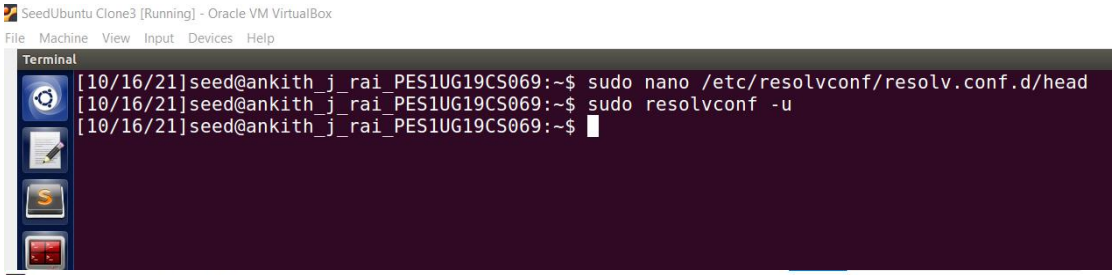
LOCAL DNS Attack Lab

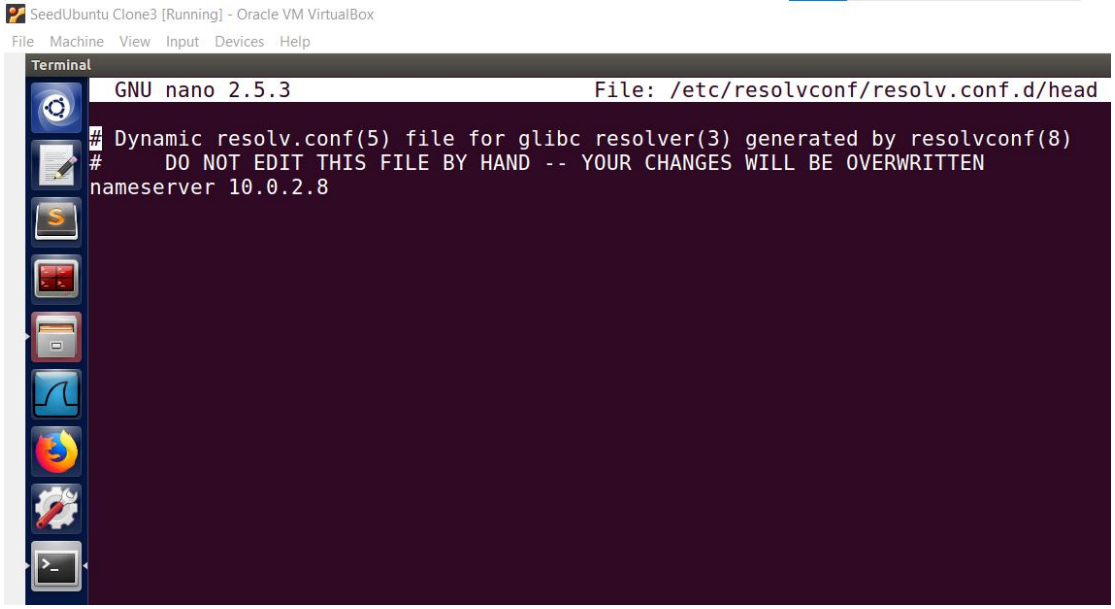
Name : Ankith J Rai
SRN : PES1UG19CS069
SEC : B

Machine	IP address
Attacker	10.0.2.5
Victim	10.0.2.9
DNS Server	10.0.2.8

Part I: Setting Up a Local DNS Server

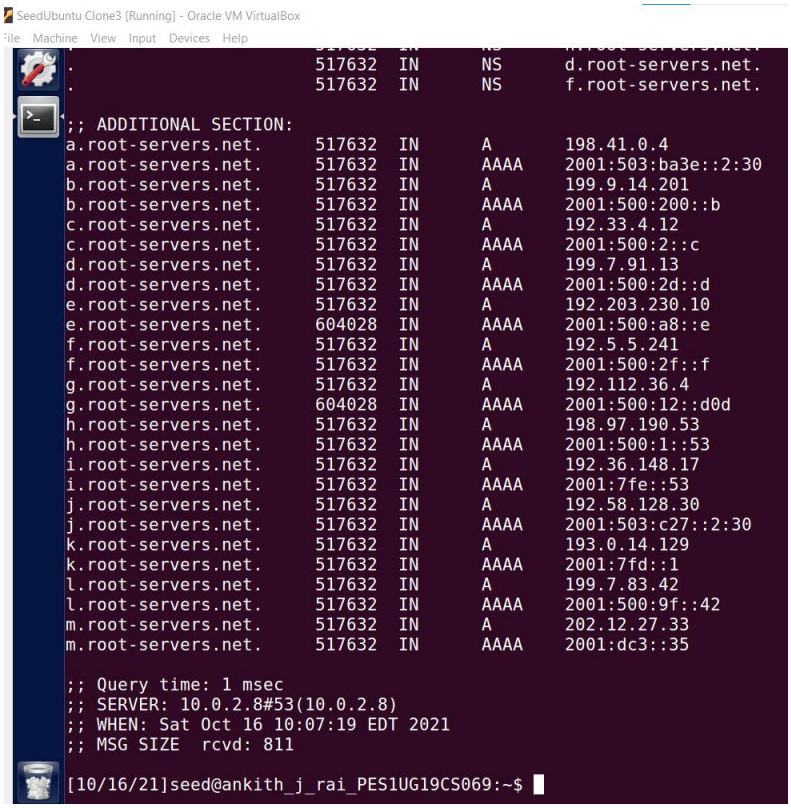
Task 1: Configure the User Machine





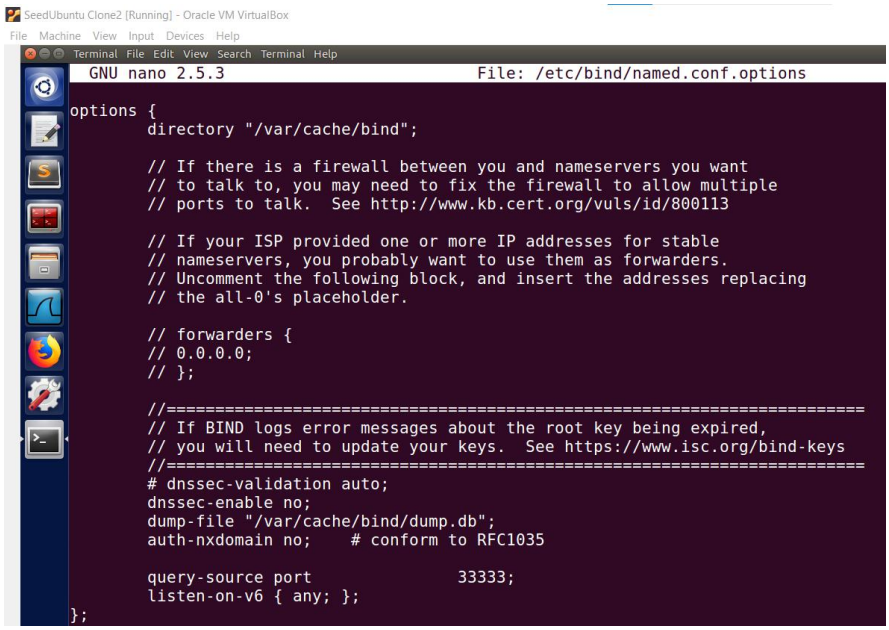
The User machine has now been configured.

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

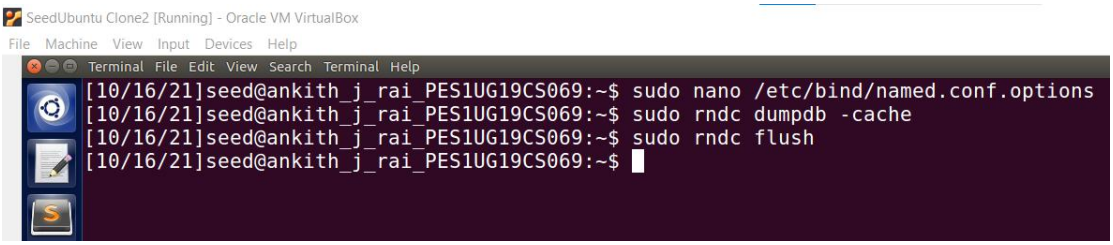


Task 2: Set Up a Local DNS Server

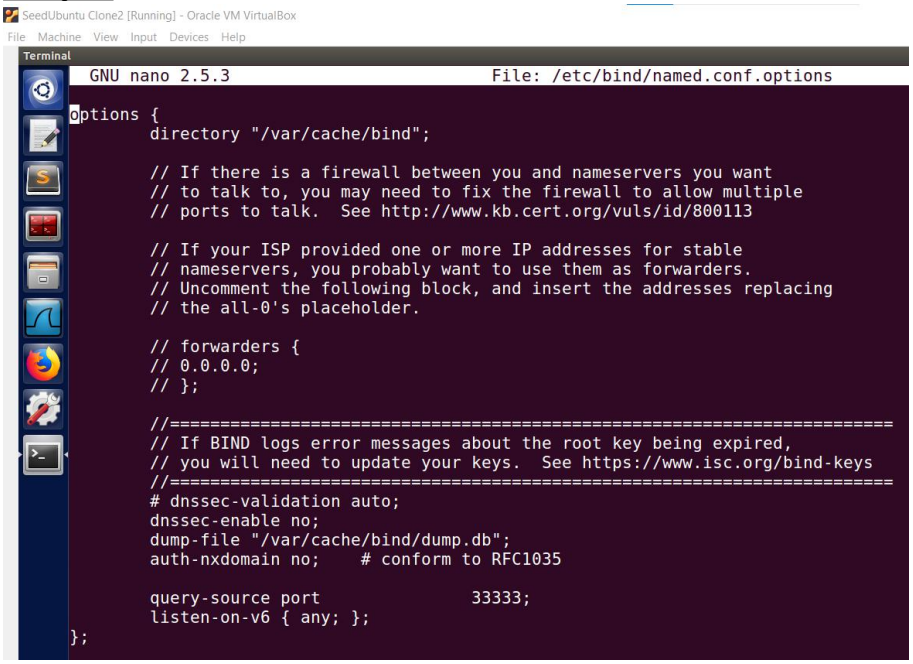
Step 1: Configure the BIND9 Server



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

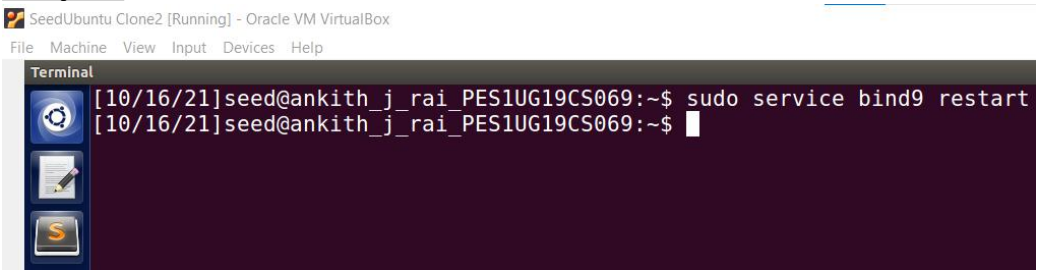


Step 2: Turn off DNSSEC



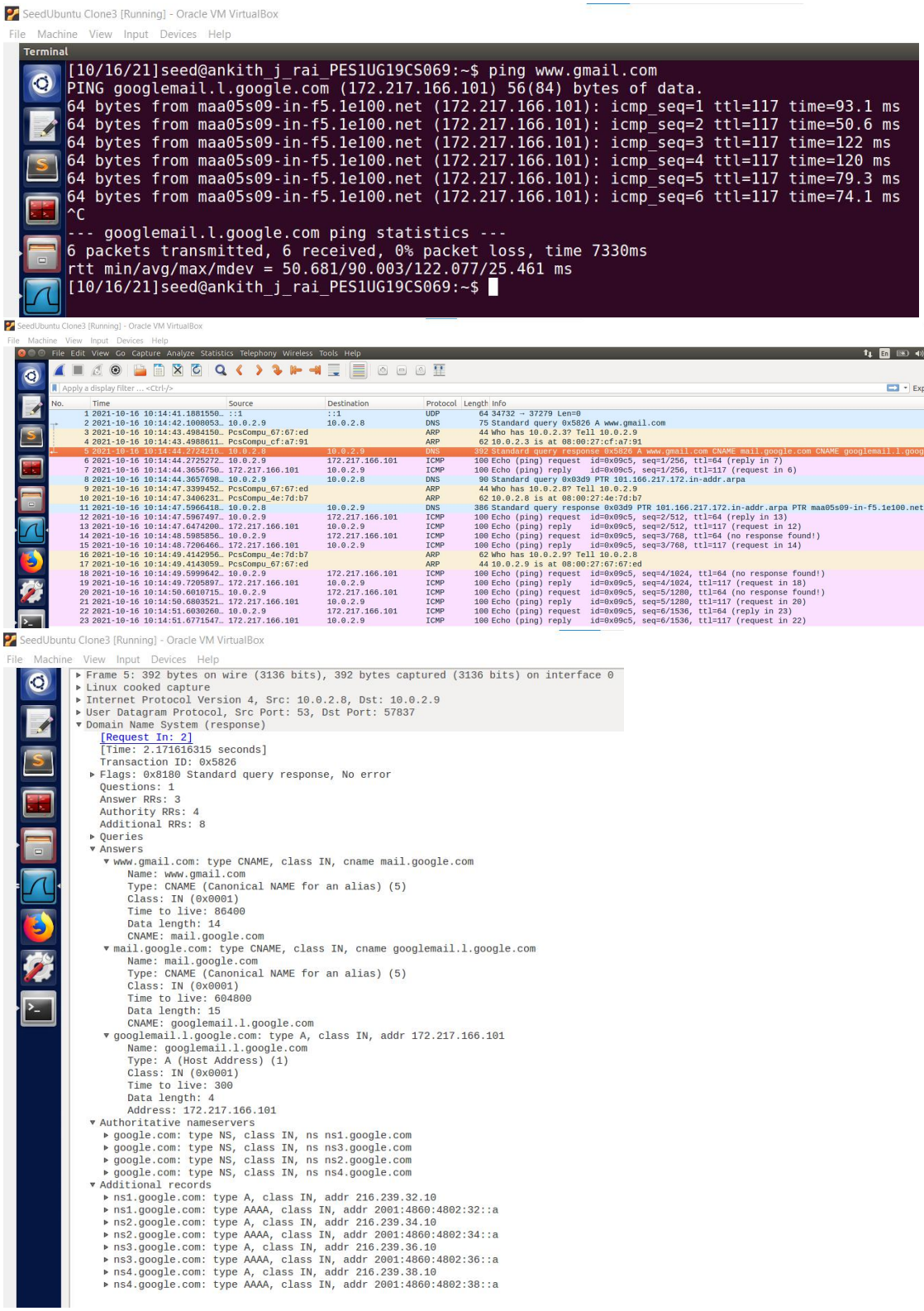
Now we switched off the DNSSEC from the options.

Step 3: Start DNS server



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.

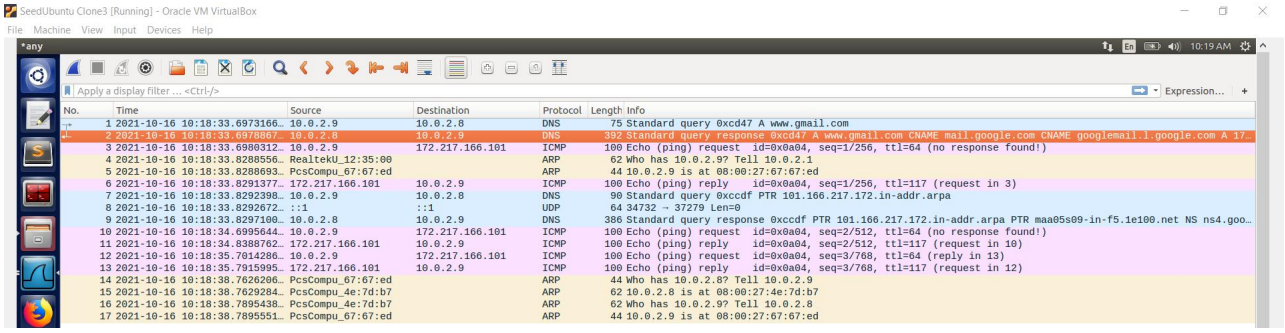
On Pinging once more:

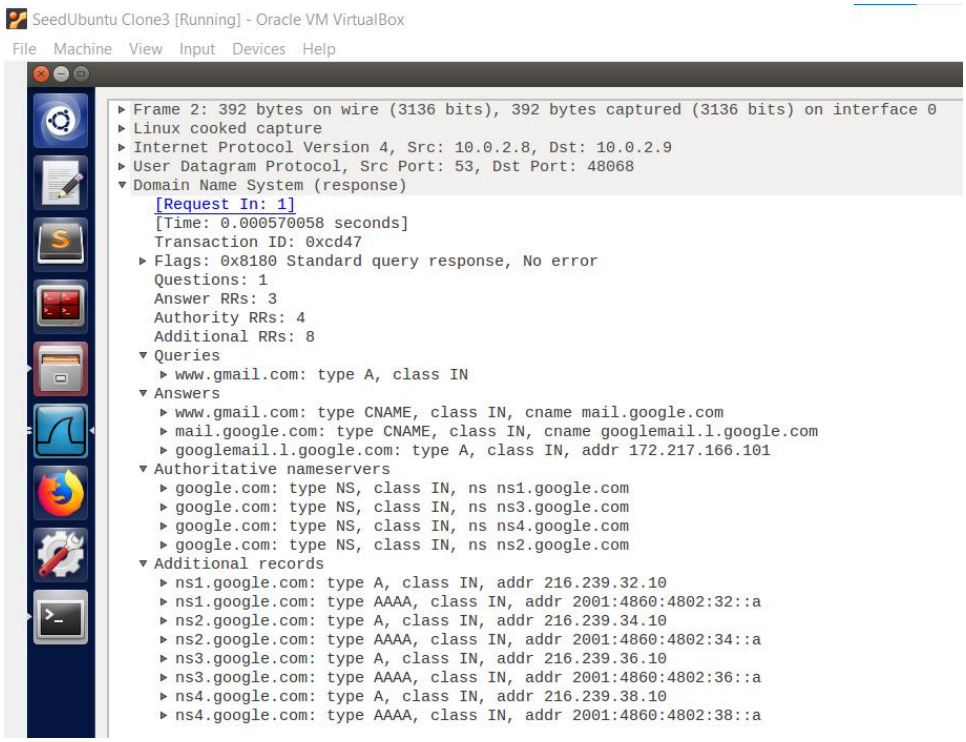
SeedUbuntu Clone3 [Running] - Oracle VM VirtualBox

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:~$
```

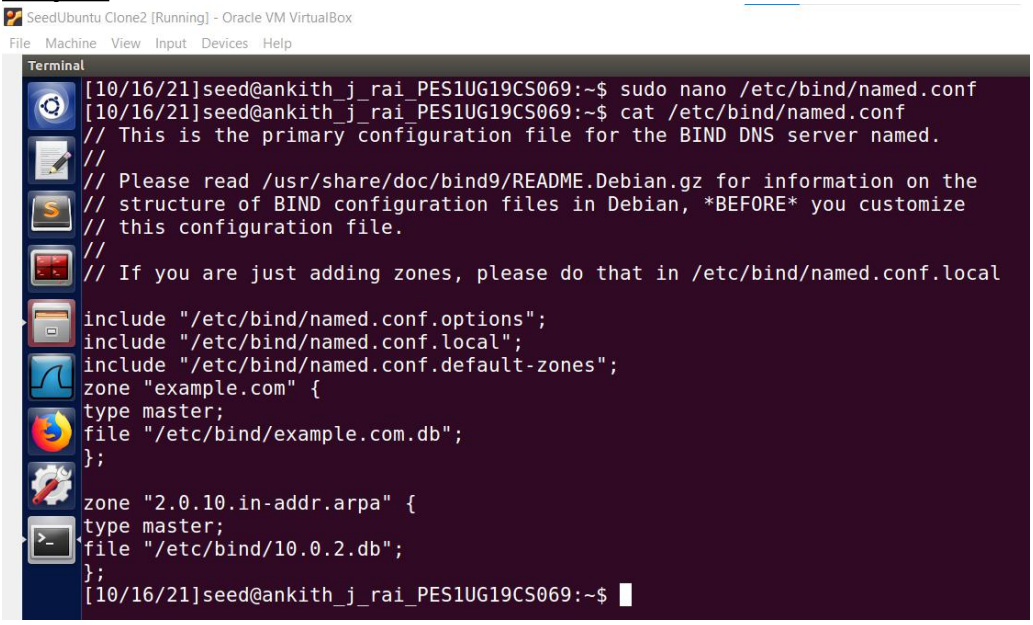




We can see that on pinging www.gmail.com once more the pinging started immediately as the DNS server had already cache the ip address of the www.gmail.com from the earlier ping.

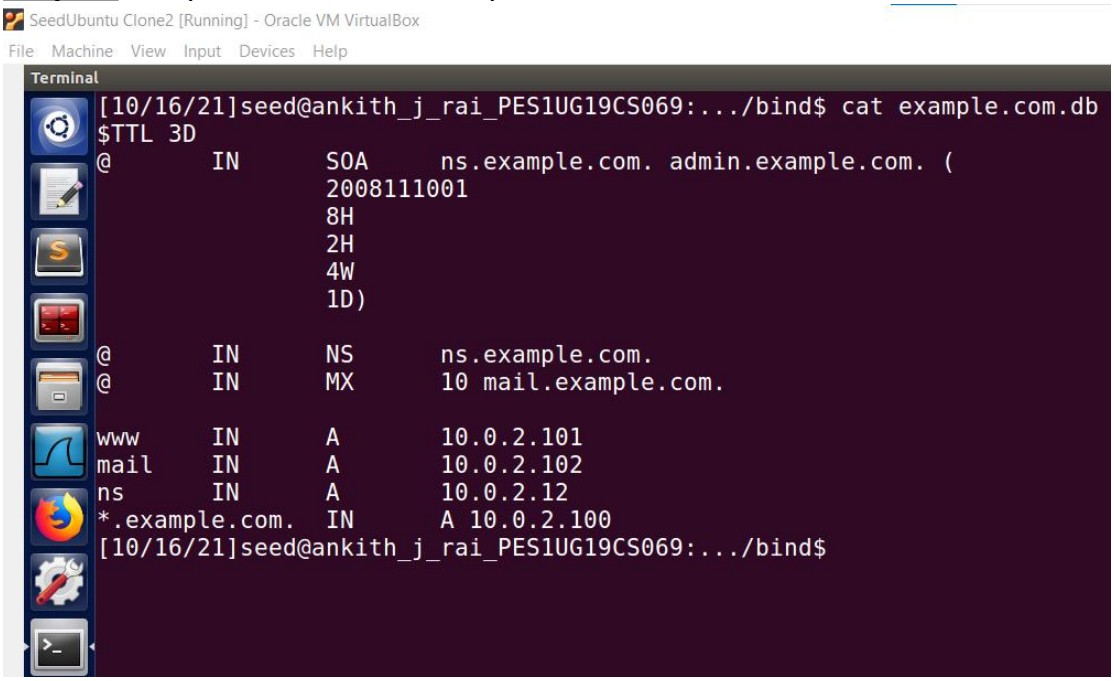
Task 3: Host a Zone in the Local DNS server

Step 1:



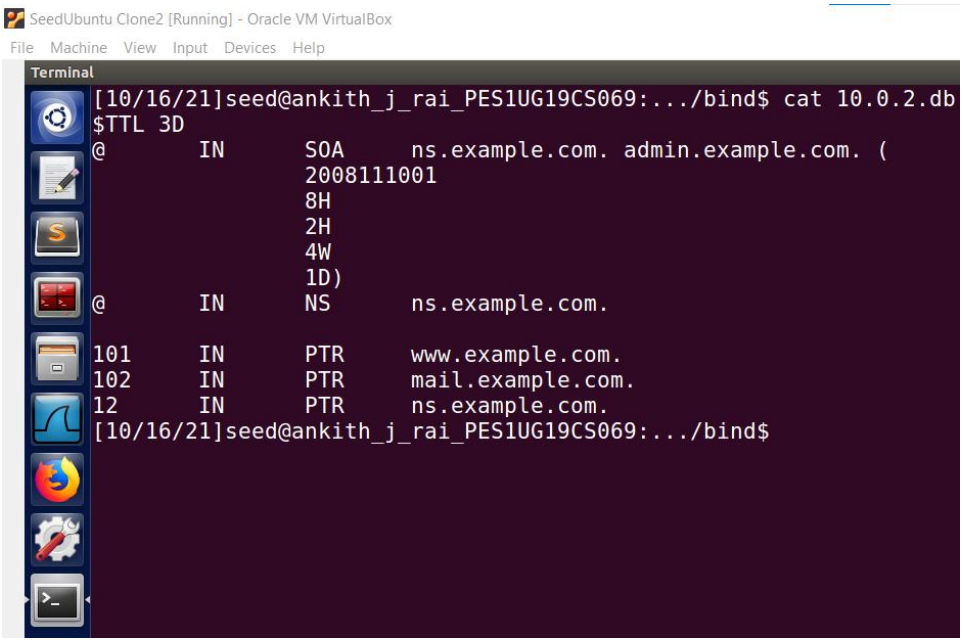
Now we have created two zones.

Step 2: Setup the forward lookup zone file



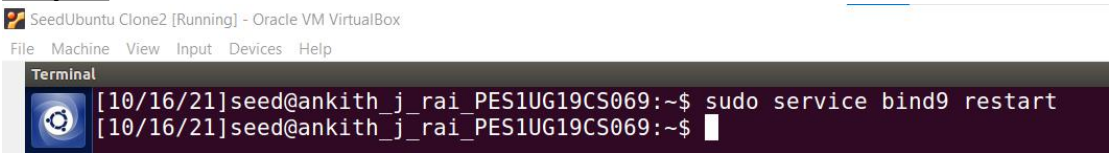
Now we have created example.com.db zone file.

Step 3: Setup the reverse lookup zone file

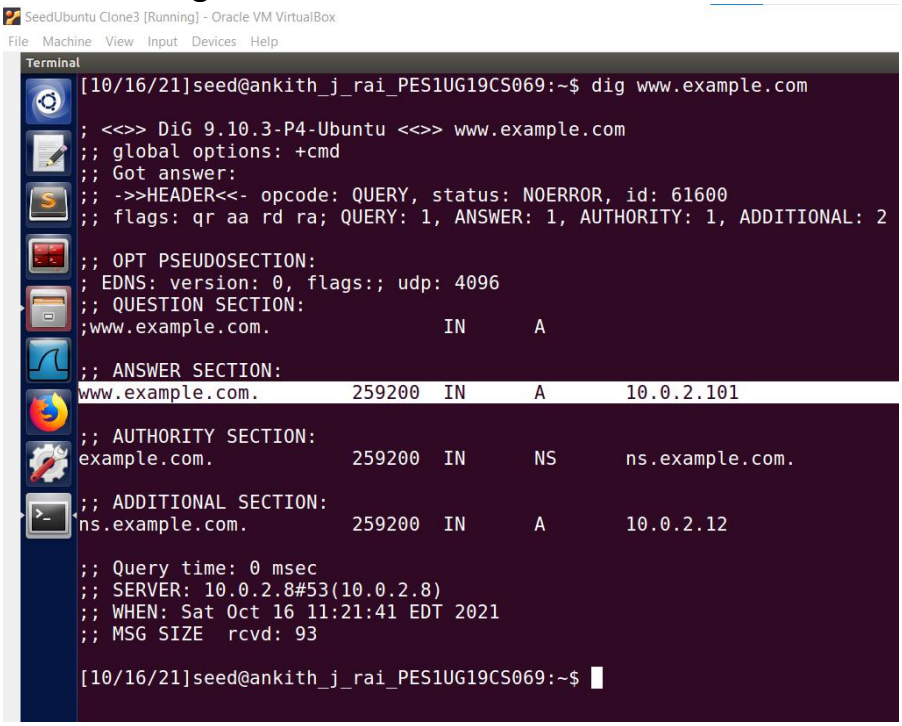


Now we have created 10.0.2.db zone file.

Step 4:



Now on using DIG command on victim machine

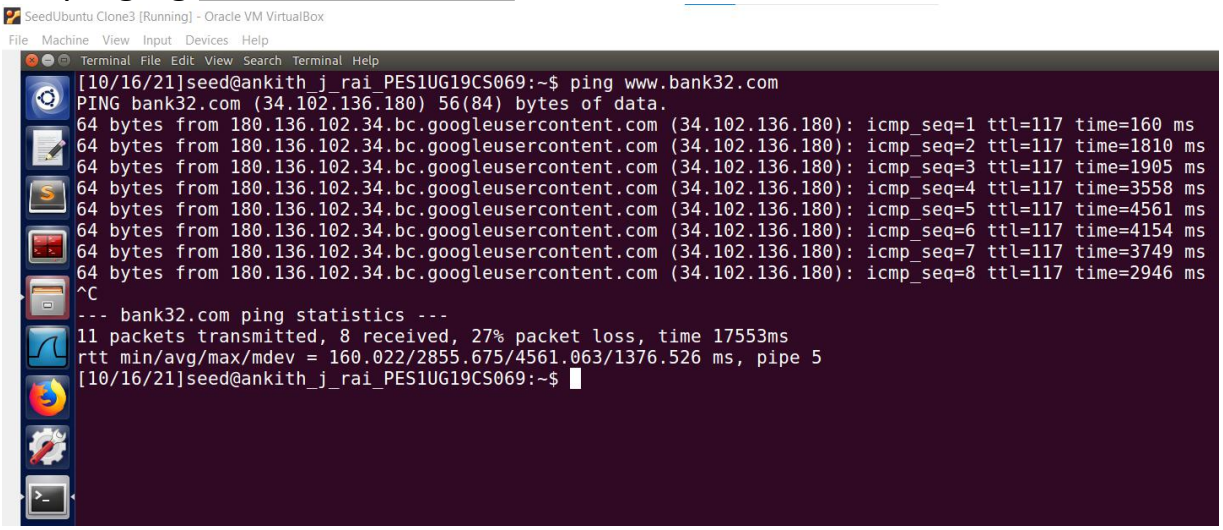


We can see that in the answer section the ip address associated with www.example.com. is 10.0.2.101

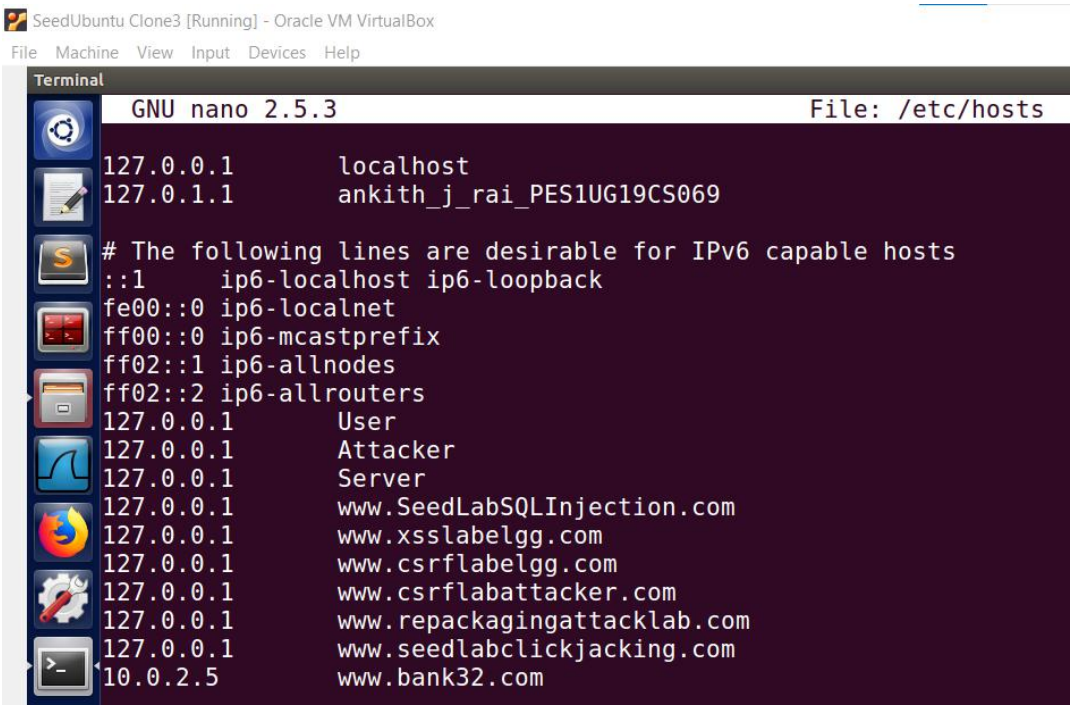
Part II: Attacks on DNS

Task 4: Modifying the Host File

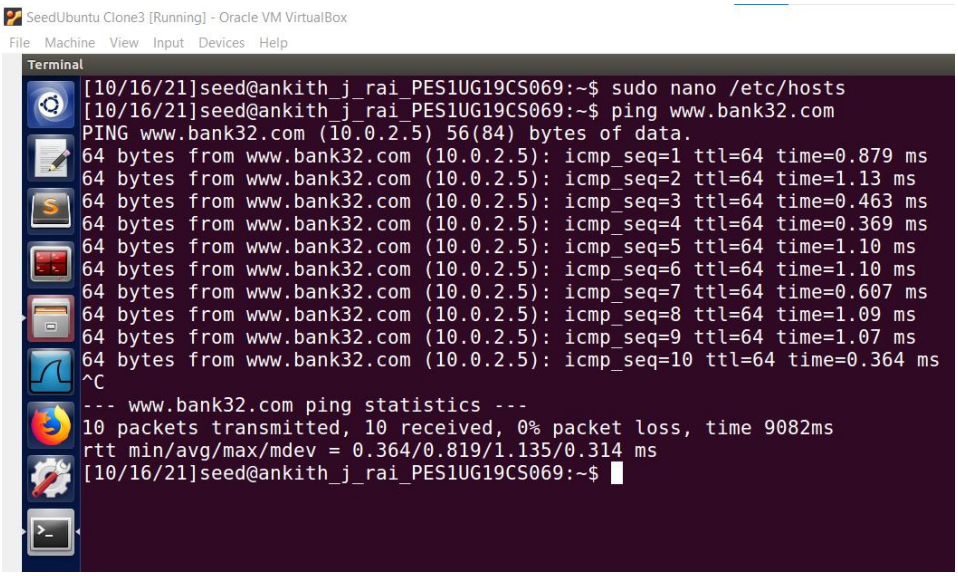
On pinging www.bank32.com on victim machine.



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



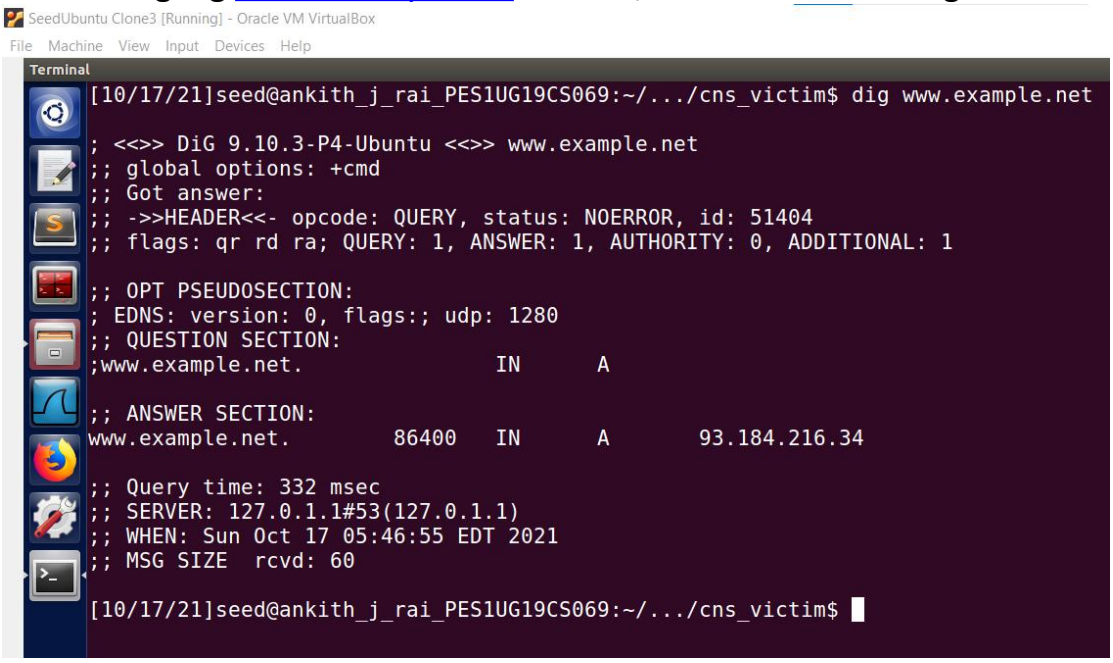
From the above screenshot we can see that www.bank32.com has been mapped to ip address 10.0.2.5 now.



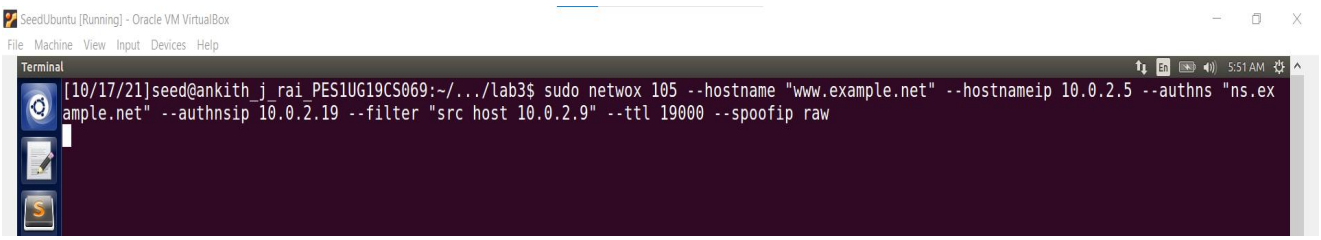
Now on pinging www.bank32.com we can see that we are getting response form 10.0.2.5 which is the ip address of attacker machine.

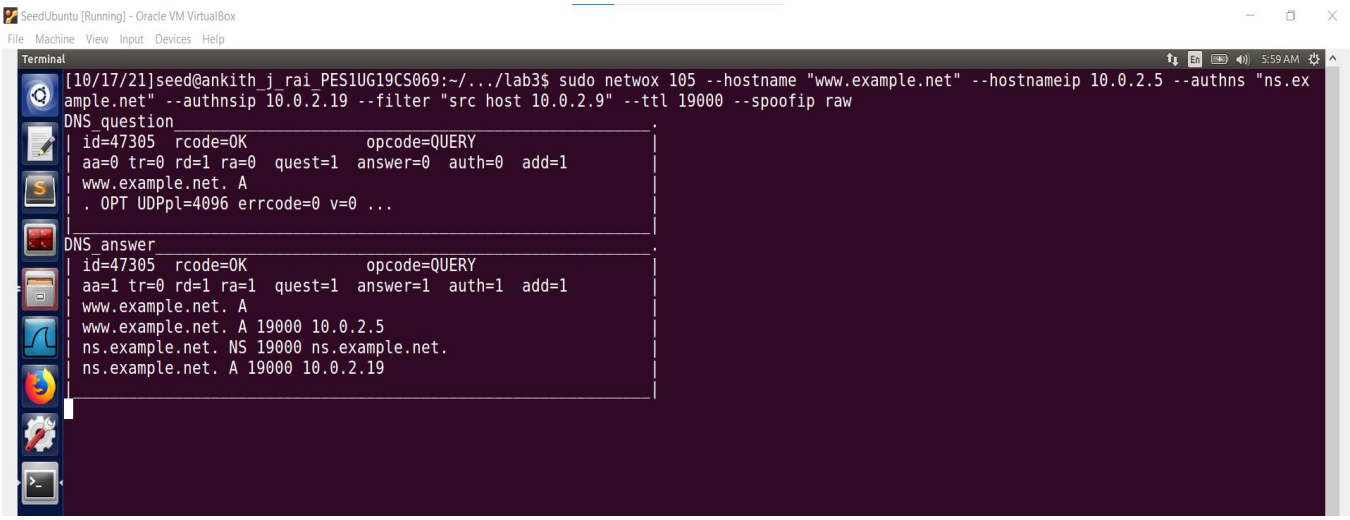
Task 5: Directly Spoofing Response to User

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

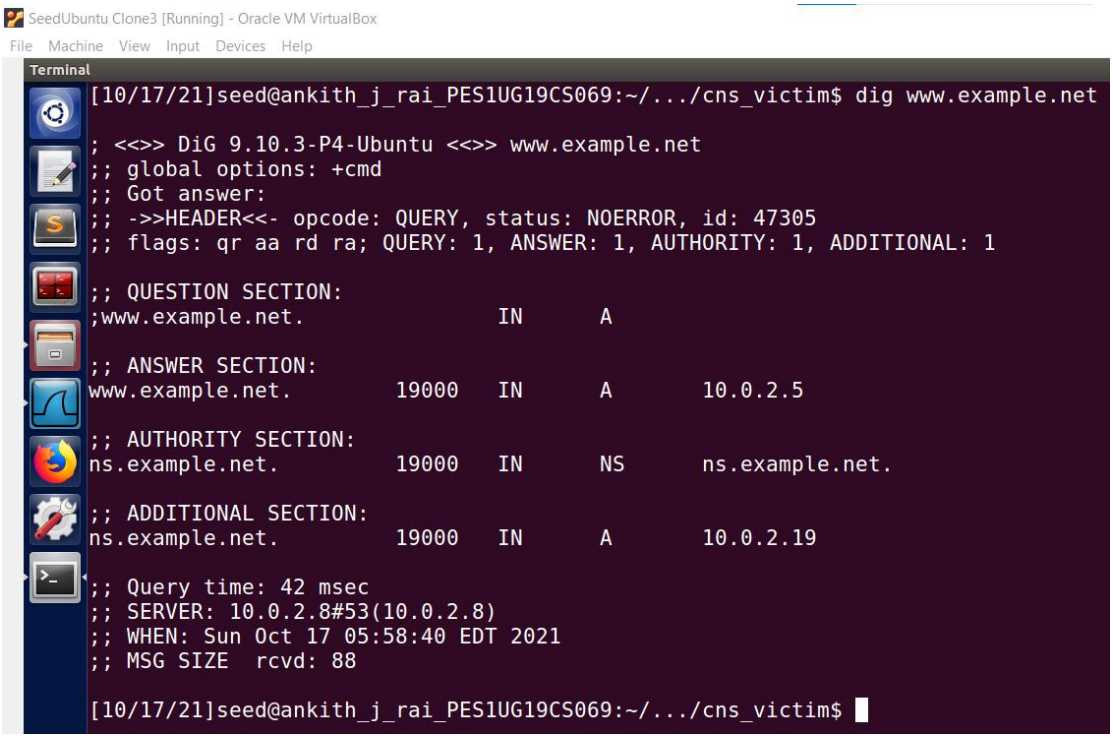


We see that the ip address of www.example.net. is 93.184.216.34

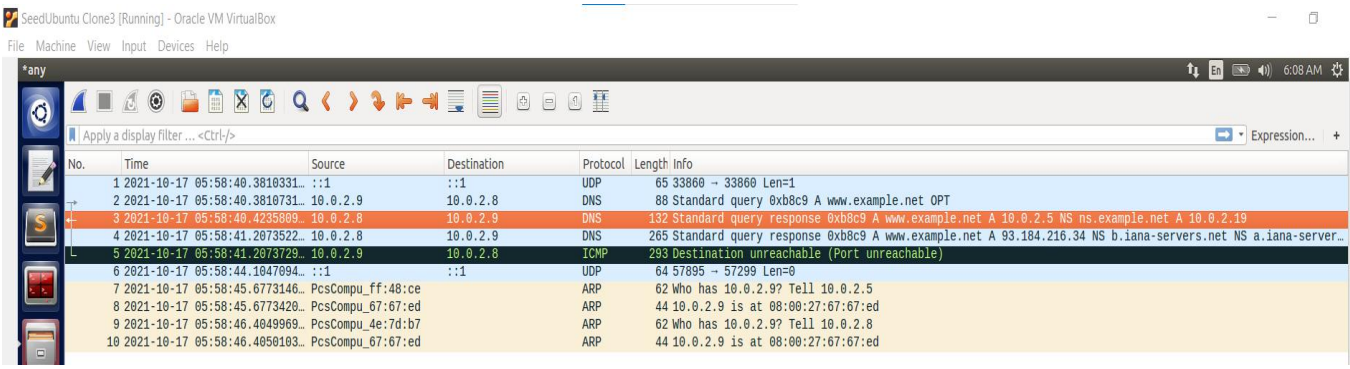




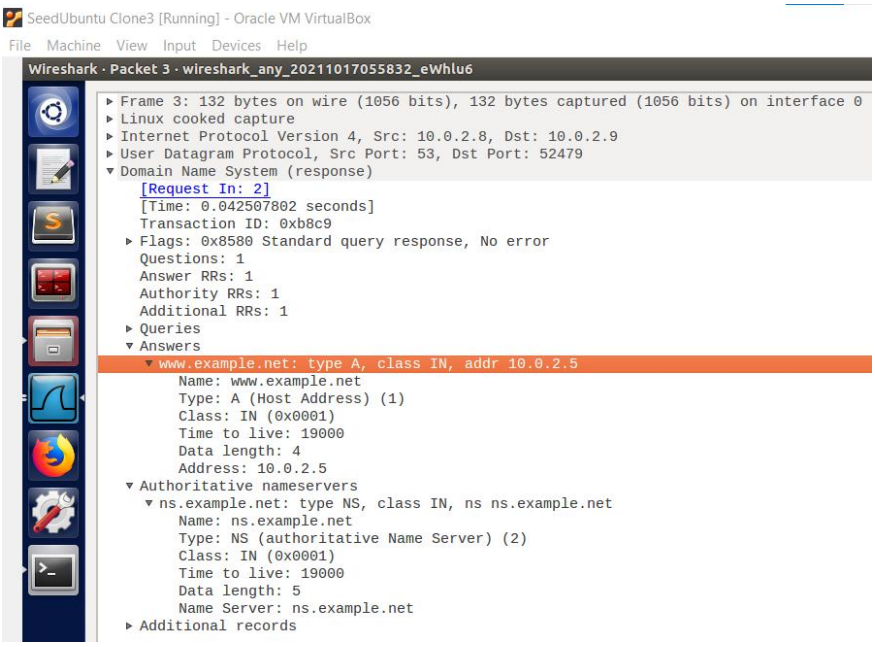
We can see that the attacker machine send's a DNS response as ip address as 10.0.2.5 and authoritative name sever as 10.0.2.19 for a DNS query made by victim for www.example.net .



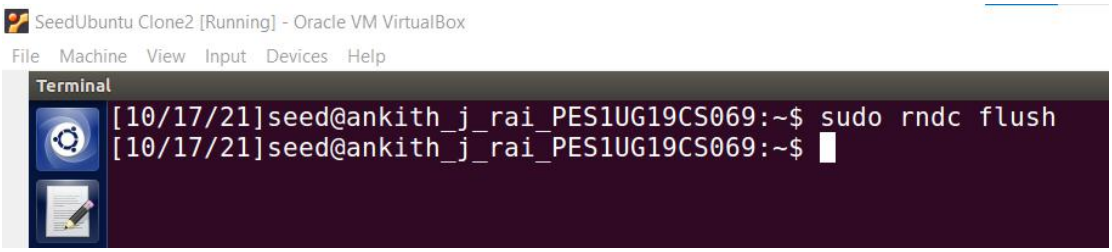
From the above screenshot of victim machine we can see that www.example.net has been mapped to 10.0.2.5 .Hence we can see the forged DNS response is successful.



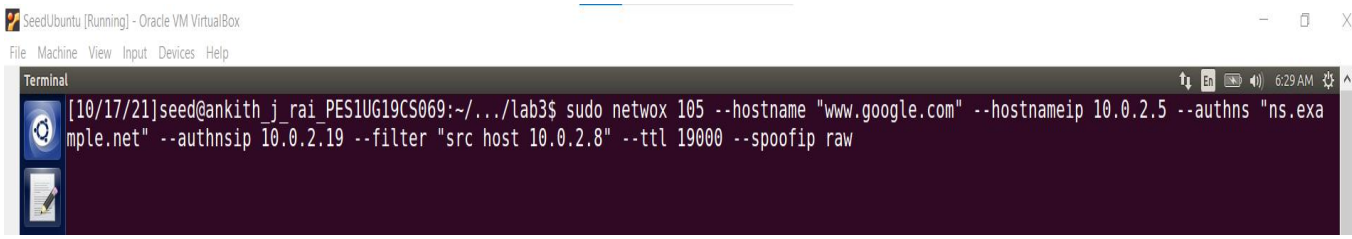
We can see from the above screenshot that before the actual DNS response reaches the victim the Forged DNS response has reached victim machine and hence the Victim machine is made to believe that the ip address for www.example.net is 10.0.2.5(attacker machine ip address) .



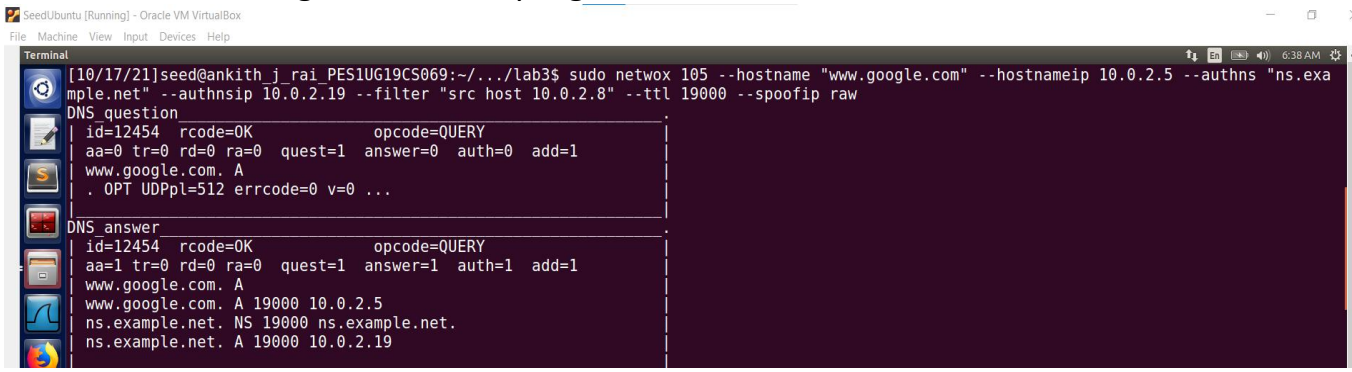
Now we are clean the cache of DNS sever.



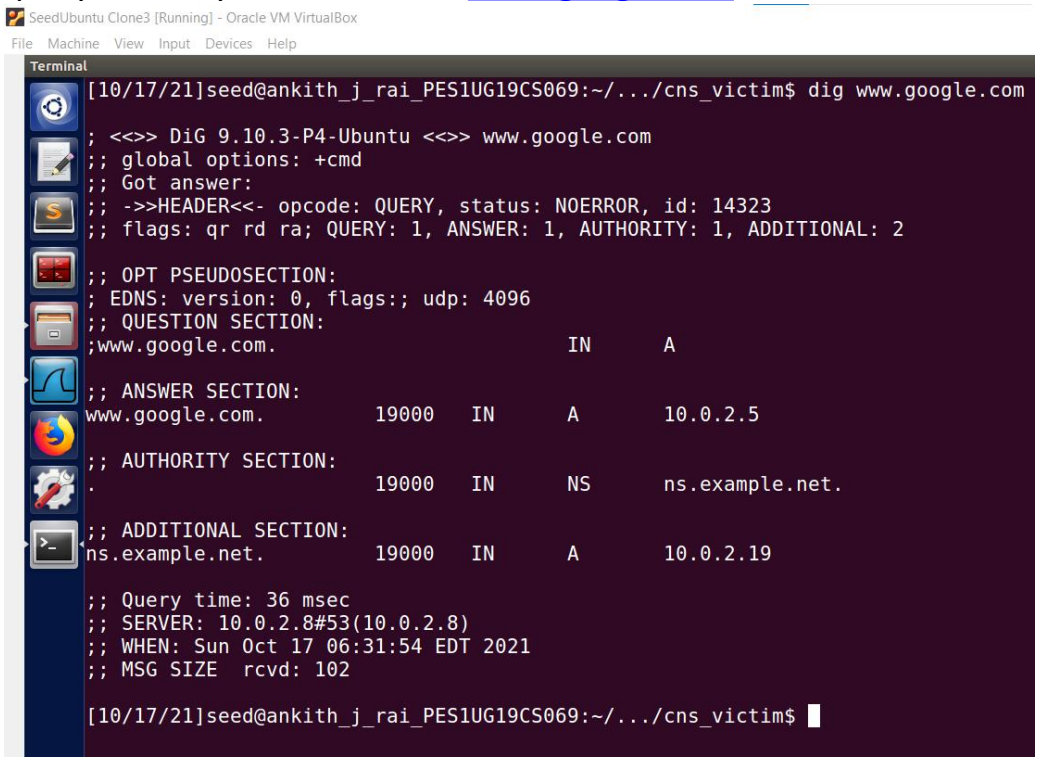
Task 6: DNS Cache Poisoning Attack



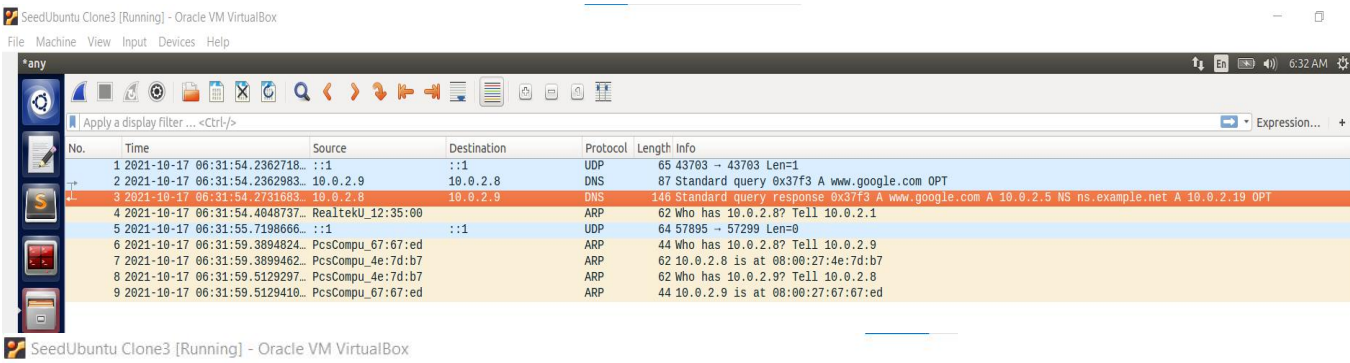
We are now running the attacker program on attacker machine.

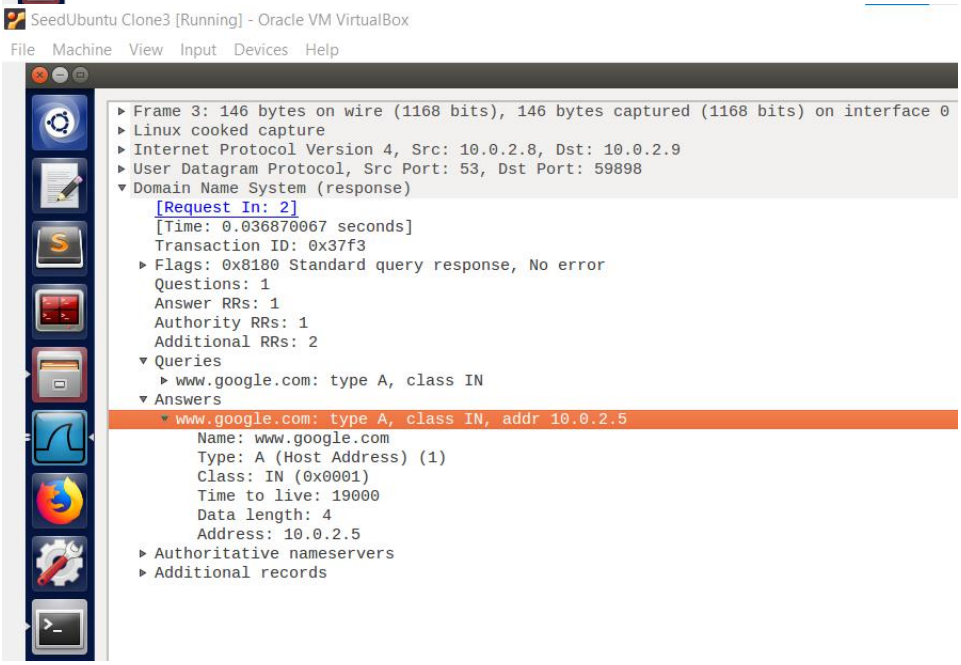


We can see that the attacker machine send's a DNS response as ip address as 10.0.2.5 and authoritative name sever as 10.0.2.19 for a DNS query made by DNS server for www.google.com .



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.

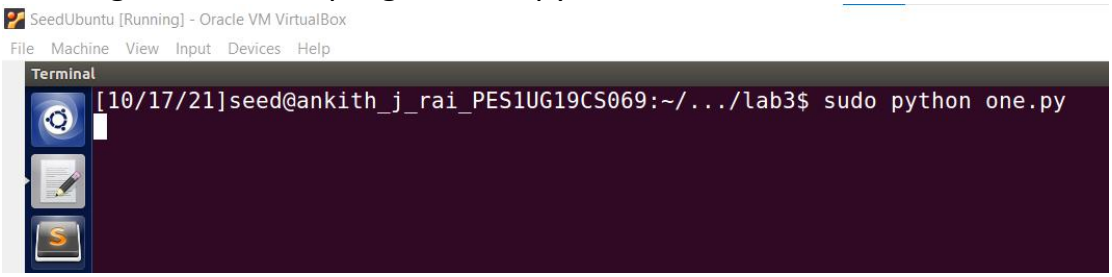




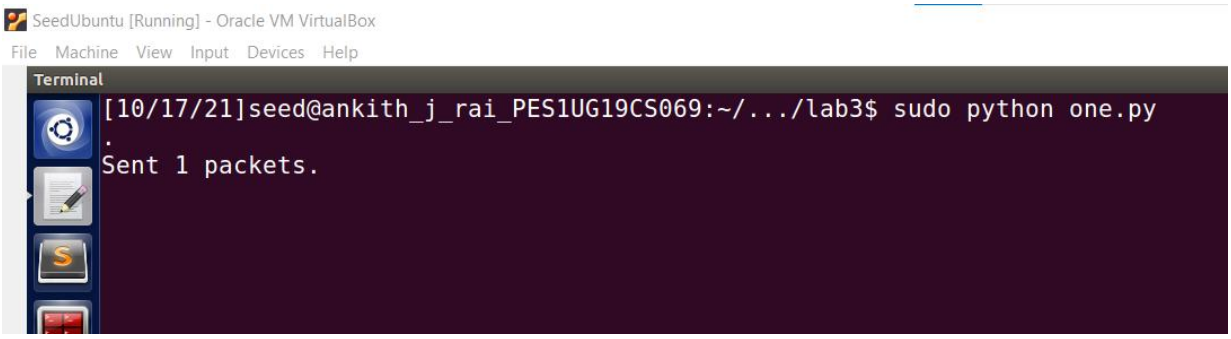
We can see from the above screenshot that the DNS cache has been poisoned and www.google.com has been mapped to 10.0.2.5(attacker ip address) in the DNS server itself.Hence for www.google.com DNS query , a DNS response is sent which tells that the ip address for the hostname is 10.0.2.5

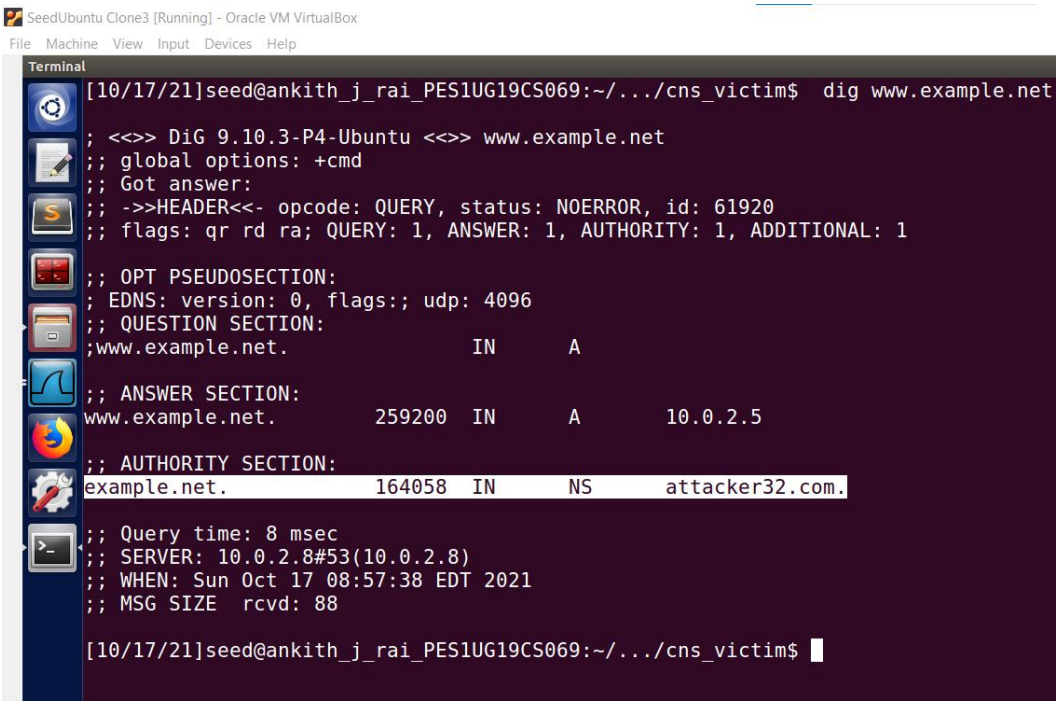
Task 7: DNS Cache Poisoning: Targeting the Authority Section

Running the attacker program one.py on attacker machine.

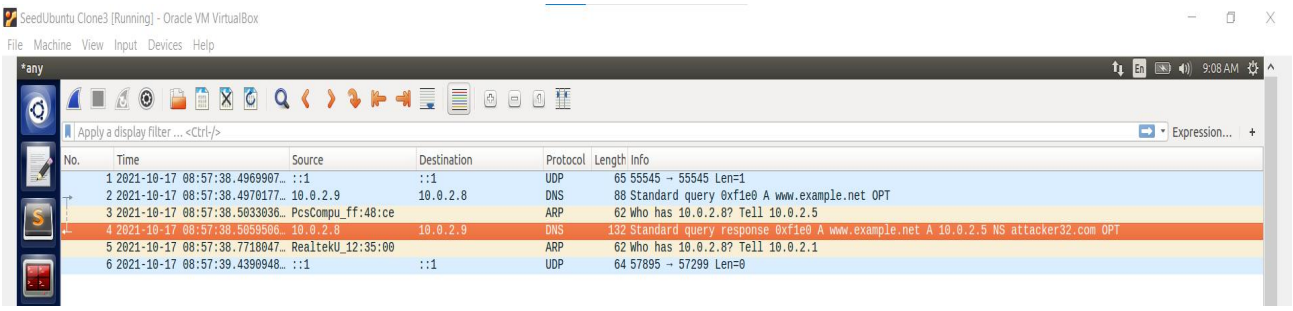


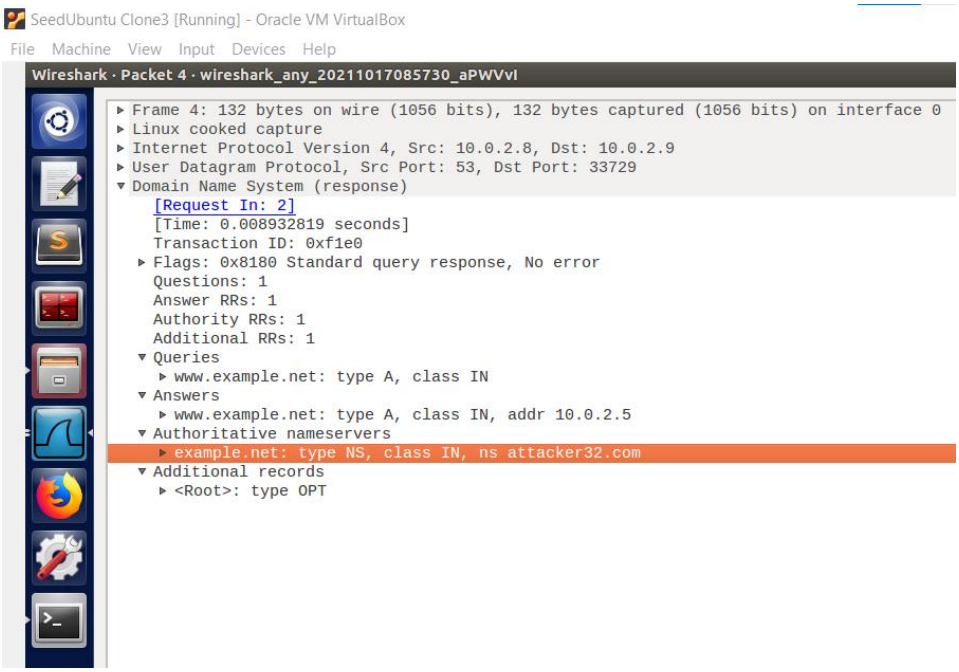
On running dig www.example.net





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

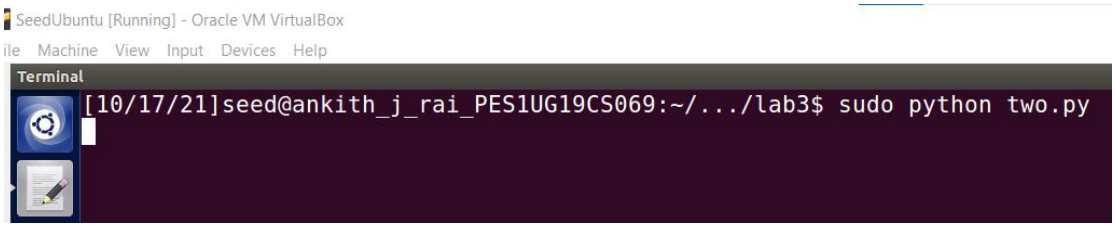




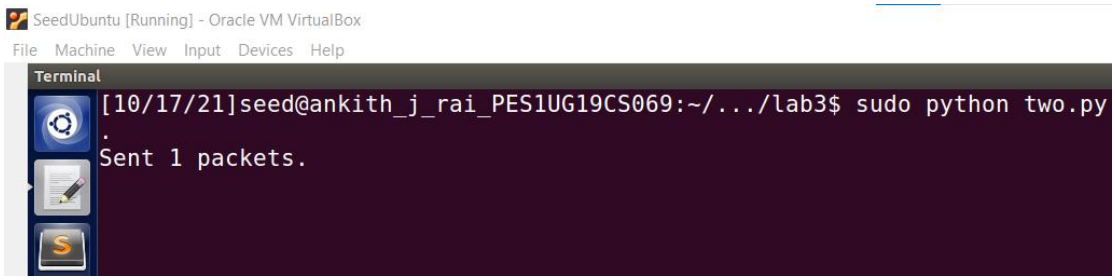
We can see from the above wireshark screenshots that DNS request has been sent to attacker32.com hence it show's our cache poisoning attack is successful.

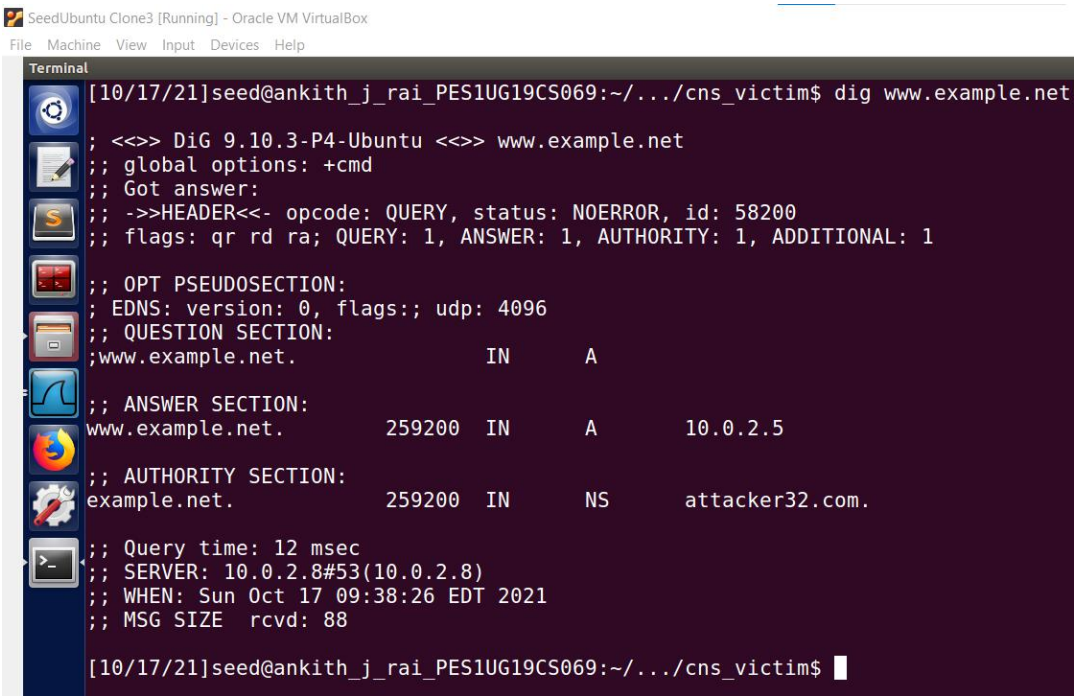
Task 8: Targeting Another Domain

Running the attacker program two.py on attacker machine.



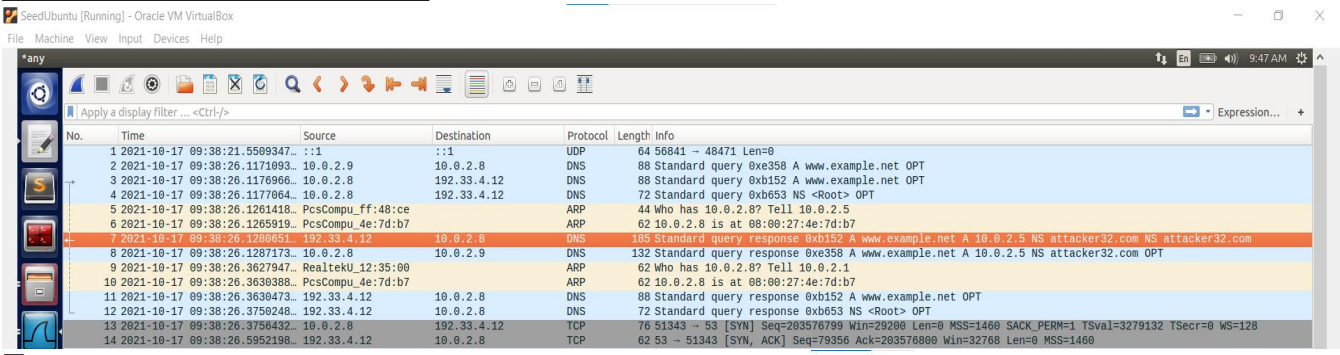
On running dig www.example.net

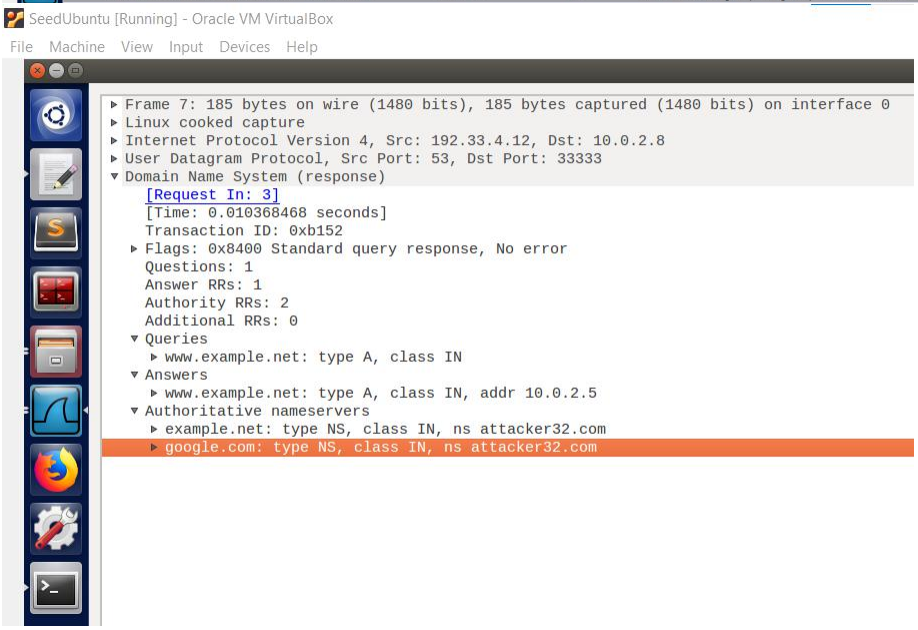




We can see from the above screenshot that the victim gets a forged reply from DNS server and authority section contains only attacker32.com not google.com .

Wireshark Screenshots:

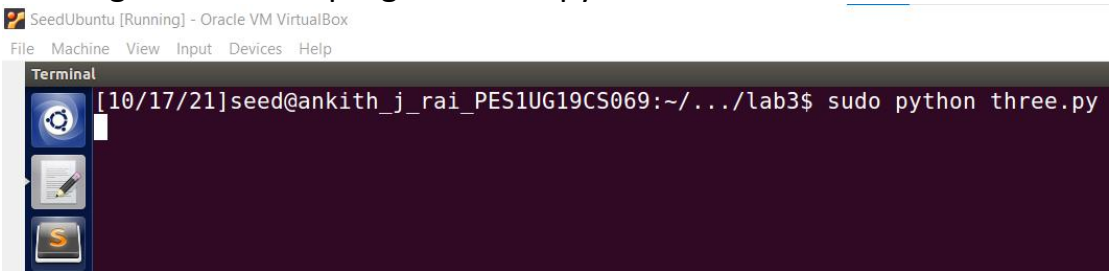




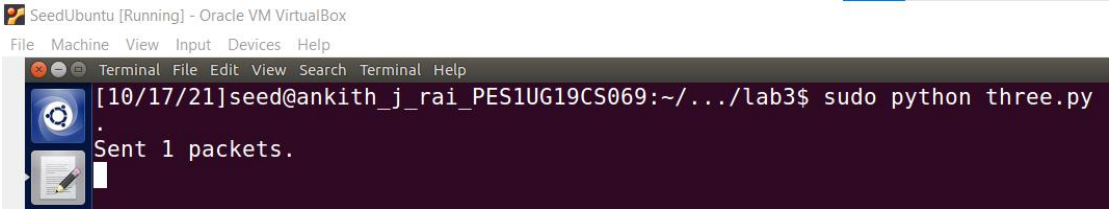
We can see that the two authoritative nameservers - example.net and google.com are present.
But the second record is discarded as it is fraudulent.

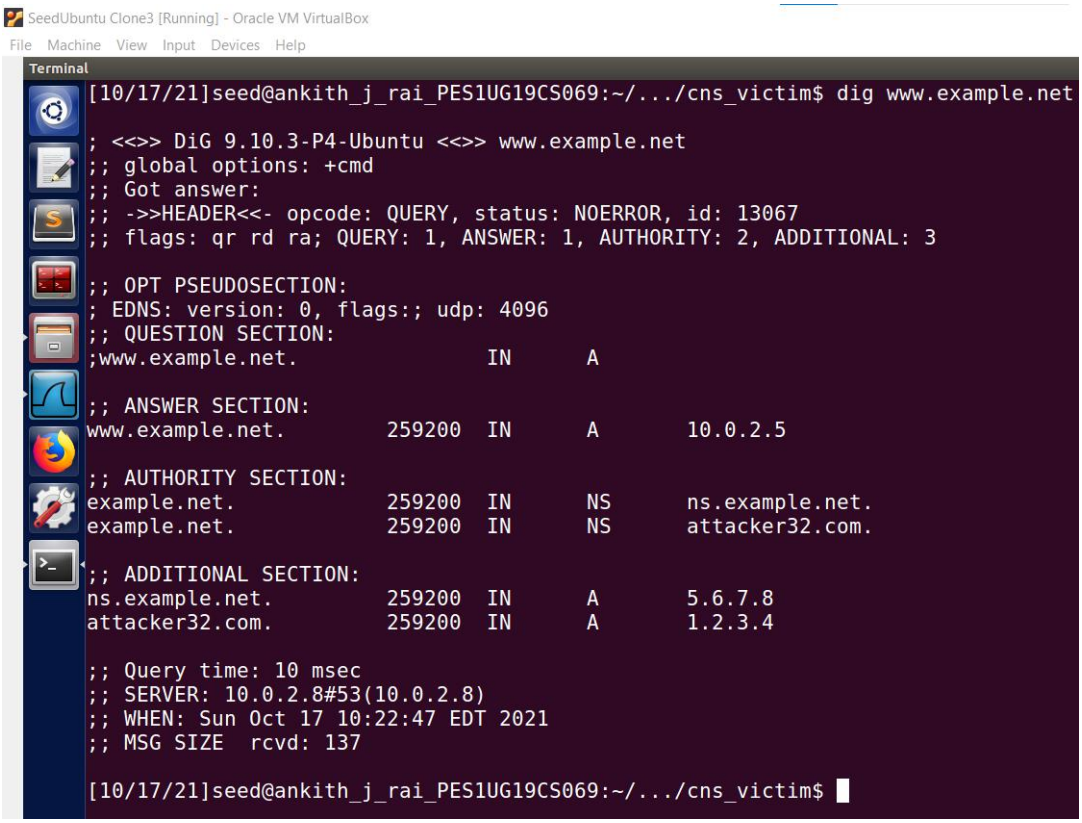
Task 9: Targeting the Additional Section

Running the attacker program three.py on attacker machine.

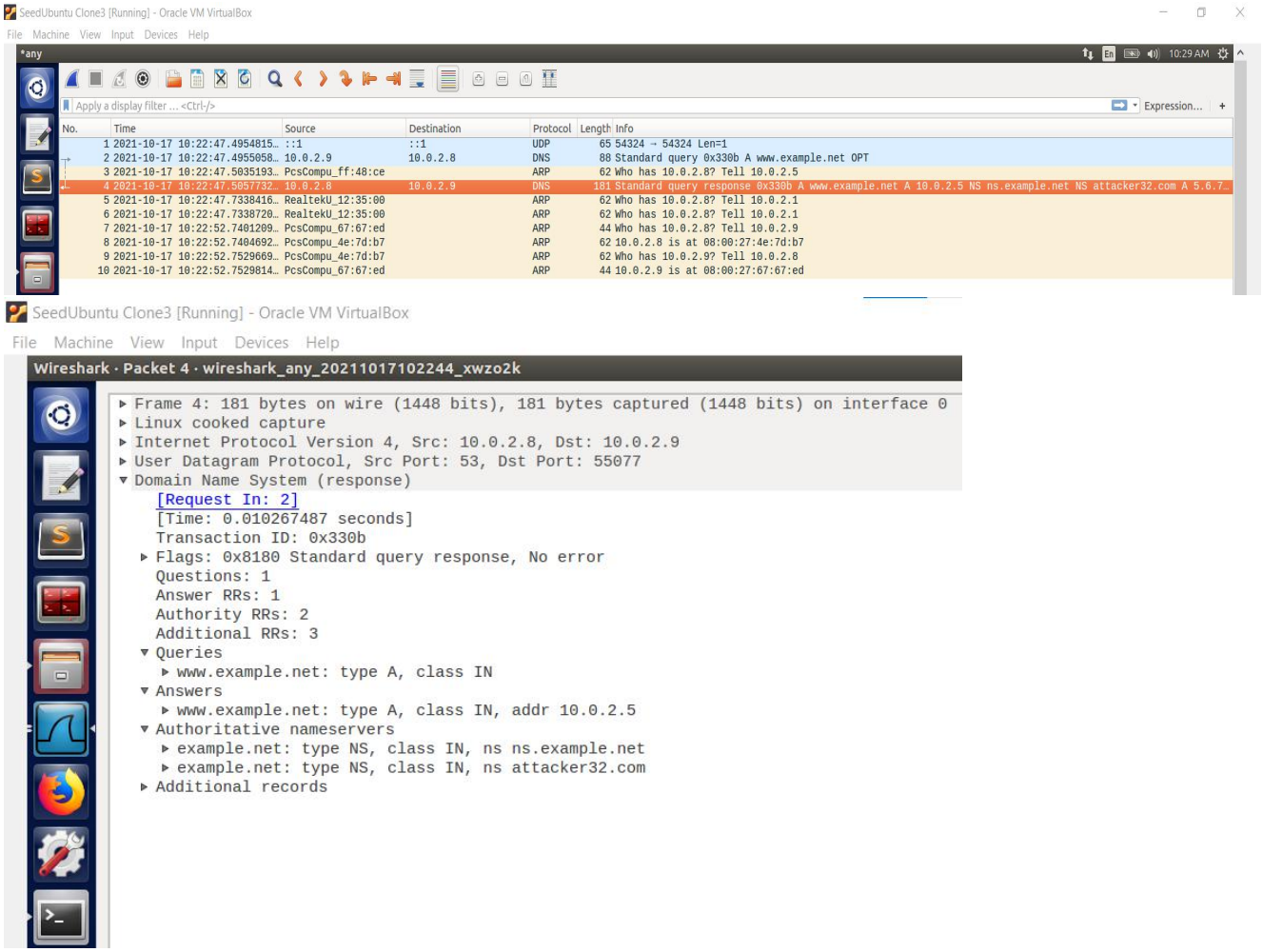


On running dig www.example.net on victim machine





Wireshark screenshot:



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.