

COMPUTER NETWORKS

LABORATORY- Week-4

Student Details-

Name	Siddharth Magadam
SRN	PES1UG19CS482
Section	H

Task 1: Configuring Client Machine

```
student@CSELAB: /etc/resolvconf/resolv.conf.d
student@CSELAB:~$ cd /etc/resolvconf
student@CSELAB:/etc/resolvconf$ cd resolv.conf
bash: cd: resolv.conf: No such file or directory
student@CSELAB:/etc/resolvconf$ cd resolv.conf.d
student@CSELAB:/etc/resolvconf/resolv.conf.d$ sudo gedit head
** (gedit:3202): WARNING **: Set document metadata failed: Setting attribute met
adata::gedit-position not supported
student@CSELAB:/etc/resolvconf/resolv.conf.d$ sudo cat 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 10.1.10.83
student@CSELAB:/etc/resolvconf/resolv.conf.d$ sudo resolvconf -u
student@CSELAB:/etc/resolvconf/resolv.conf.d$
```

Editing Ethernet connection 1

Connection name: Ethernet connection 1

General | Ethernet | 802.1x Security | DCB | IPv4 Settings | IPv6 Settings

Method: Automatic (DHCP)

Addresses

Address	Netmask	Gateway

Additional DNS servers: 10.1.10.83

Additional search domains:

DHCP client ID:

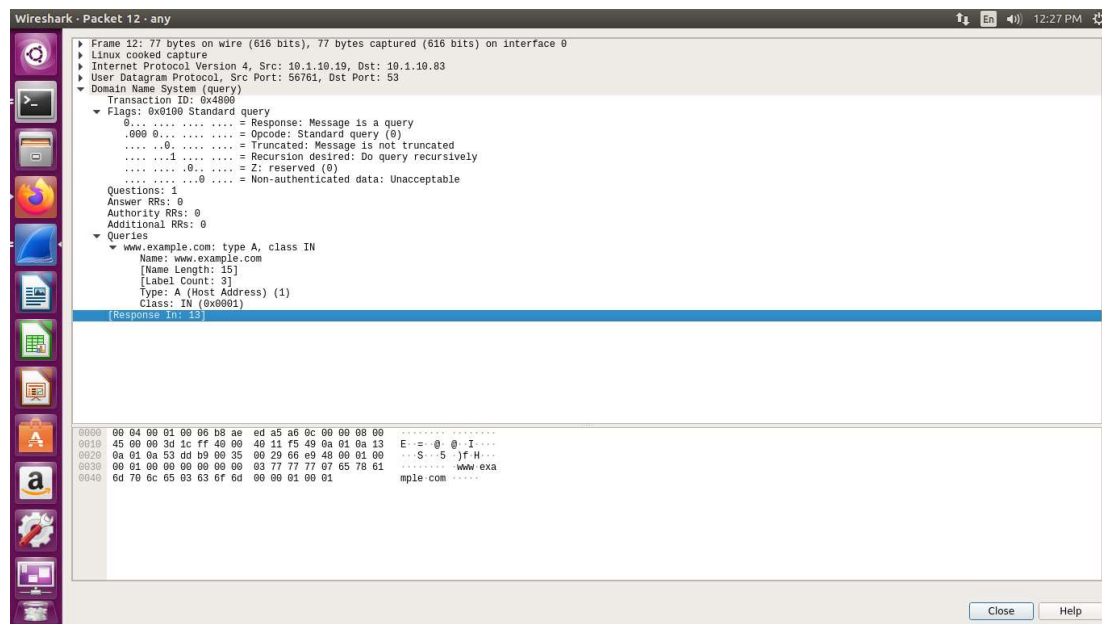
☐ Require IPv4 addressing for this connection to complete

Routes...

Cancel Save

DNS query captured by wireshark for pingng www.example.com

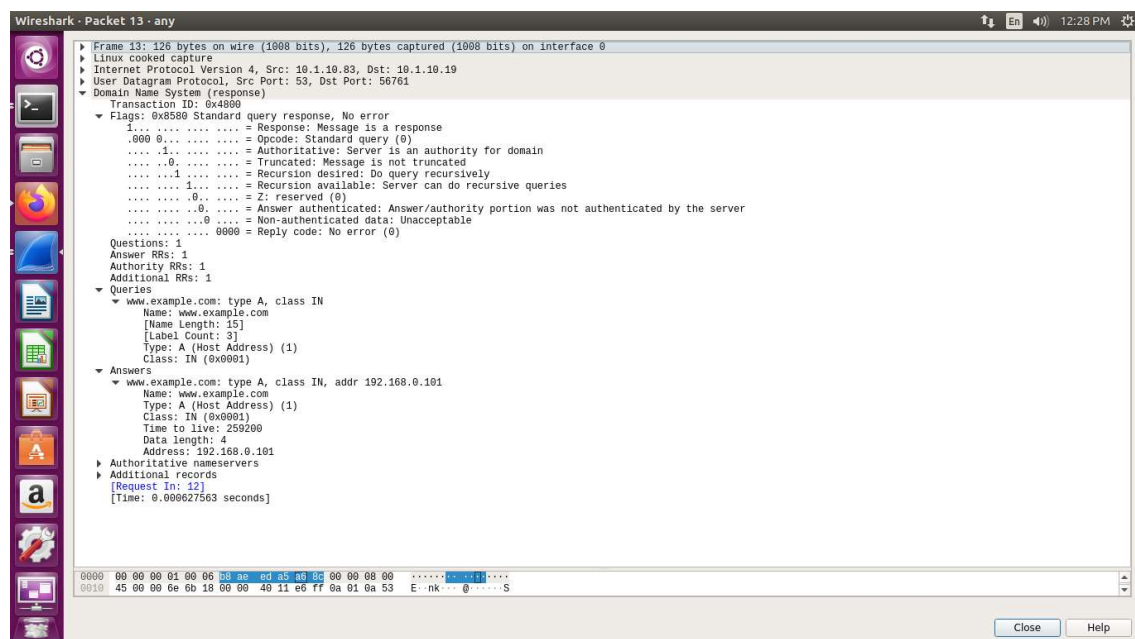
Observation 2:



We can see under the flags section that message is a query. In Queries, the name of query is `www.example.com`.

The corresponding DNS response

We get a message response, in which we can observe the Name, Class, TTL and Data length.

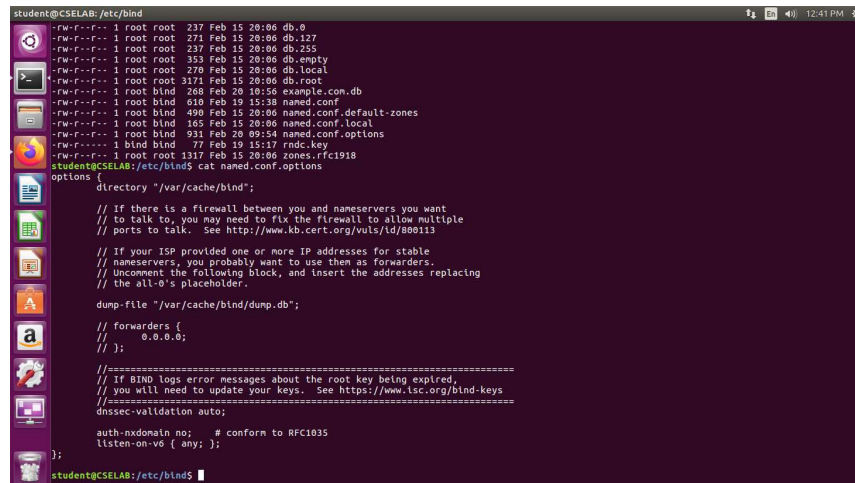


Task 2: Setting up local DNS server

Installing bind9

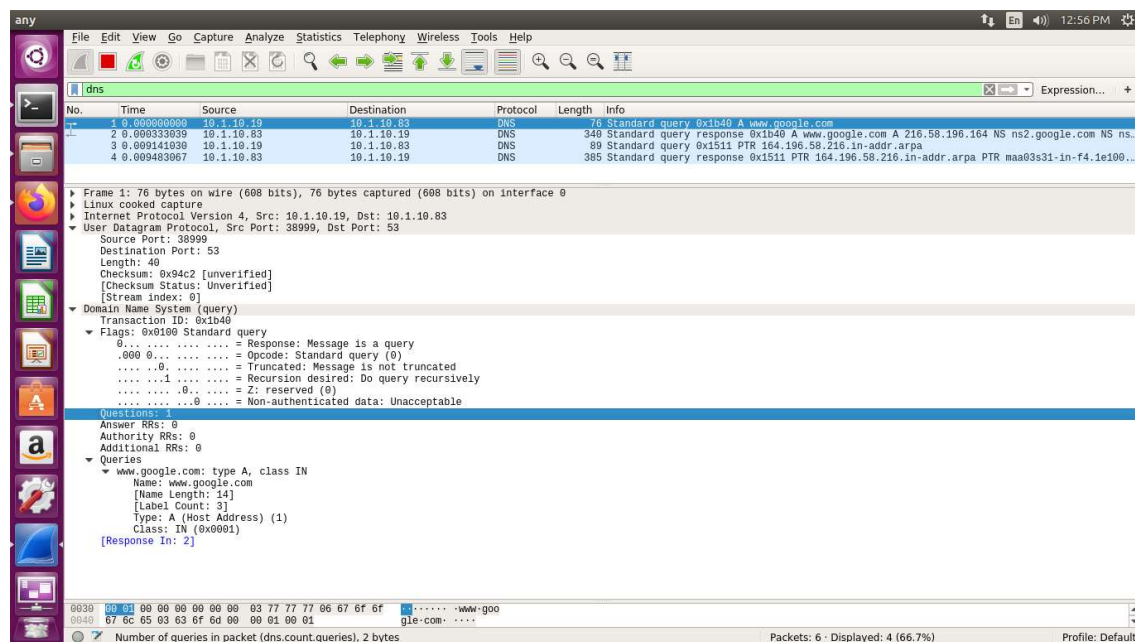
```
$ sudo apt-get update
```

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



```
student@CSELAB: /etc/bind$  
-rw-r--r-- 1 root root 237 Feb 15 20:06 db.0  
-rw-r--r-- 1 root root 271 Feb 15 20:06 db.127  
-rw-r--r-- 1 root root 237 Feb 15 20:06 db.255  
-rw-r--r-- 1 root root 353 Feb 15 20:06 db.empty  
-rw-r--r-- 1 root root 270 Feb 15 20:06 db.local  
-rw-r--r-- 1 root root 371 Feb 15 20:06 db.root  
-rw-r--r-- 1 root blind 268 Feb 20 10:56 example.com.db  
-rw-r--r-- 1 root blind 610 Feb 19 15:38 named.conf  
-rw-r--r-- 1 root blind 490 Feb 15 20:06 named.conf.default-zones  
-rw-r--r-- 1 root blind 165 Feb 15 20:06 named.conf.local  
-rw-r--r-- 1 root blind 931 Feb 20 09:54 named.conf.options  
-rw-r--r-- 1 blind blind 77 Feb 19 15:17 rndc.key  
-rw-r--r-- 1 root root 1317 Feb 15 20:06 zones.rfc1918  
student@CSELAB: /etc/bind$ cat 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;  
  
    auth-nxdomain no;    # conform to RFC1035  
    listen-on-v6 { any; };  
};  
student@CSELAB: /etc/bind$
```

Observation 3: We ping www.google.com and observe the DNS query.



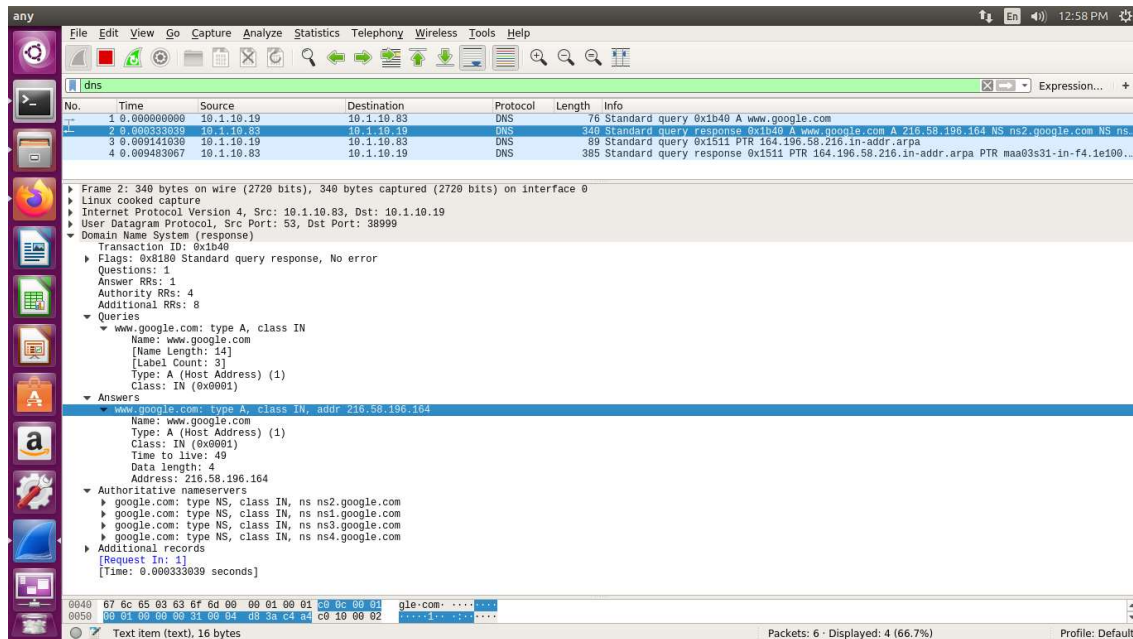
The Wireshark packet capture shows a DNS query and response. The packet list on the left shows four packets related to the DNS query. The packet details pane on the right shows the structure of the DNS query and response.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.1.10.19	10.1.10.83	DNS	76	Standard query 0x1b40 A www.google.com
2	0.000333039	10.1.10.83	10.1.10.19	DNS	340	Standard query response 0x1b40 A www.google.com A 216.58.196.164 NS ns2.google.com NS ns...
3	0.000141030	10.1.10.19	10.1.10.83	DNS	89	Standard query 0x1511 PTR 164.196.58.216.in-addr.arpa
4	0.000483067	10.1.10.83	10.1.10.19	DNS	385	Standard query response 0x1511 PTR 164.196.58.216.in-addr.arpa PTR naa03s31-in-f4.1e109...

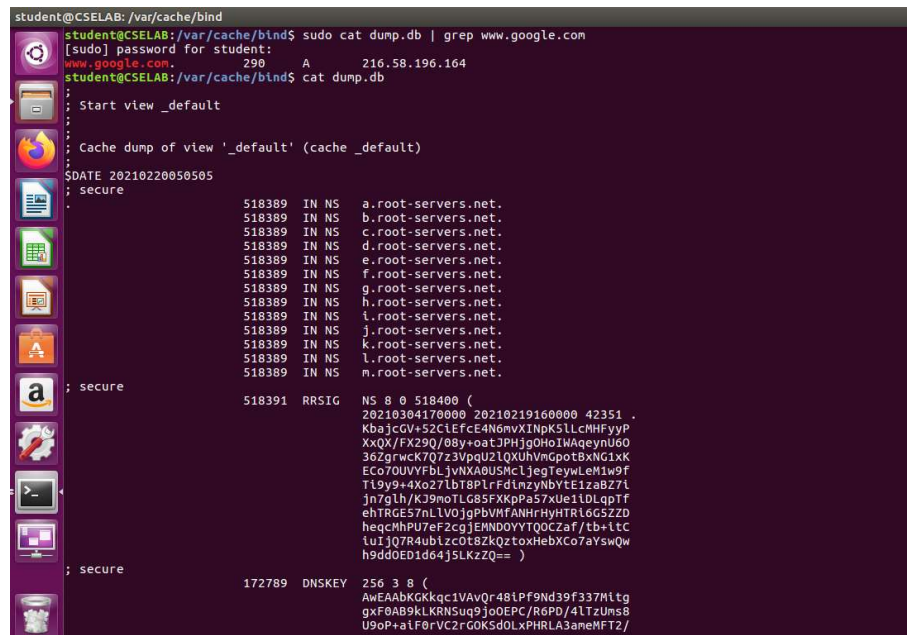
The packet details pane shows the structure of the DNS query and response. The query is a standard query for the domain www.google.com. The response is a standard query response containing the IP address 216.58.196.164 and the nameserver ns2.google.com.

We observe that the client machine send the request for the server machine who's IP address is the one with destination IP address, in this query.

The server now responds to that query, hence, the server computer acts as a local DNS server.



Observation 4: We now observe the cache dumped in the local DNS (server machine). We use grep command for filtering the query name i.e. www.google.com



Task 3: Host a Zone in Local DNS server .

Step 1: Creating Zones.

We had two zone entries in the DNS server by adding the following contents to **/etc/bind/named.conf**

```
student@CSELAB:~$ sudo gedit /etc/bind/named.conf
student@CSELAB:/etc/bind$ sudo cat named.conf
** (gedit:4477): WARNING **: Set document metadata failed: Setting attribute metadata
student@CSELAB:/etc/bind$ sudo cat 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 "10.1.10.in-addr.arpa" {
    type master;
    file "/etc/bind/10.1.10.db";
};
student@CSELAB:/etc/bind$
```

The first zone is forward lookup zone. It resolves the host name to IP address.

The second zone is forward lookup zone. It resolves the IP address to hostname.

Step 2 &3: Setup the forward lookup zone.

We copy the two files given by the faculty members to **/etc/bind** location.

```
student@CSELAB:~$ sudo cp 10.1.10.db /etc/bind
student@CSELAB:~$ sudo cp example.com.db /etc/bind
student@CSELAB:~$
```

Task 4 : Restart the BIND server and test

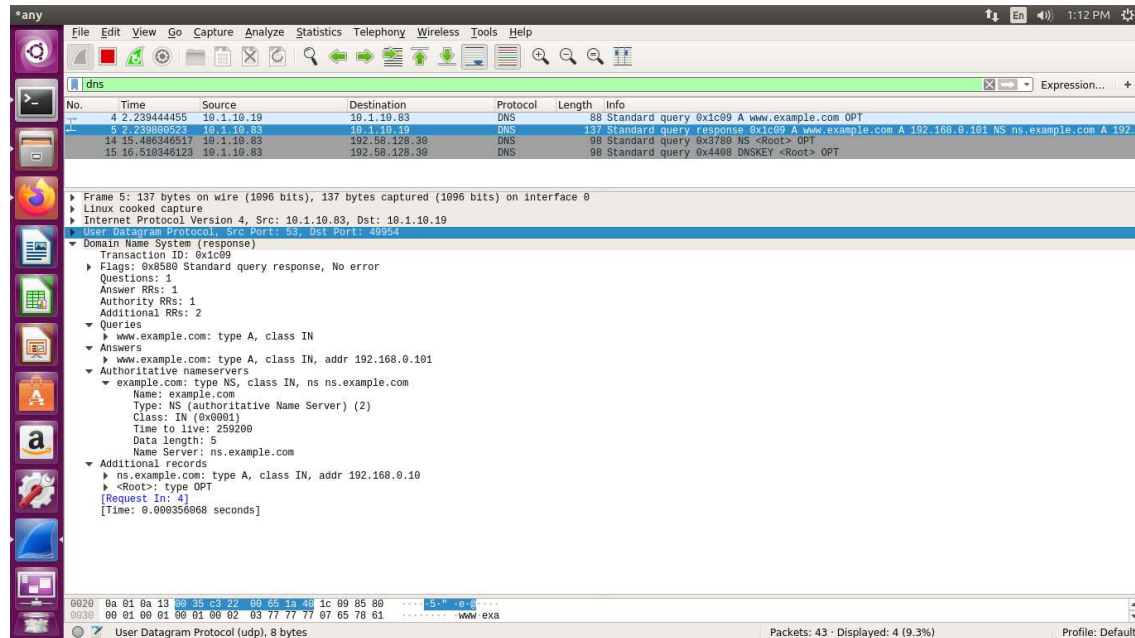
Step1: We restart the bind9 server using command **sudo service bind9 restart**

Step 2: We find the IP address of Local DNS server from the client computer using dig command

```
student@CSELAB:/etc/resolvconf/resolv.conf.d$ dig www.example.com
;; MSG SIZE rcvd: 93
student@CSELAB:/etc/resolvconf/resolv.conf.d$ dig www.example.com
;<<>> Dig 9.10.3-P4-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;;->HEADER:- opcode: QUERY, status: NOERROR, id: 7177
;; 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 192.168.0.101
;; AUTHORITY SECTION:
example.com. 259200 IN NS ns.example.com.
;; ADDITIONAL SECTION:
ns.example.com. 259200 IN A 192.168.0.10
;; Query time: 0 msec
;; SERVER: 10.1.10.83#53(10.1.10.83)
;; WHEN: Sat Feb 20 13:11:12 IST 2021
;; MSG SIZE rcvd: 93
student@CSELAB:/etc/resolvconf/resolv.conf.d$
```


We can see that the ANSWER SECTION contains the DNS mapping. We can see that the IP address of `www.example.com` is now `192.168.0.101`, which is what we have setup in the DNS server.

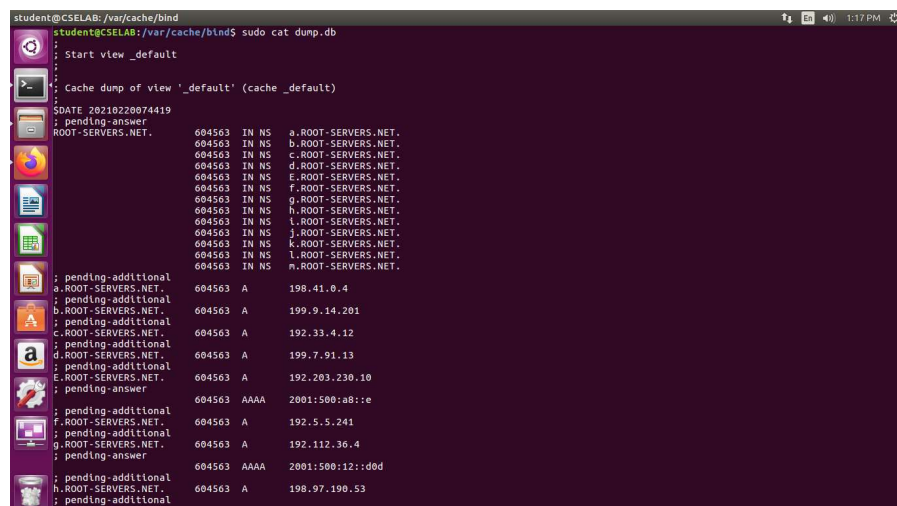
Step 3: Observe the wireshark capture



We load DNS cache, use the below command.

sudo rndc dump.db -cache

The Local DNS cache on server machine after dig command



We clear DNS cache by using command **sudo rndc flush**

-----X-----