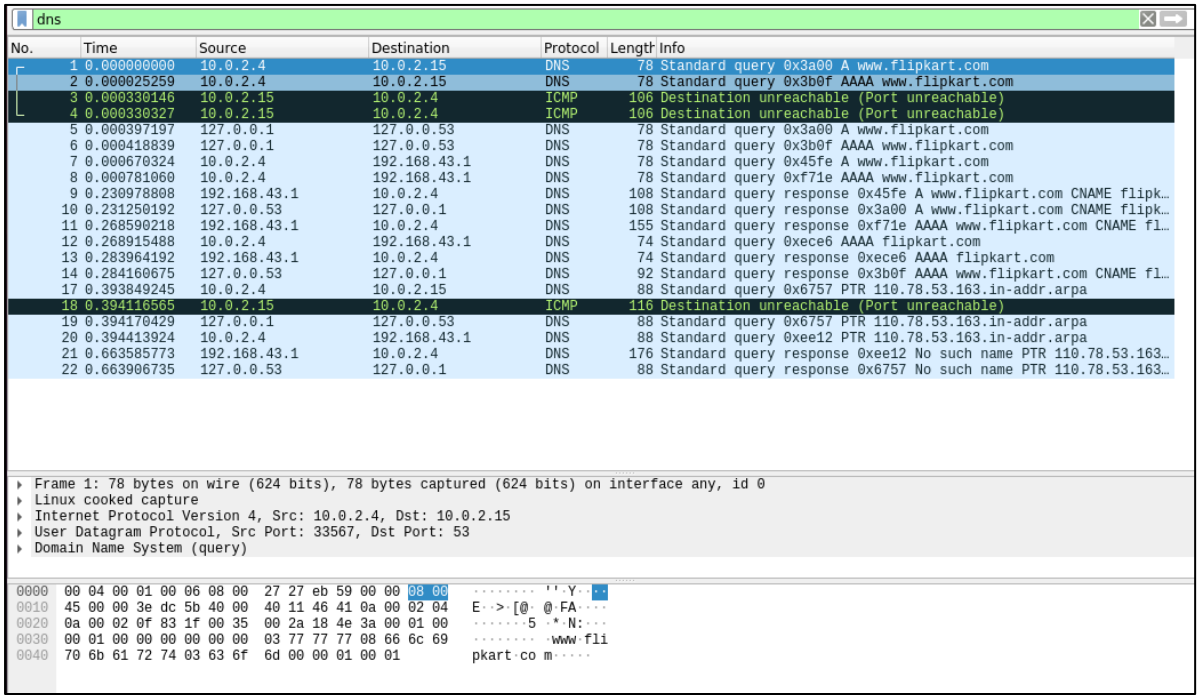
Also, add 10.0.2.15 in 'Additional DNS servers' field in IPv4 settings of client machine.



**Second Test:**

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

```
itsatul@Ruby:~$ ping www.flipkart.com
PING flipkart.com (163.53.78.110) 56(84) bytes of data.
64 bytes from 163.53.78.110 (163.53.78.110): icmp_seq=1 ttl=53 time=109 ms
64 bytes from 163.53.78.110 (163.53.78.110): icmp_seq=3 ttl=53 time=101 ms
64 bytes from 163.53.78.110 (163.53.78.110): icmp_seq=4 ttl=53 time=99.1 ms
^C
--- flipkart.com ping statistics ---
4 packets transmitted, 3 received, 25% packet loss, time 3009ms
rtt min/avg/max/mdev = 99.052/103.078/109.234/4.420 ms
itsatul@Ruby:~$
```



**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.

```
 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";
```

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**

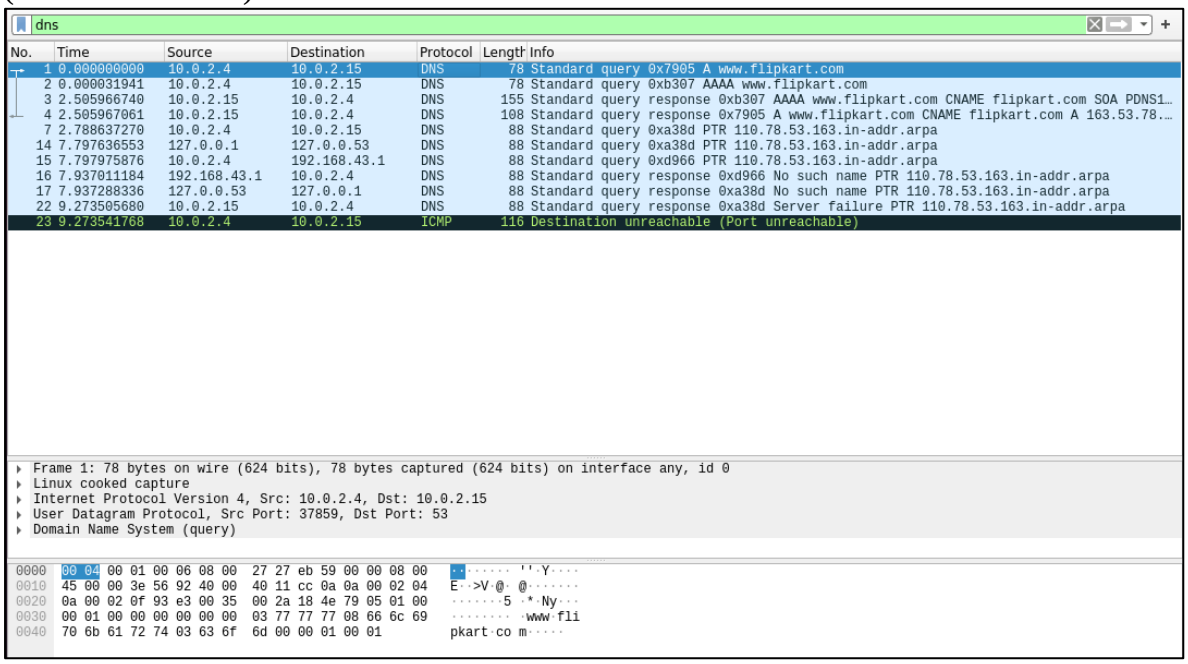
The two commands shown below are related to DNS cache. The first command dumps the content of the cache to the file specified above, and the second command clears the cache.

```
itsatul@Pearl:~$ sudo service bind9 restart
itsatul@Pearl:~$ sudo rndc dumpdb -cache
itsatul@Pearl:~$ sudo rndc flush
itsatul@Pearl:~$
```

**Step 3: Use the DNS server**

**Third Test:**

Now, go back to your user machine (10.0.2.4), and ping a computer such as www.flipkart.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).




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

**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 **pastel.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).

```
itsatul@Pearl:~$ sudo nano /etc/bind/named.conf
[sudo] password for itsatul:
itsatul@Pearl:~$ 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 "pastel.com" {
type master;
file "/etc/bind/pastel.com.db";
};

zone "10.0.2.in-addr.arpa" {
type master;
file "/etc/bind/10.0.2.db";
};
itsatul@Pearl:~$
```

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

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

Open

pastel.com.db
/etc/bind

```
1  $TTL 3D
2  @       IN      SOA ns.pastel.com. admin.pastel.com. (
3                  2008111001
4                  8H
5                  2H
6                  4W
7                  1D)
8
9  @       IN      NS      ns.pastel.com.
10 @       IN      MX      10 mail.pastel.com.
11
12 www     IN      A       10.0.2.101
13 mail    IN      A       10.0.2.102
14 ns      IN      A       10.0.2.10
15 *.pastel.com.   IN      A 10.0.2.100
```

The symbol '@' is a special notation representing the origin specified in **named.conf** (the string after "**zone**"). Therefore, '@' here stands for **pastel.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**

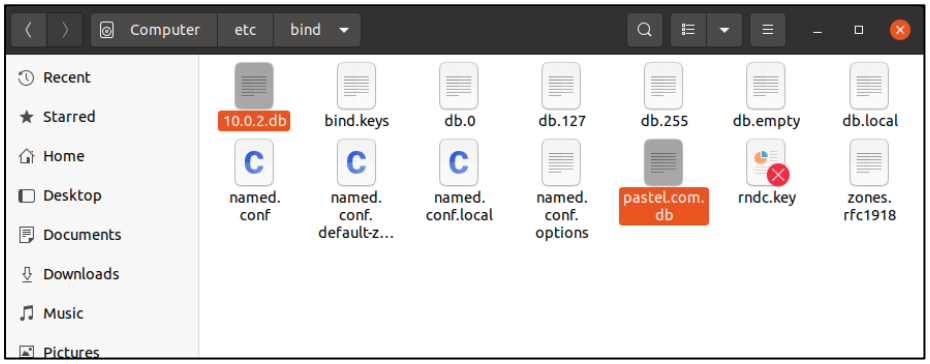
We create a reverse DNS lookup file called **10.0.2.db** for the example.net domain to support DNS reverse lookup, i.e., from IP address to hostname in the **/etc/bind/** directory with the following contents.

Open

10.0.2.db
/etc/bind

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

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

**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**

```
itsatul@Pearl:~$ sudo service bind9 restart
itsatul@Pearl:~$
```

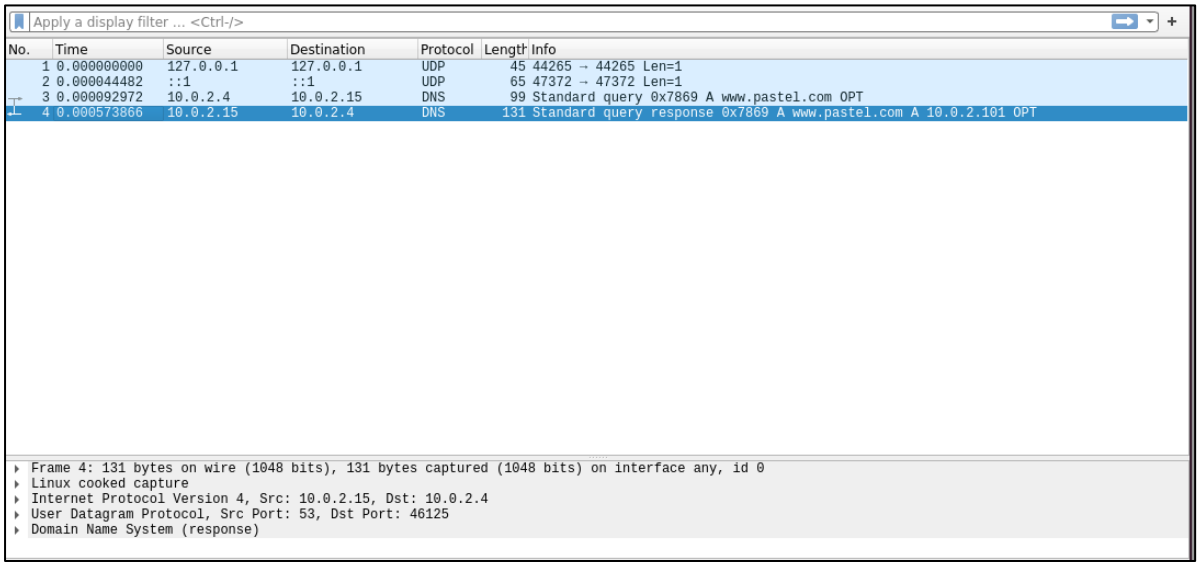
**Step 2:** Now, go back to the client machine and ask the local DNS server for the IP address of www.pastel.com using the dig command.

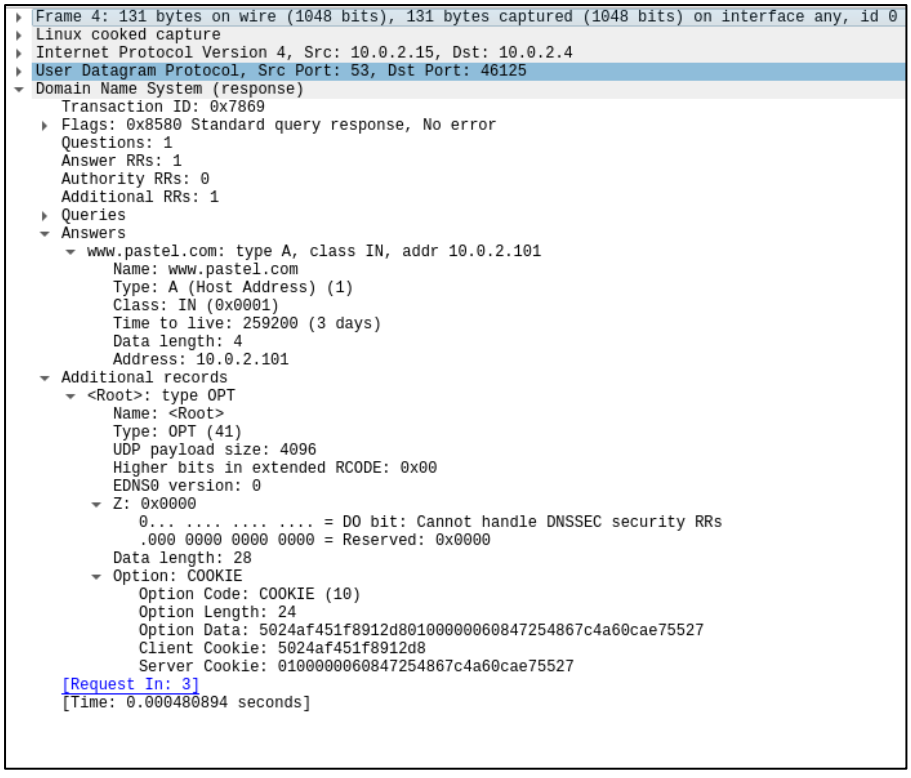
**Dig** stands for (Domain Information Groper) is a network administration command-line tool for querying DNS name servers. It is useful for verifying and troubleshooting DNS problems and also to perform DNS lookups and displays the answers that are returned from the name server that were queried. dig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

```
itsatul@Ruby: ~
itsatul@Ruby:~$ dig www.pastel.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.pastel.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14550
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 5e2c4d5cbb9a46c701000000608471bec73ca3b9960957e4 (good)
;; QUESTION SECTION:
;www.pastel.com.                        IN      A

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

;; Query time: 3 msec
;; SERVER: 10.0.2.15#53(10.0.2.15)
;; WHEN: Sun Apr 25 01:00:06 IST 2021
;; MSG SIZE  rcvd: 87

itsatul@Ruby:~$
```

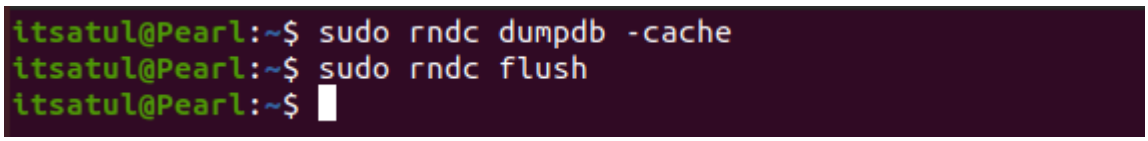
We can see that the ANSWER SECTION contains the DNS mapping. We can see that the IP address of www.pastel.com is now 10.0.2.101, which is what we have setup in the DNS server.

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




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



**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?

- The DNS query and response messages are visible in the screenshots.They are sent over UDP.

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 source port of DNS response message 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 message is sent to the IP Address 10.0.2.15 which is the same for local DNS Server.

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.