Cyber and Network Security

Assignment-1

Submitted by:

Muhammad Osama Khalid

20I-1955 (MS-CNS)

Section: 1

Table of Contents

Part-1 Setting up Local DNS Attack Lab

In this part, I am going to set up the local DNS attack lab which I used later to perform various attacks in the next parts.

1. First, I download the Seed Virtual Machine from the SEED Lab website and then open the virtual box and click on add button. Then I add details of my VM and click Next. After that, I set the memory size (RAM) to 1 GB and click next. After that from the options, I select “Use an existing virtual hard disk file” and go to the path where I download the Seed VM and select that. After that, I click create to create my VM.

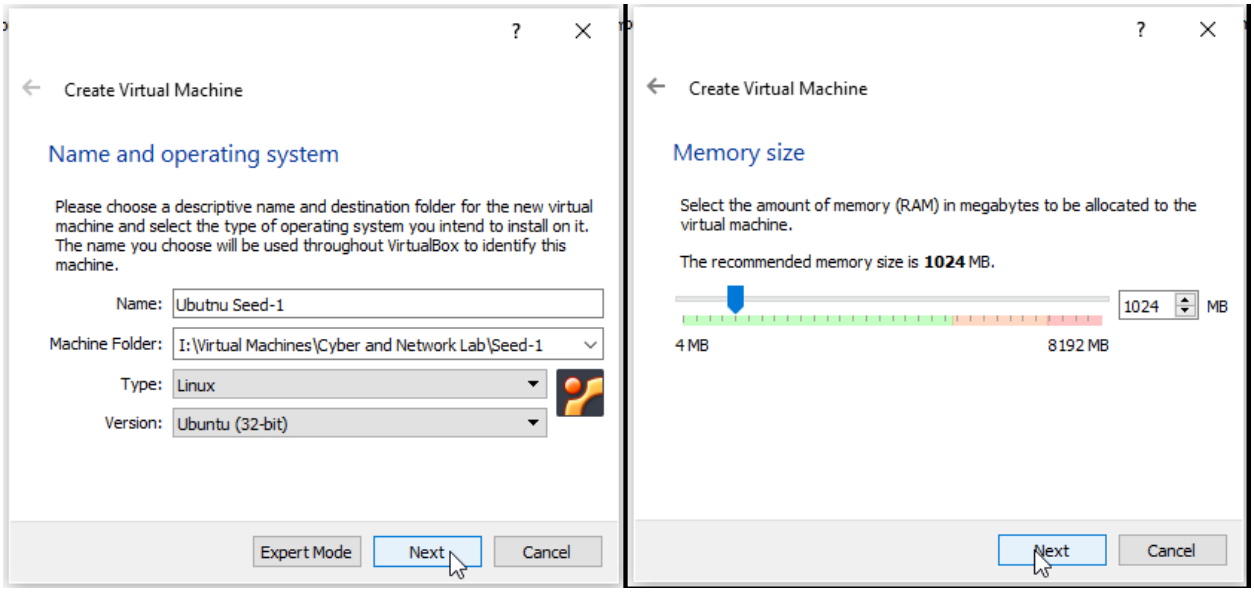

Figure 1: Creating the Virtual Machine

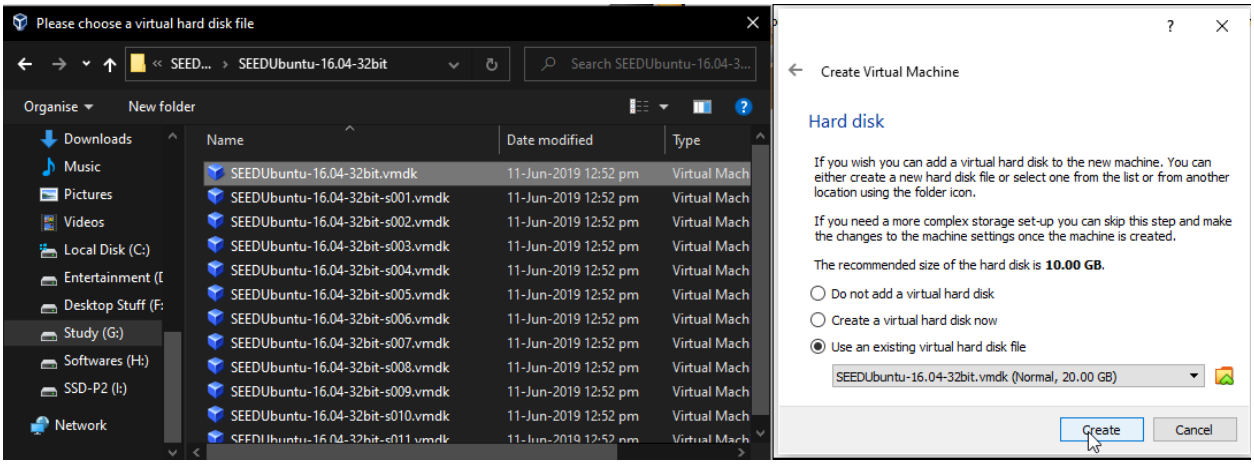

Figure 2: Creating Virtual Machine

2. After that, I go to the VM settings and click on the shared folder to create the shared folder between the host and the virtual machine so that every time I put something in that folder, it can be accessed by both VM and host machine.

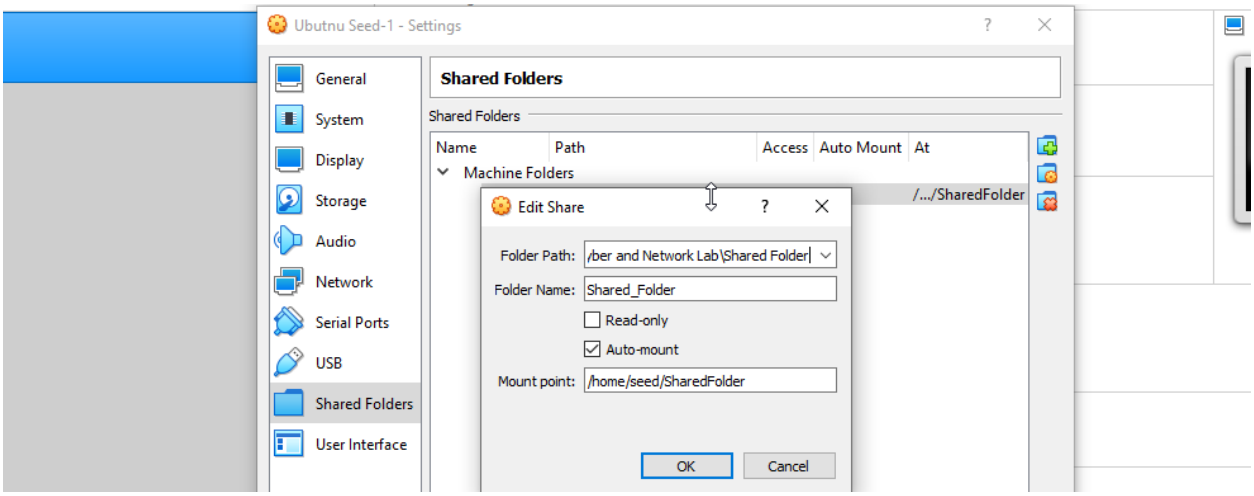

Figure 3: Go to Virtual machine Shared Folder Setting

3. After that, I click the start button to start the virtual machine. Then I open the "rc.local" file and add the following command in that file so that after rebooting the VM our shared folder will be accessible to us "**sudo mount -t vboxsf -o rw,uid=1000,gid=1000 share /home/seed/host**".

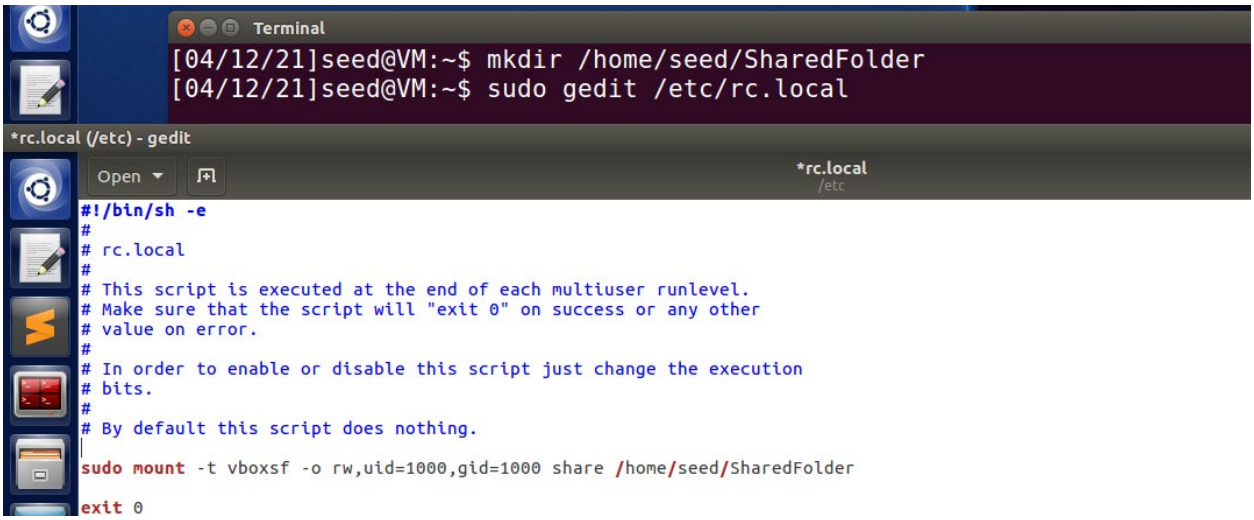

Figure 4: Opening and editing the re.local file

4. Then I open the terminal and write the command so that our user can get permission to view and edit the shared folder contents

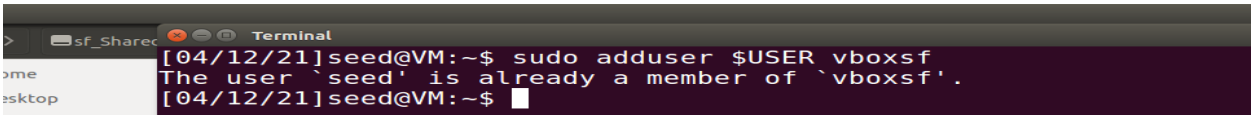

Figure 5: Granting the current user access to the shared folder

5. Then I power off the VM and go to the Virtual box setting and select Network from the tab on the left panel. Then I click on the "+" button to create a new NAT Networks adapter. After that, I open the virtual machine setting, select Network from the tab on the left panel, and staying on Adapter 1 and under enable network adapter I click on the "Attach to" drop-down menu and select NAT Network adapter that I just created and click ok.

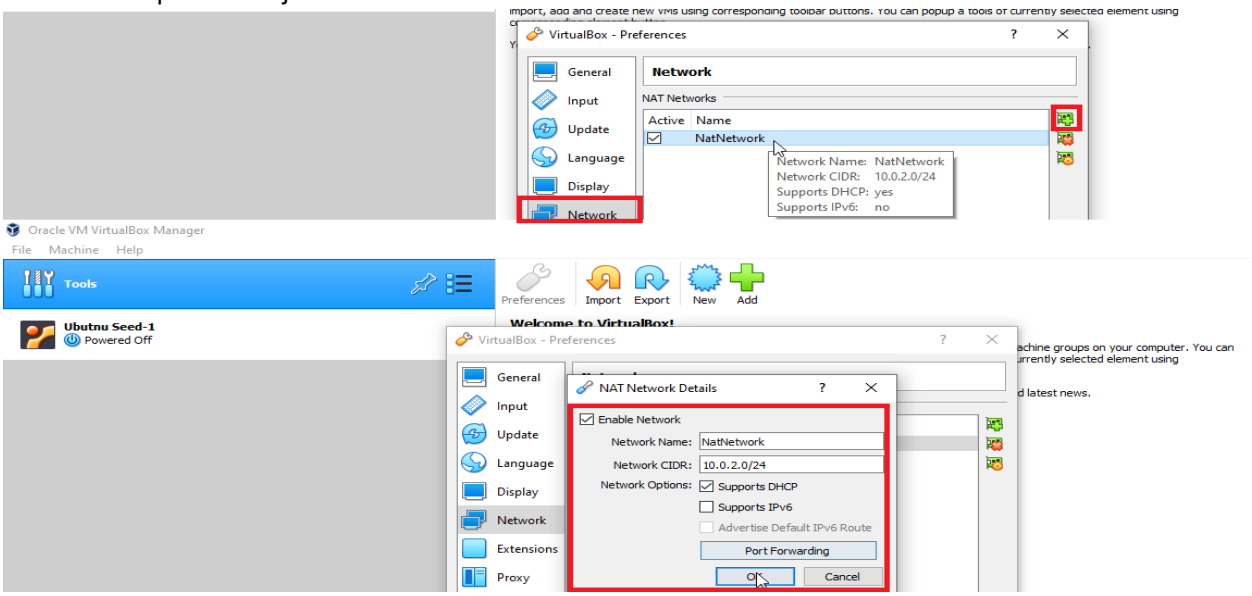

Figure 6: Creating new Nat Adapter from Virtual Box Setting

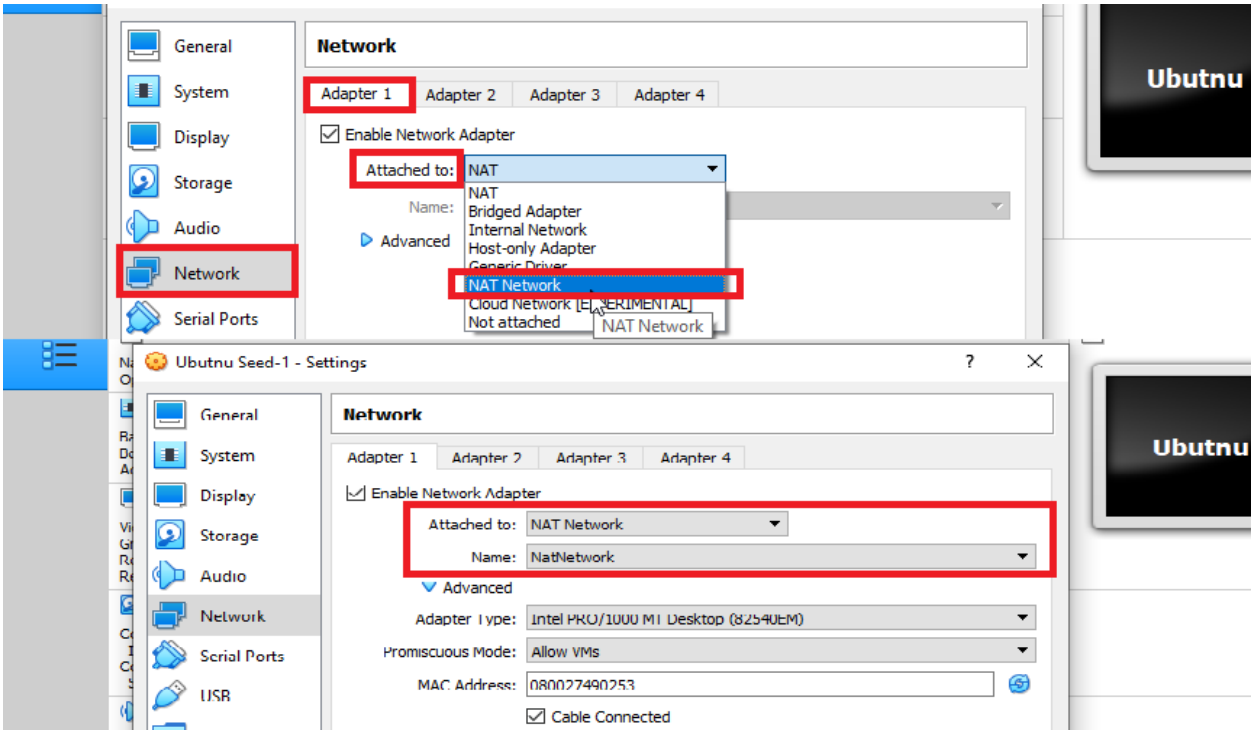

Figure 7: Adding Nat adapter in Virtual Machine setting

6. Then I right-click on the virtual machine and click then on the clone. Then in the popup dialogue box, I enter the clone machine name and select the path where I want to store my clone virtual machine and then I click on the Next button. Then from the clone type options, I select “Full done” and click on the clone button. I repeat this step one more time so that I can have one actual virtual machine and two clone machines.
7. Then I change the names of my virtual machines to Ubuntu Seed-1 (Attacker), Ubuntu Seed-2 (Server), and Ubuntu Seed-3 (User).

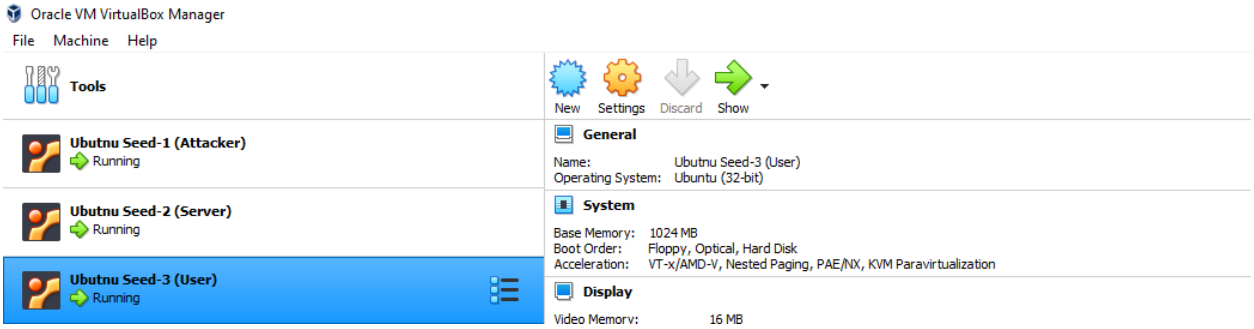

Figure 8: Virtual Machines to be used in Attack Lab

8. After that, I run all my virtual machines and open the terminal to check the IP address of all my virtual machines.

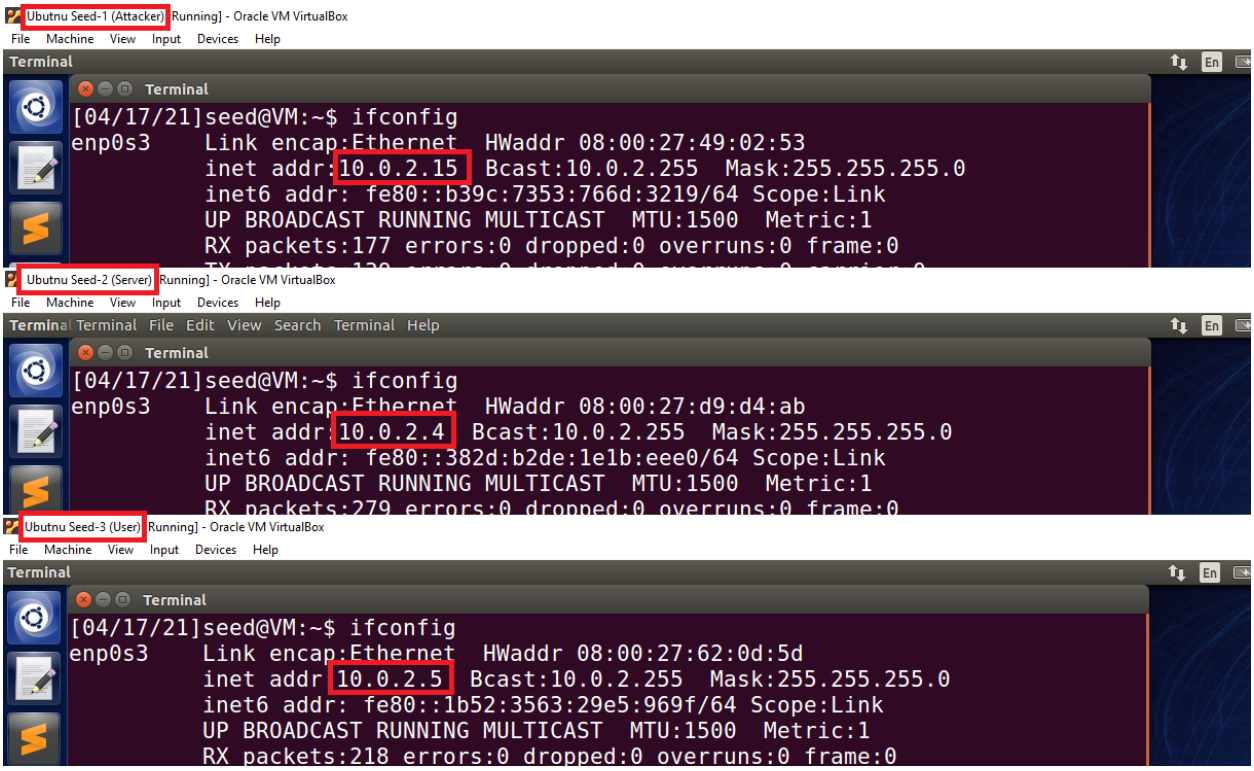

Figure 9: Ip addressed of my virtual machines

9. Then from the user virtual machine, I open the terminal and ping both attacker and server machines to check that all the machines are connected or not.

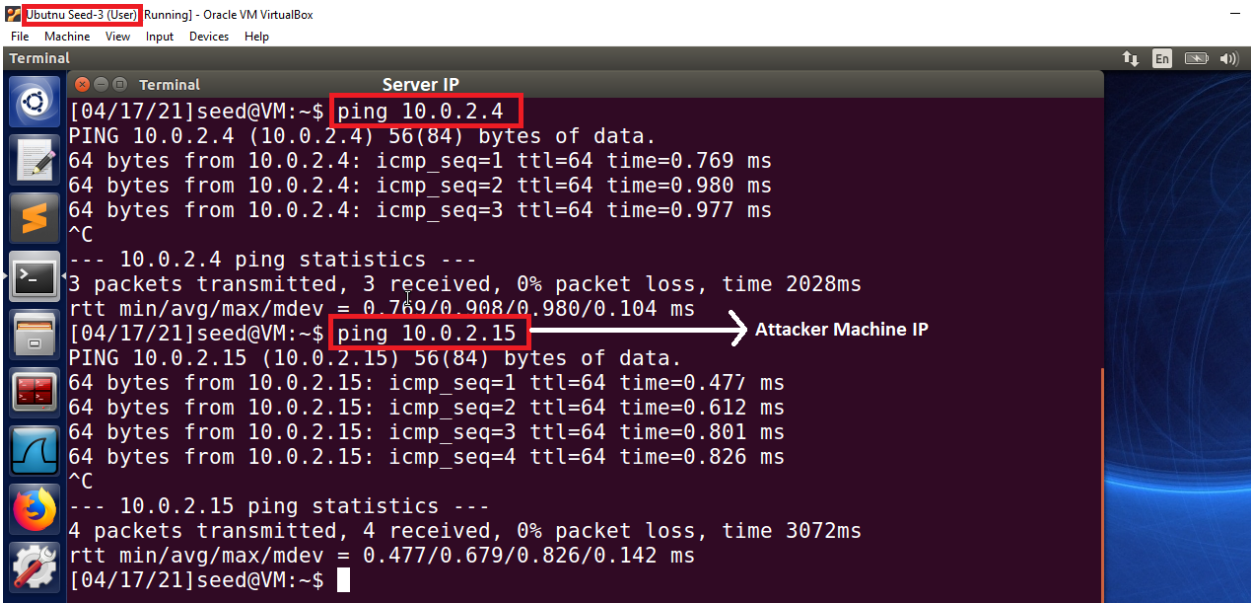

Figure 10: Ping request to Server and Attacker machine

10. Ip addresses of my machines
    - Attacker Machine IP: 10.0.2.15
    - Server IP: 10.0.2.4
    - User Machine IP: 10.0.2.5

Part-2 Setting Up a Local DNS Server

Now here we have to set up the local DNS server which we will be using in the later tasks.

Task-1 Configure the user machine

In this task, we have to configure the user machine so that whenever the user requests the DNS services it would be served by our DNS server. For that, I perform the following tasks.

1. First, I need to add the “nameserver 10.0.2.4” in the “resolv.conf.d/head” file so that this server will be used as the primary DNS server. So I open the terminal and then open the “resolv.conf.d/head” file and add the name server to that file.

Figure 11: User: Opening and editing the resolv.conf.d/head file

2. Then I run the “sudo resolvconf -u” command so that the changes I made in the “resolv.conf.d/head” file take effect.

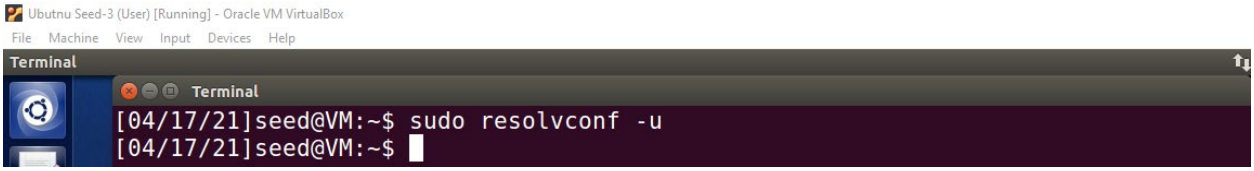

Figure 12: User: Run the "resolv -u" command so that changes take effect

3. Then I request my DNS server the IP address of the www.nu.edu.pk website using the dig command.

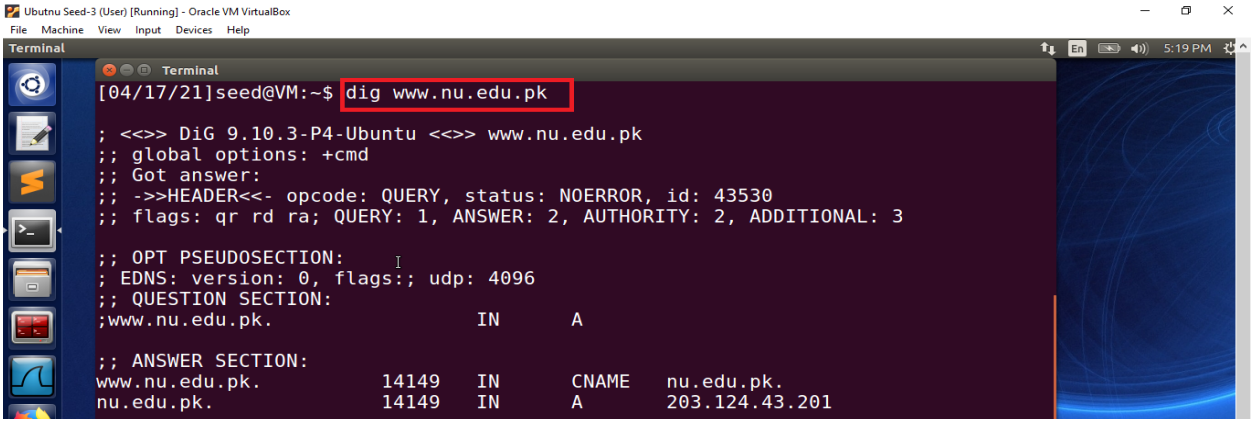

Figure 13: User: Using dig command to request the IP address of the website

4. While viewing the result of my query I found that the request for the IP address of the website is fulfilled by my Local DNS server having IP address “10.0.2.4”.

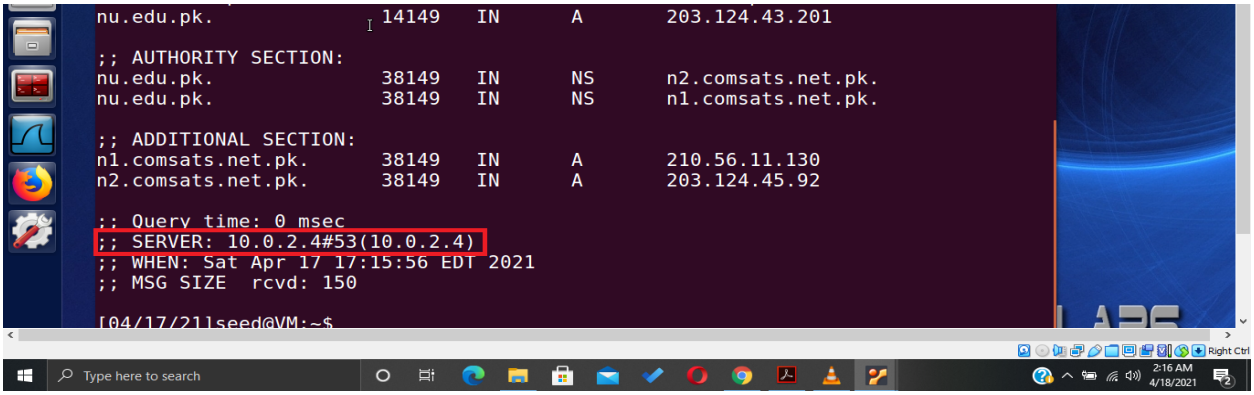

Figure 14: User: Request fulfilled by my DNS server

Task-2 Set up a Local DNS Server

In this task, I have to set up the BIND 9 server program so that my DNS server will set up for that server program.

Step-1 Configure the BIND 9 Server

1. In this step, I go to the server machine, and then I need to enter the DNS cache dump file entry in the “name.conf.options” file so that the cache of my DNS server will be dumped in that file. For that I open the terminal and open the “name.conf.options” file using the nano editor and add the dump file entry with the path to the dump file “dump.db”.

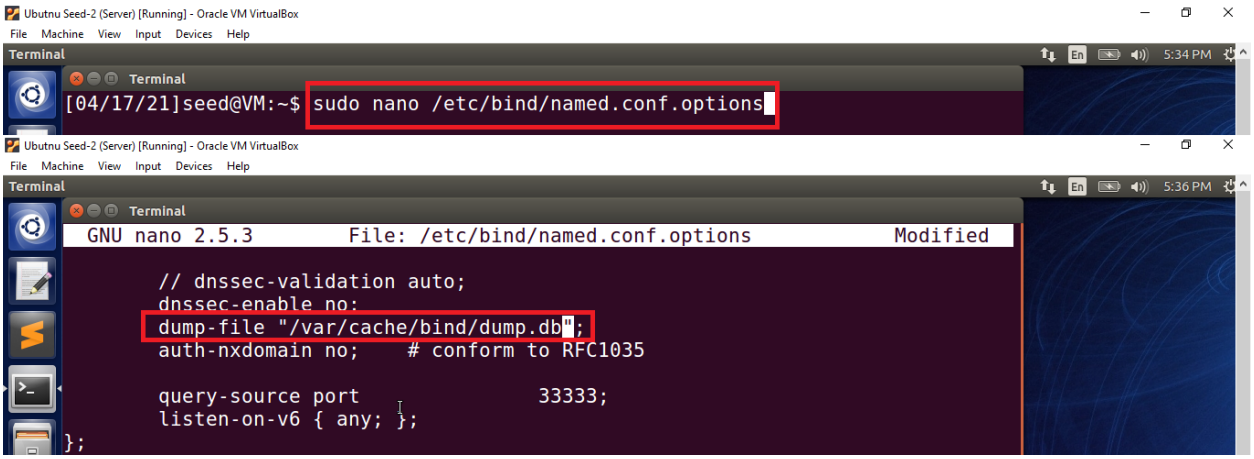

Figure 15: Server: Setup DNS Server cache file

2. After that, I dump the cache of my DNS server in the dump.db file using the “rndc dump.db -cache” command. Then I flush the DNS cache using the “rndc flush” command

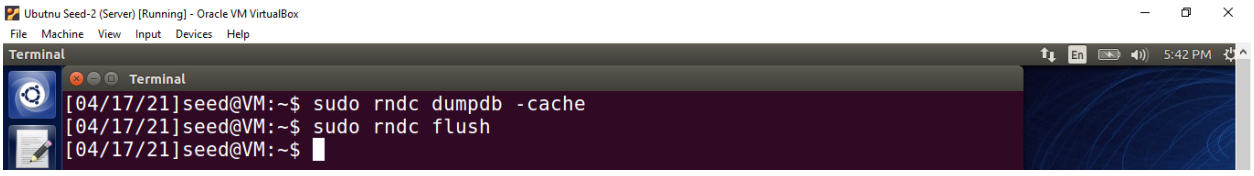

Figure 16: Server: Dump the cache and then flush it

Step-2 Turn off DNSSEC

In this step, I need to turn off the DNS security so that it becomes vulnerable to spoofing attacks. For that I again open the “name.conf.options” file and comment out the “dnssec-validations” entry and add the “dnssec-enable” entry in the file and set it up to no. So that it will disable the DNS Security.

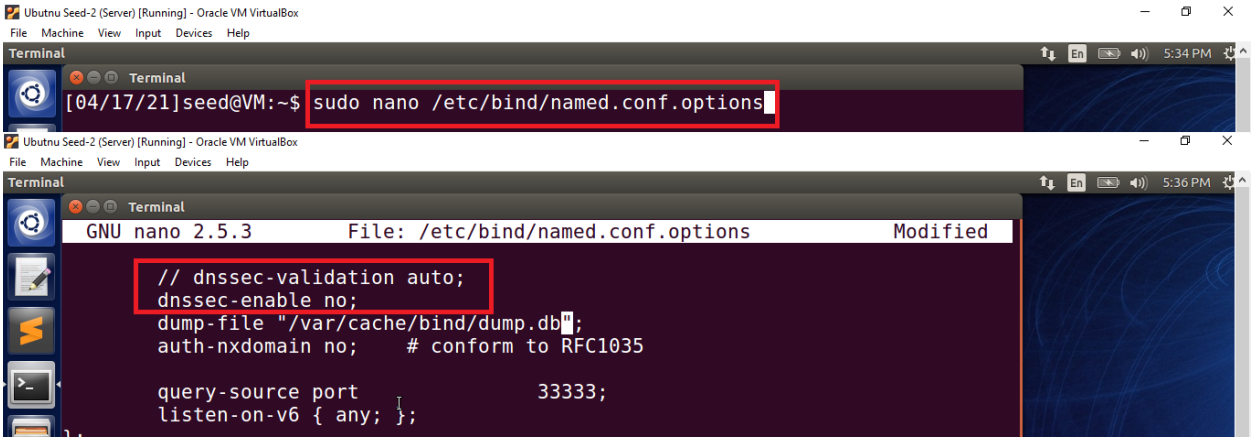

Figure 17: Server: Turning of DNS Server Security

Step-3 Start DNS Server

Now I restart the DNS server using the “service bind9 restart” command.

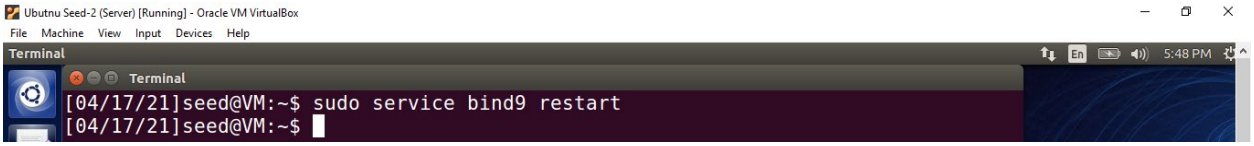

Figure 18: Server: Restart the DNS server

Step-4 Use the DNS server

1. In this task, I again switch to the user machine and run the Wireshark. After that, I ping the Google and Facebook websites from the terminal.

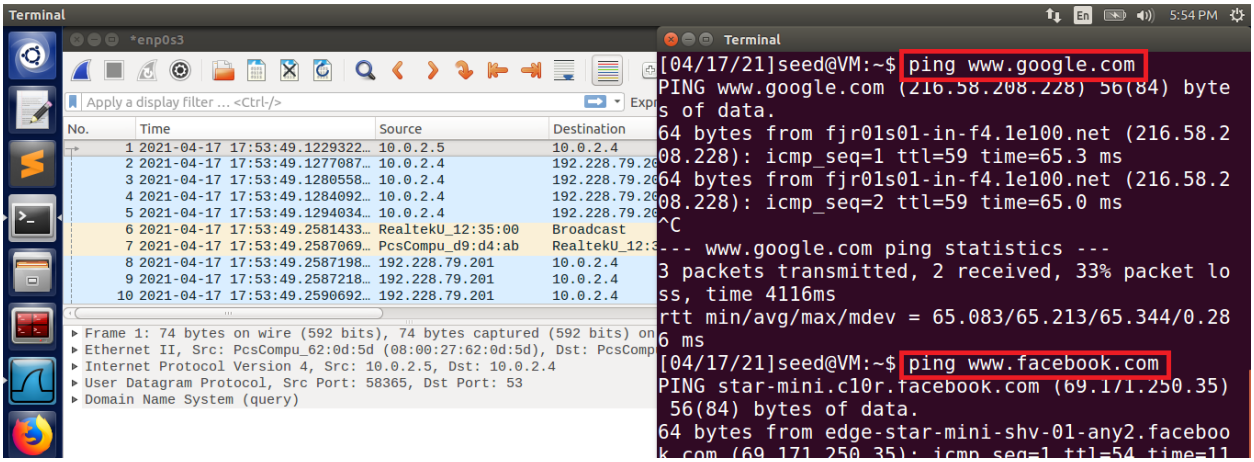

Figure 19: User: Ping request to the DNS Server

2. After that, I go to the Wireshark to see the packets captured by it.
3. Now here we can see that the user machine sends the ping request to the DNS server for www.google.com and in return server respond to the user with the IP address and name servers of www.google.com.

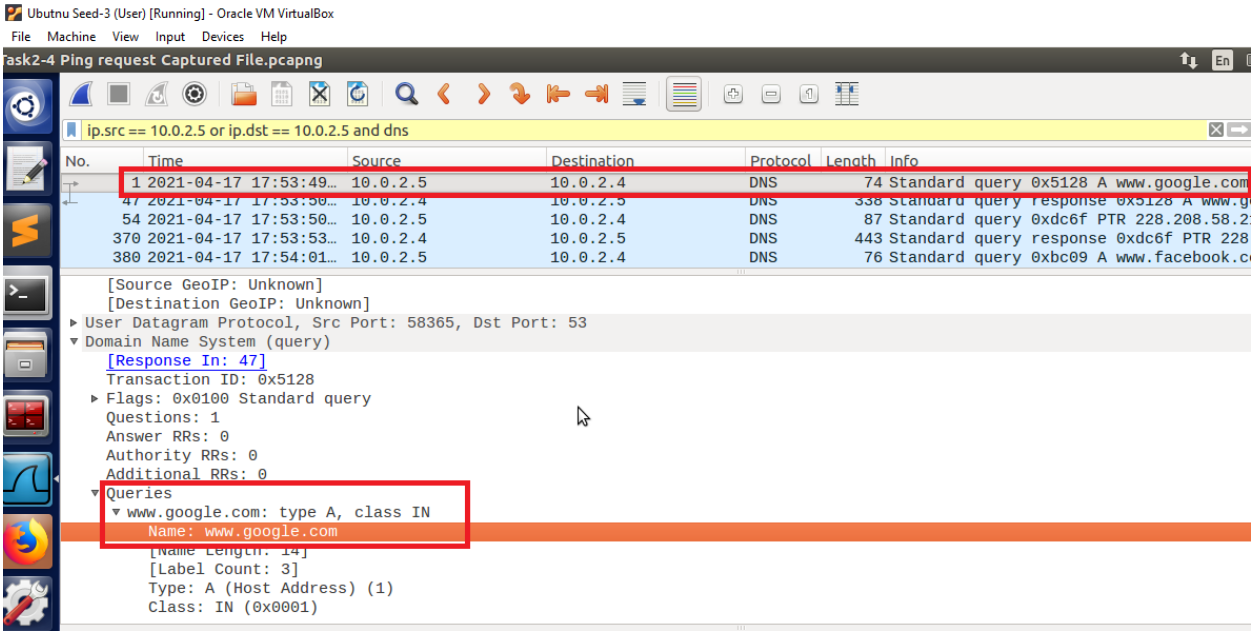

Figure 20: User: The user sends the ping request to the DNS server

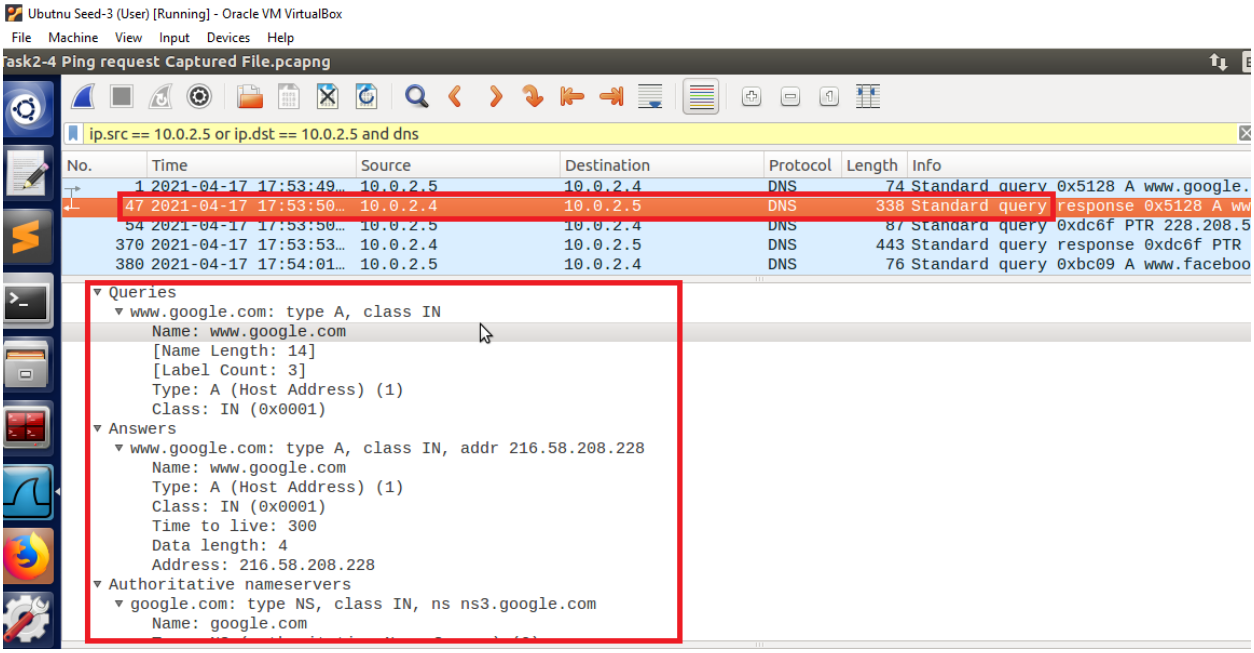

Figure 21: User: DNS server responds to the User request

4. Again we can see that the user machine sends the ping request to the DNS server for www.facebook.com and in return server respond to the user with the IP address and name servers of www.facebook.com.

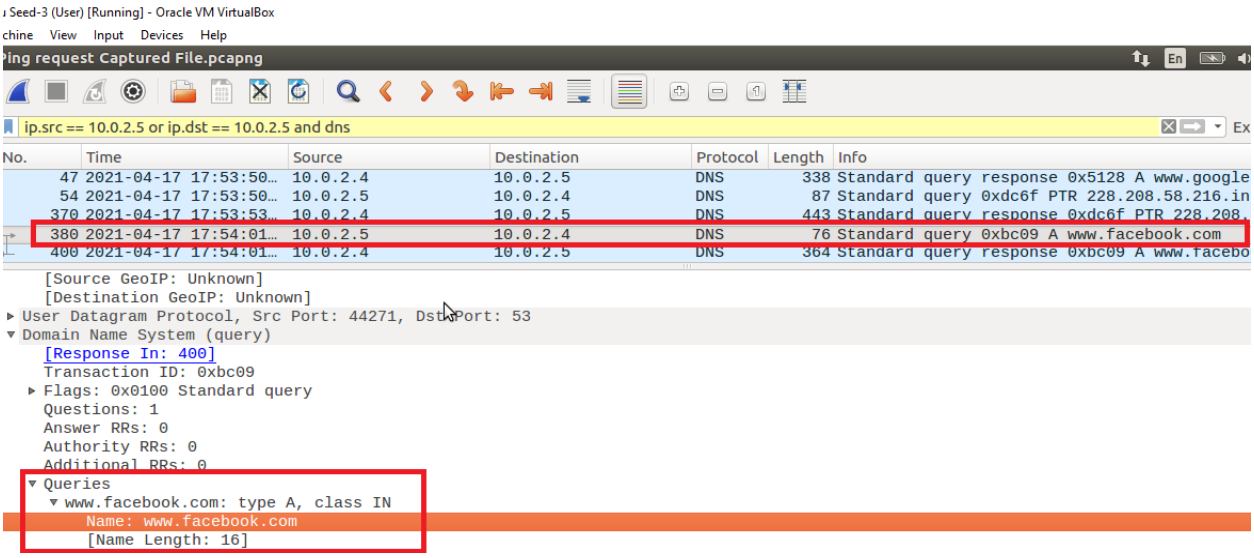

Figure 22: User: The user sends the ping request to the DNS server

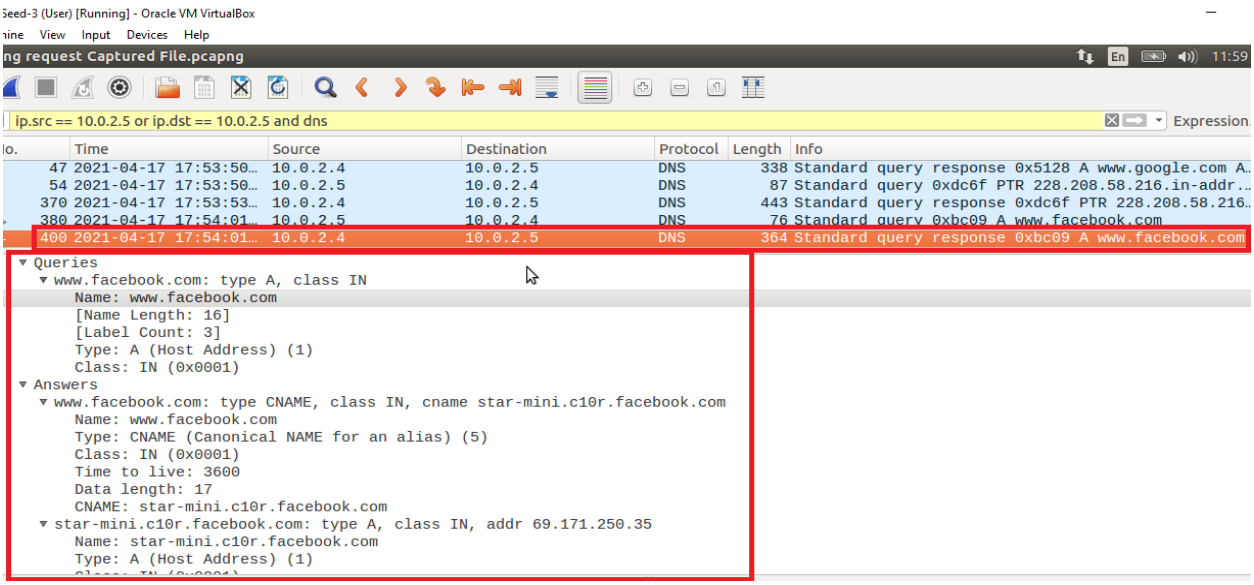

Figure 23: User: DNS server responds to the User request

5. Now here to fulfill the request of the user first server checks its cache for the details of www.google.com and www.facebook.com. If it found the record in its cache it directly responds to the user. If the record is not present in the cache then the server forwards the request to the root server and waits for the response. When it gets the response from the root server, the DNS server first saves the information in its cache so that the next time the user requests the same query it directly responds to the user, then it sends back the response to the user machine.

Task-3 Host a zone in the Local DNS Server

In this task, I go back to my server machine and set up my Local DNS Server as the authoritative name server for the www.example.com domain.

Step-1 Create Zones

In this step I need to create two-zone entries for www.example.com, first for forward lookup and second for reverse lookup in the DNS server by adding them in the "named.conf.local" file. For that I open the terminal and open the "named.conf.local" file using the nano editor and enter the zone entries in that file.

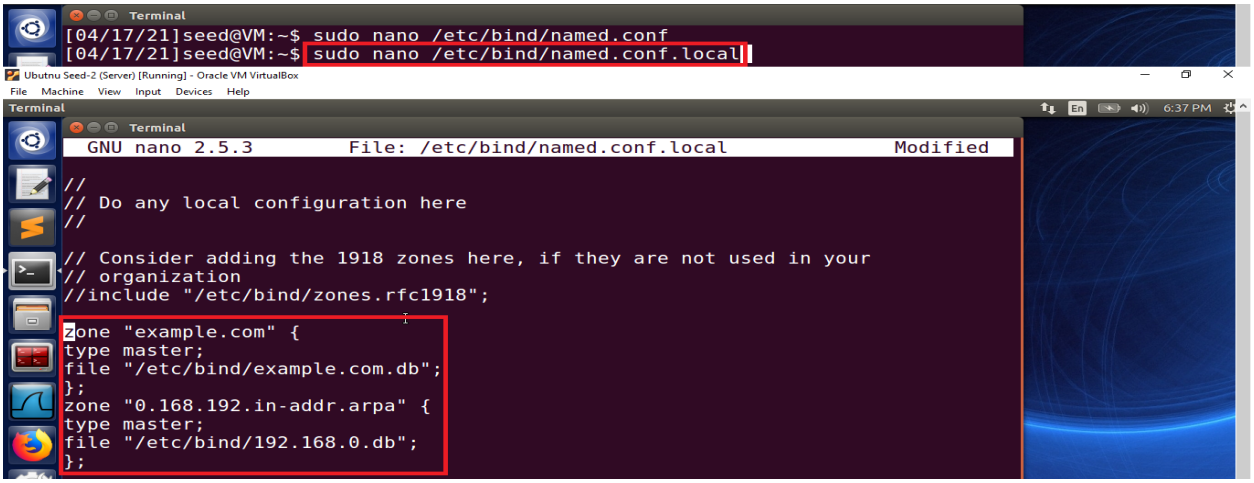

Figure 24: Server: Entering the zone details in the named.conf.local file

Step-2 Setup the forward lookup zone file

1. In this step, I create the zone file "example.com.db" in the "/etc/bind" directory. This is the file where DNS resolution is stored.

Figure 25: Server: Creating forward lookup zone file

2. After that, I open the file and add the zone code in that file.

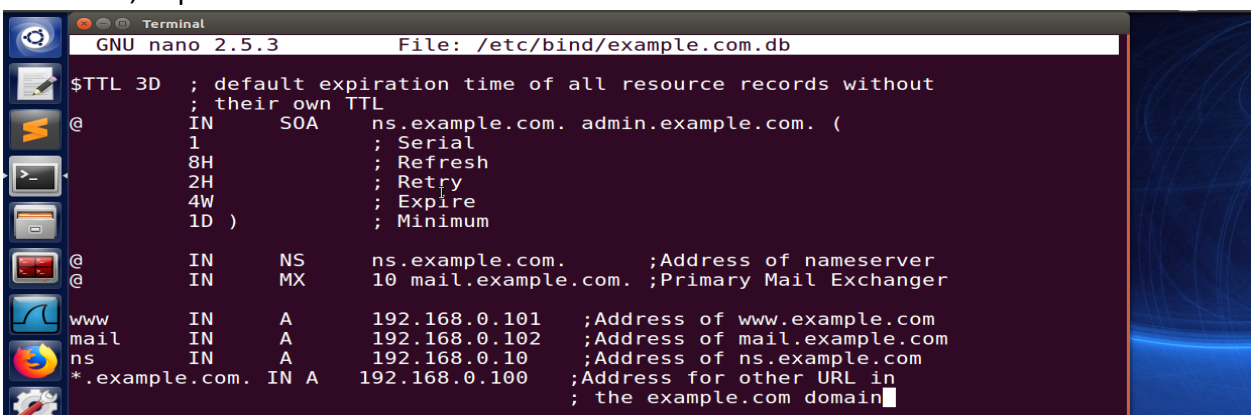

Figure 26: Server: Adding conde in the forward zone file

3. In the code, we add the detail of the name and mail server of the www.example.com file with their IP addressed and also add the details of when to refresh, retry and expire the record.

Step-3 Set up the reverse lookup zone file

1. In this step, I create the DNS reverse lookup file "192.168.0.db" in the "/etc/bind" directory. This is the file where DNS resolution is stored.

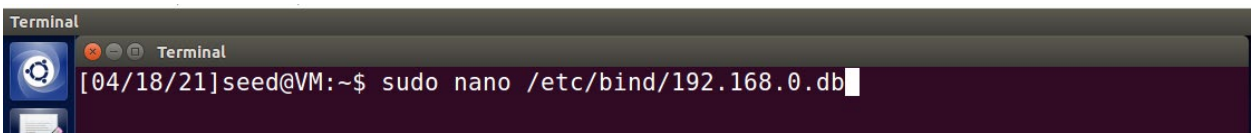

Figure 27: Server: Creating reverse lookup zone file

2. After that, I open the file and add the reverse lookup code in that file.

Figure 28: Server: Adding the reverse lookup code in the reverse lookup zone file

Step-4 Restart the BIND server and test

1. First, I restart the bind server using the "service bind9 restart" command.

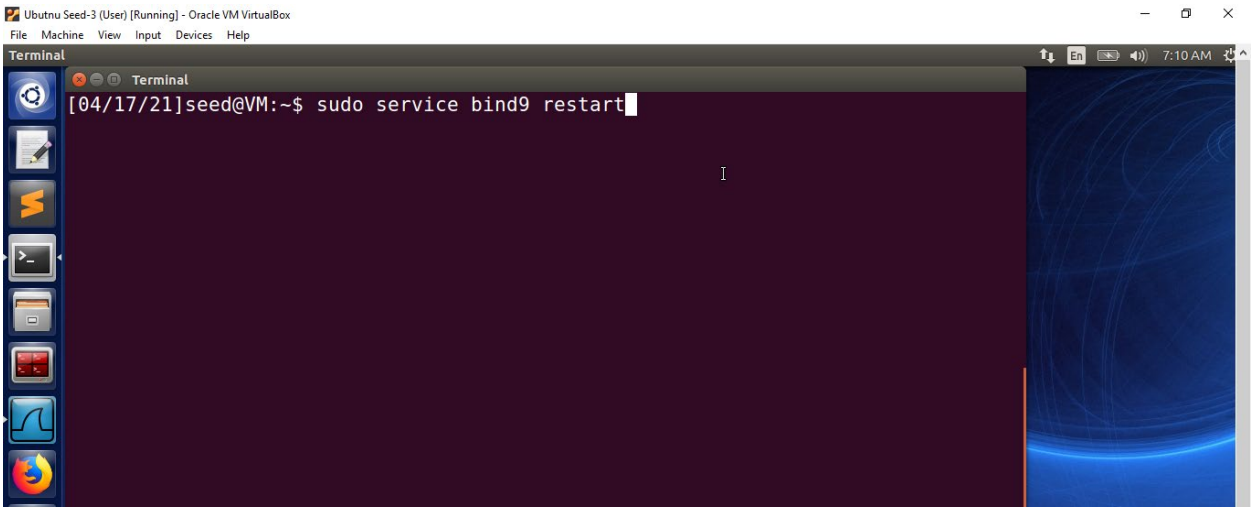

Figure 29: Server: Restart the DNS Server

2. Then I go back to the user machine and ask the local DNS server for the IP address of www.example.com using the dig command.

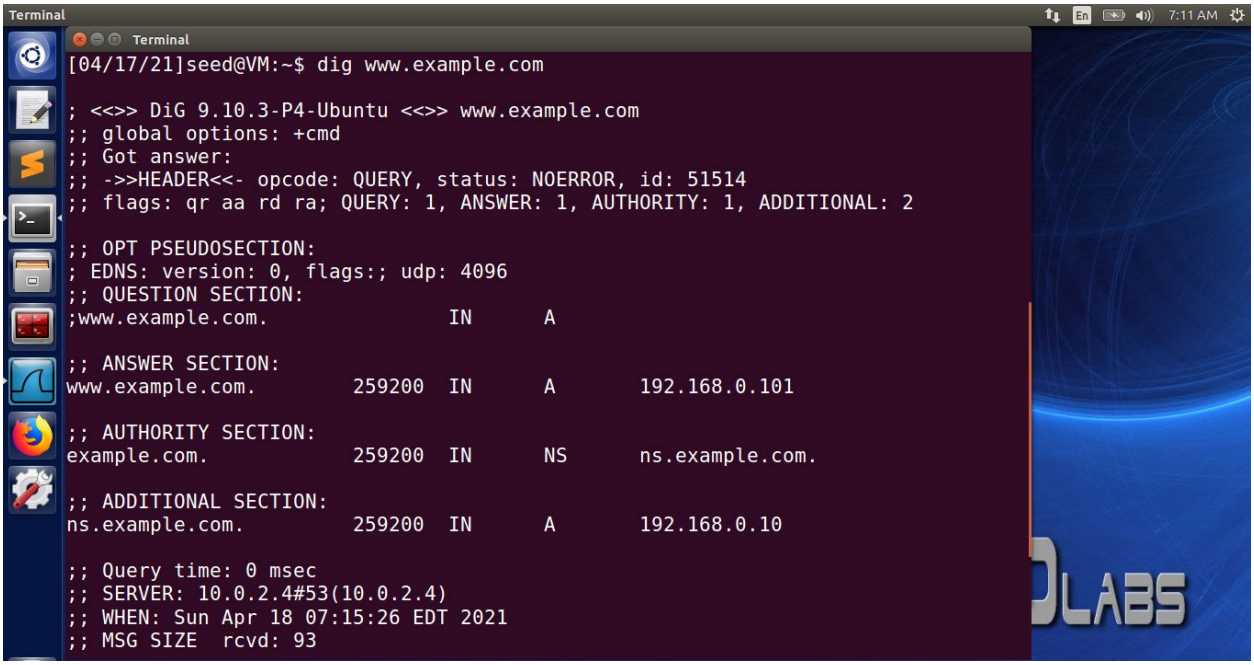

Figure 30: User: Dig command to ask DNS server the IP of www.example.com

3. Now here we can see that my DNS Server responds to the user with the IP address, name server, and its IP address of www.example.com.

Part-3 Attack on DNS

Now in this part of the lab, we are going to perform different attacks on DNS servers and user machines through the Attacker machine.

Task-4 Modifying the Host file

1. In this task, I first go to the attacker machine and from here I access the user machine using the “ssh” command because both attacker and user are on the same network and the user machine is also not secure.

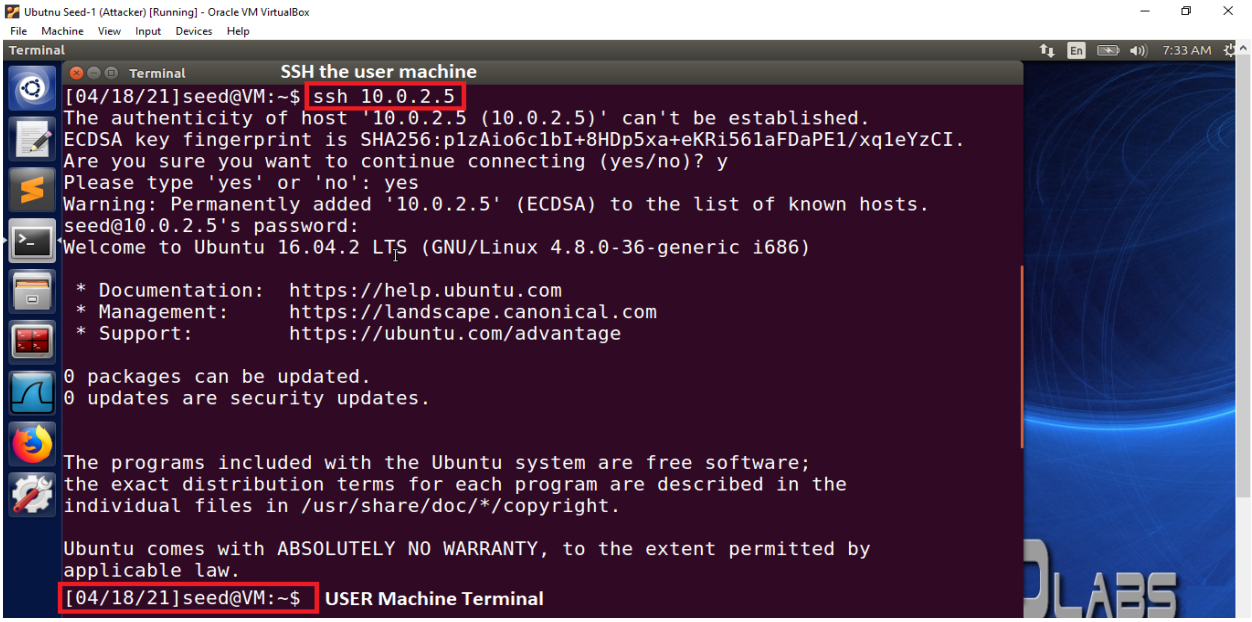

Figure 31: Attacker: SSH the user machine

2. After that from the attacker machine, I open the hosts file in the user machine and add the host and IP address of www.example.com and www.bank32.com in that file so that the user can be redirected to the attacker's malicious website.

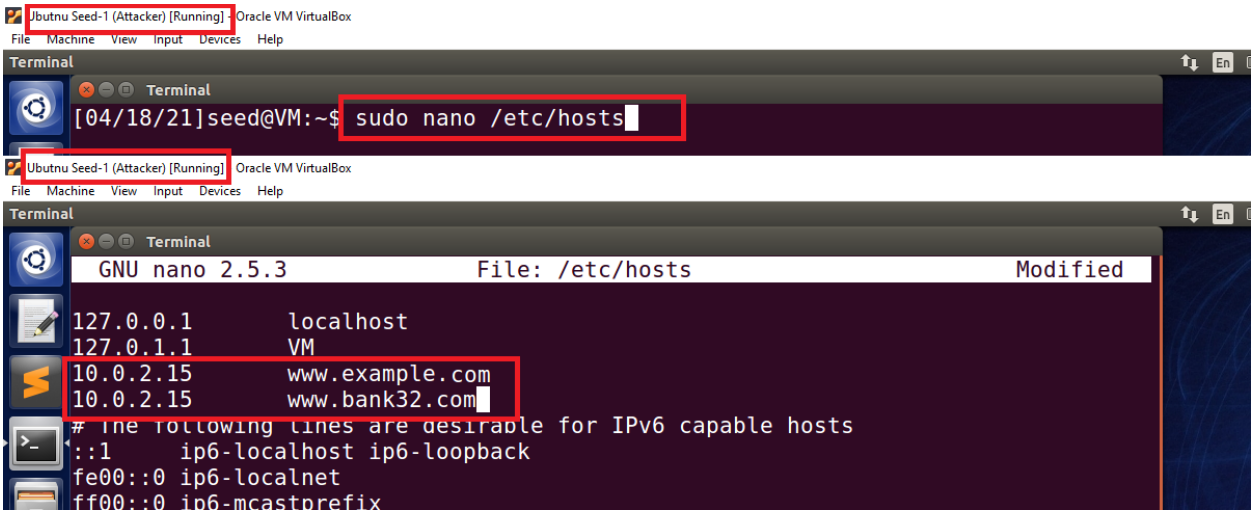

Figure 32: Attacker: Opening the hosts file and adding the malicious entry in the file

3. After that, I go to the user machine and open the terminal to ping the website www.bank32.com to see the response and found that the response of the ping is coming from the malicious IP which the attacker set up in the user's machine "hosts" file

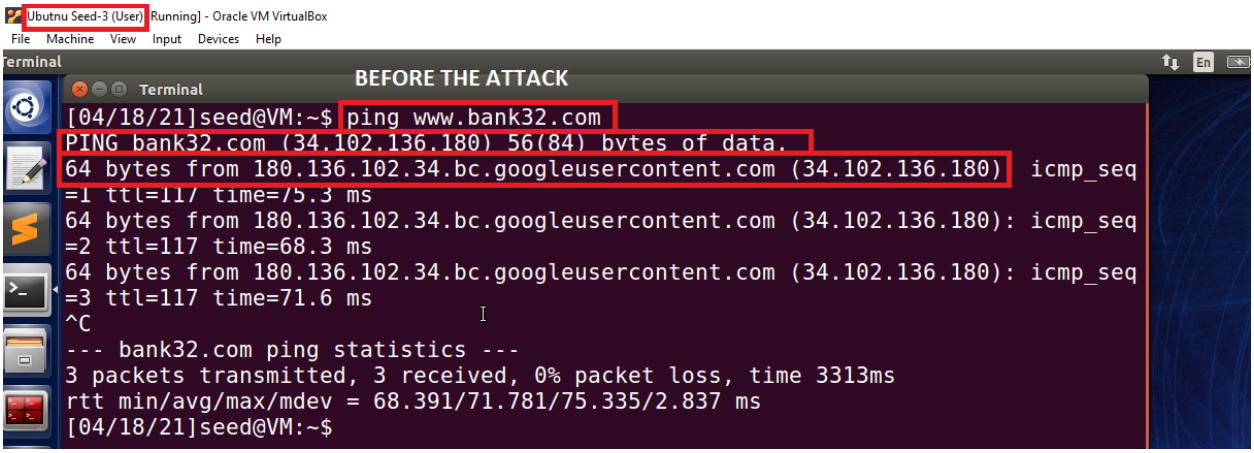

Figure 33: User: Ping response before the attack

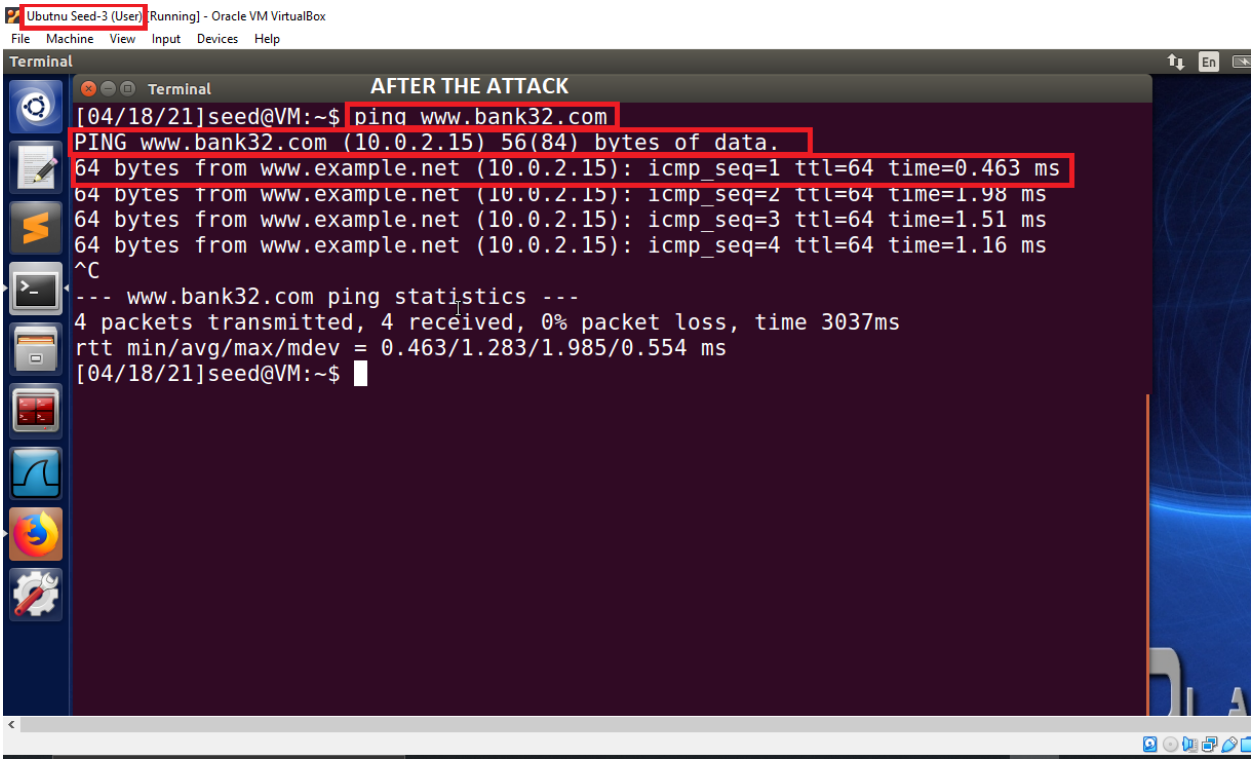

Figure 34: User: Ping response after the attack

4. Now here we can see that in the ping response of www.bank32.com before the attack user is getting a response from the actual IP and the google server but after the attack user is getting the response from the attacker's malicious IP.

Task-5 Directly Spoofing Response to User

1. In this task, we have to capture the user's DNS request to resolve the IP address of the website and spoof that request with the fake DNS response.
2. For that, I go to the attacker's machine and use the Netwox tool's 105 utility "Sniff and send DNS answers" to attack the user to sniff its DNS request and spoof it with the attacker's response.
3. To start the attack I open the terminal in the attacker machine and run the Netwox command "Netwox 105 -h www.example.net -H 10.0.2.15 -a ns.example.net -A 10.0.2.4 -f "src host 10.0.2.5" ". In this command "-h" defines the hostname, "-H" defines the host IP, "-a" defines the authoritative nameserver, "-A" defines the nameserver IP and "-f" defines the filter.

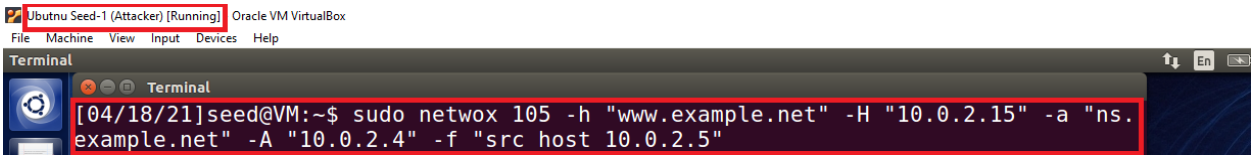

Figure 35: Attacker: Runs the Netwox command

4. Then that I go to the server machine and flush the DNS server cache.

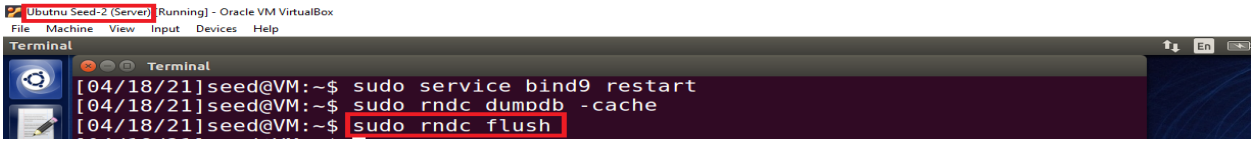

Figure 36: Server: Flush the DNS Server Cache

5. After that, I go to the user machine, open the terminal and then ask the local DNS server for the IP address of www.example.net using the dig command. Now here I can see that when the attack is not launched by the attacker I get the accurate information of www.example.net including its IP address, nameservers, and IP address but when the attacker launches the attack I get the spoofed information in the reply.

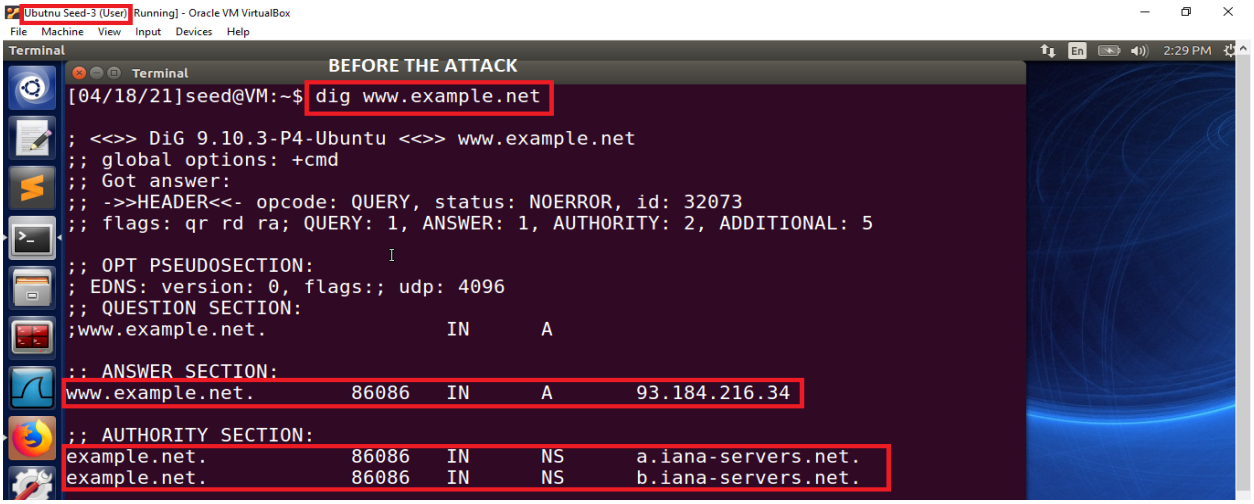

Figure 37: User: Dig response before the attack

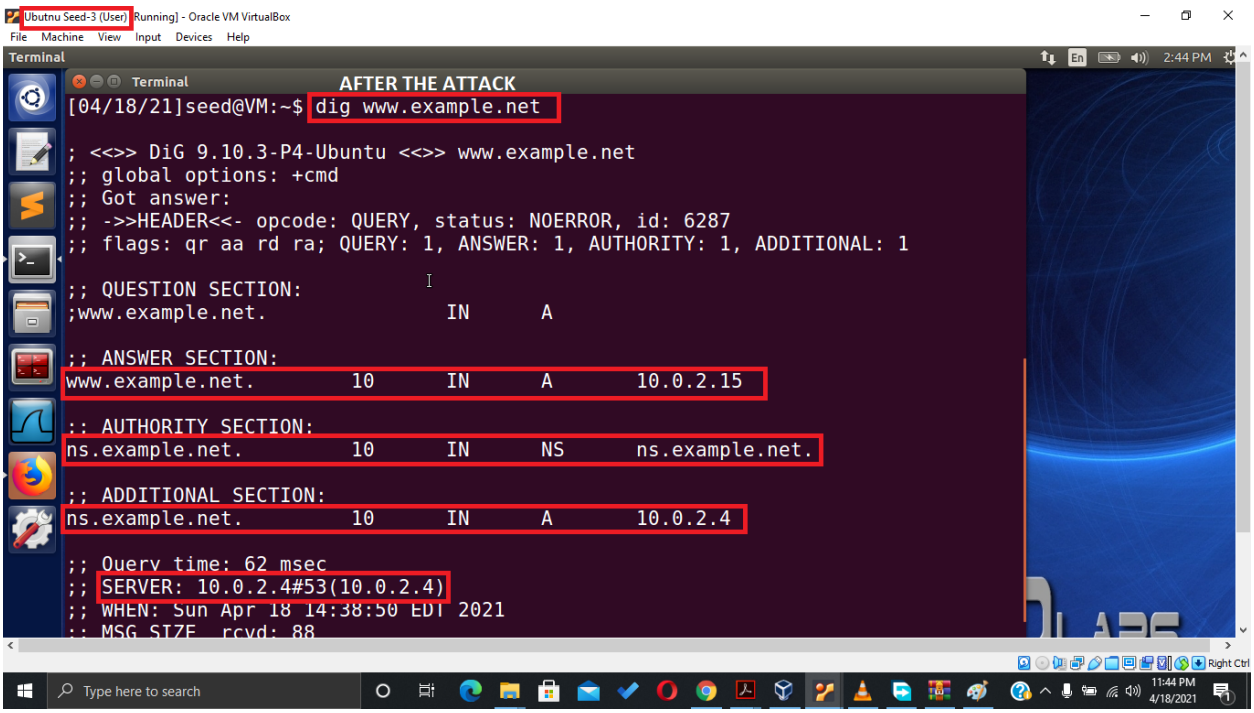

Figure 38: User: Dig response after the attack

6. Then I go back to the attacker machine and see the DNS responses of the user request.

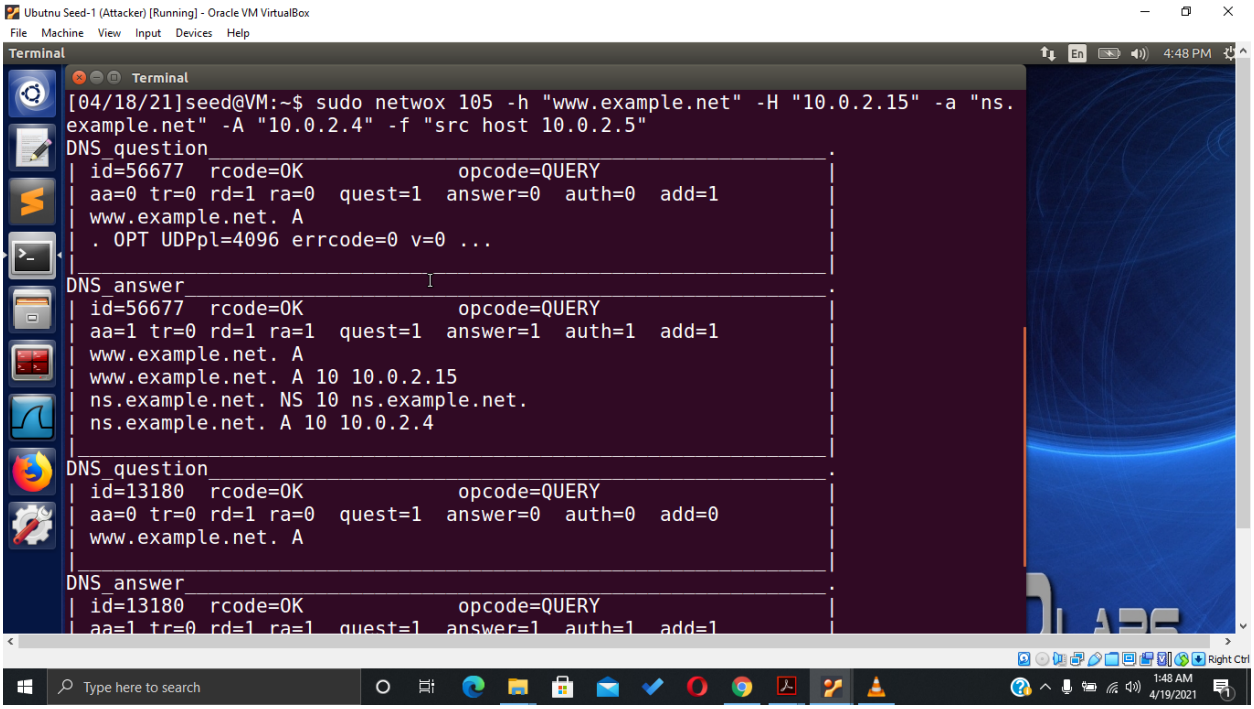

Figure 39: Attacker: DNS responses to the user request

Task-6 DNS Cache Poisoning

1. In this task instead of spoofing the user request I a going to spoof the DNS server now. For that, I again go to the attacker's machine and use the Netwox tool's 105 utility "Sniff and send DNS answers" to attack the DNS server.
2. To start the attack I open the terminal in the attacker machine and run the Netwox command "Netwox 105 -h www.example.net -H 10.0.2.15 -a ns.example.net -A 10.0.2.4 -T 600 -f "src host 10.0.2.4" -s raw ". In this command "-h" defines the hostname, "-H" defines the host IP, "-a" defines the authoritative nameserver, "-A" defines the nameserver IP, "-T" defines the time to live in seconds, "-f" defines the filter and -s defines the spoof initialization type.

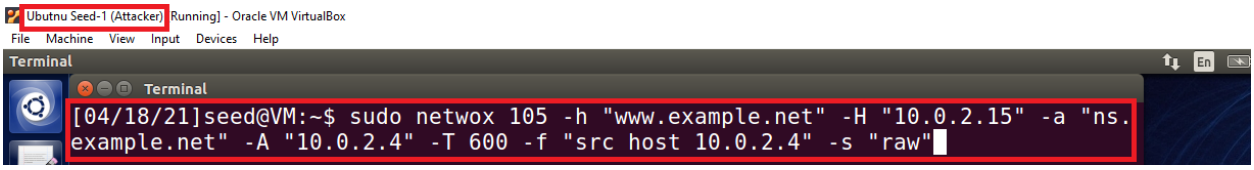

Figure 40: Attacker: Runs the Netwox command

3. Then that I go to the server machine and flush the DNS server cache.

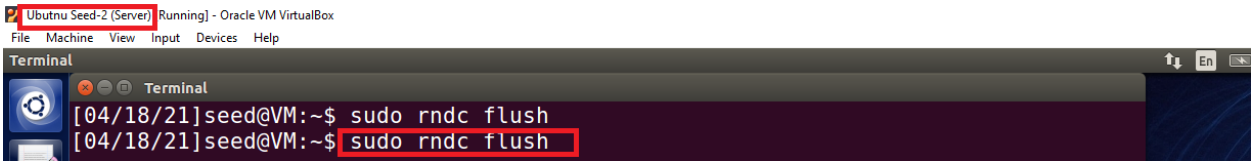

Figure 41: Server: Flush the DNS Server Cache

4. Then I go to the user machine, open the Wireshark and run it, then I open the terminal and ask the local DNS server for the IP address of www.example.net using the dig command. Now here I get the spoofed information in the reply.

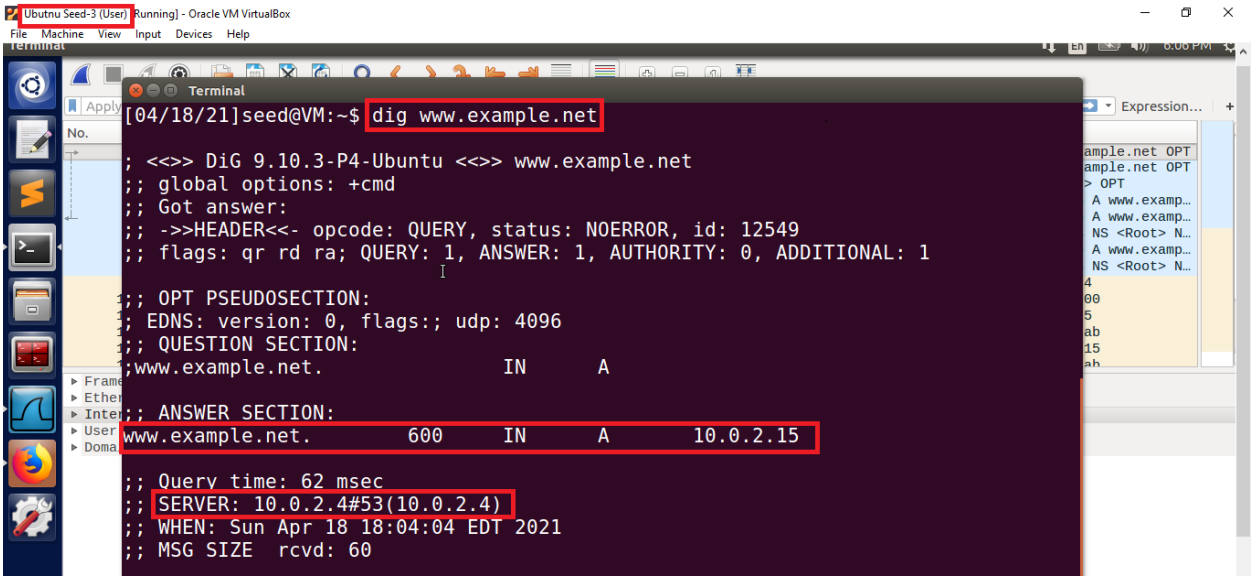

Figure 42: User: Dig response after the attack

5. Then I go to Wireshark to analyze the DNS traffic and found that on request of the user DNS server respond to the user with spoof information by the attacker.

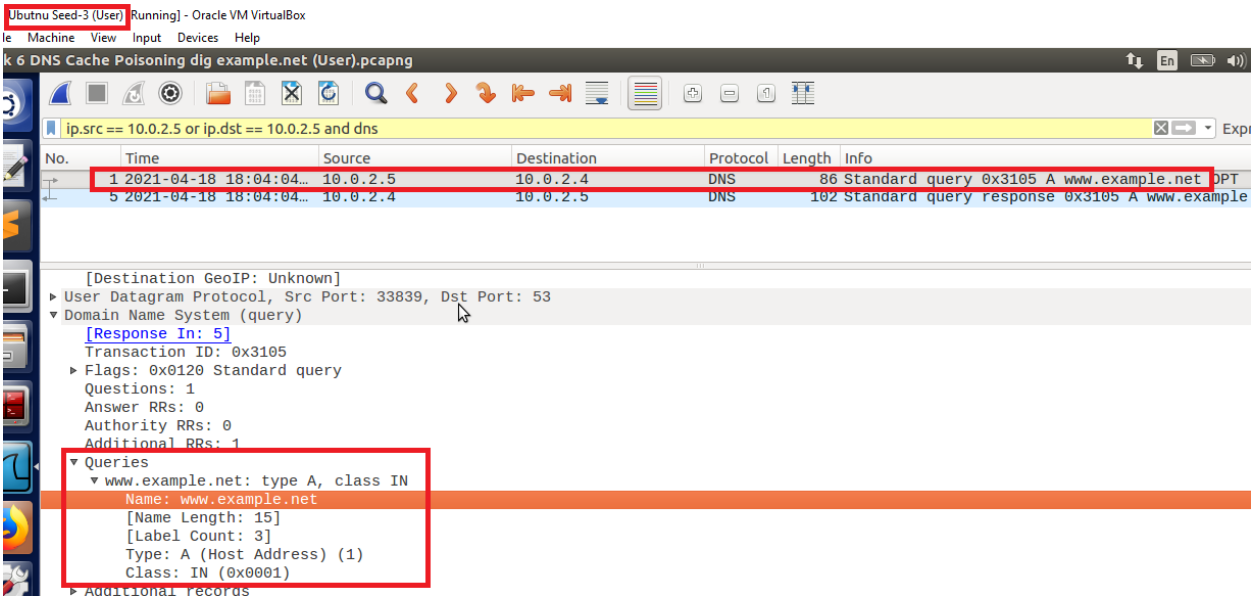

Figure 43: User: User request server for www.example.net IP and Nameserver

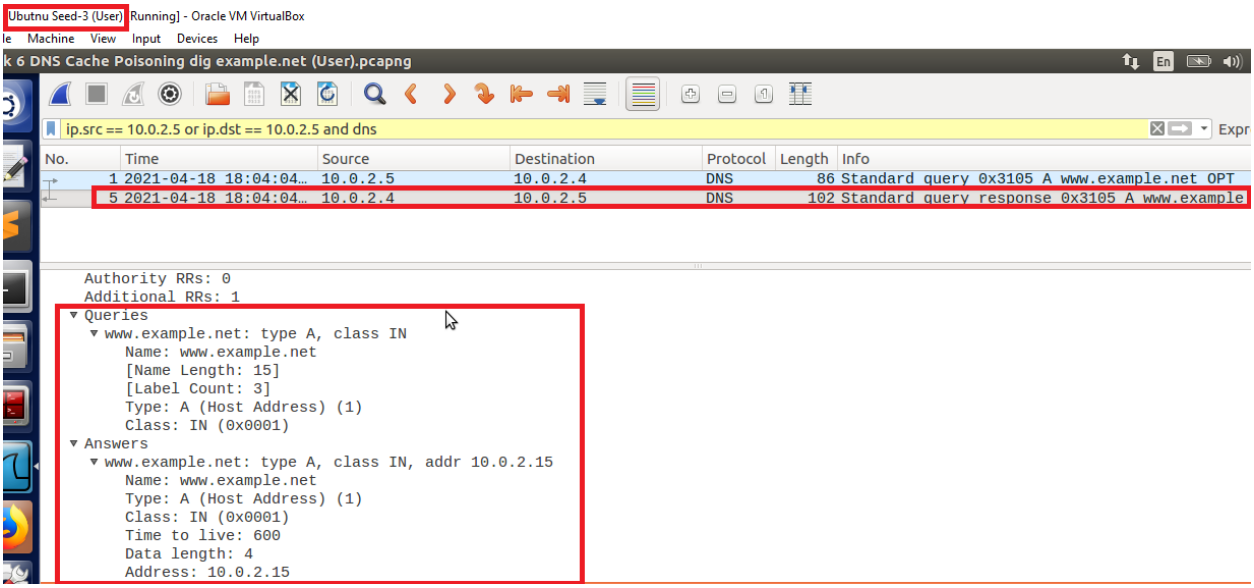

Figure 44: User: DNS server spoofed response to the user's query

6. Then I go back to the server machine and dump the cache in the dump.db file using the "rndc dumped -cache" command and then open the file using the "cat" command. Now here I found that the spoofed response by the attacker is also dumped in the DNS server's cache.

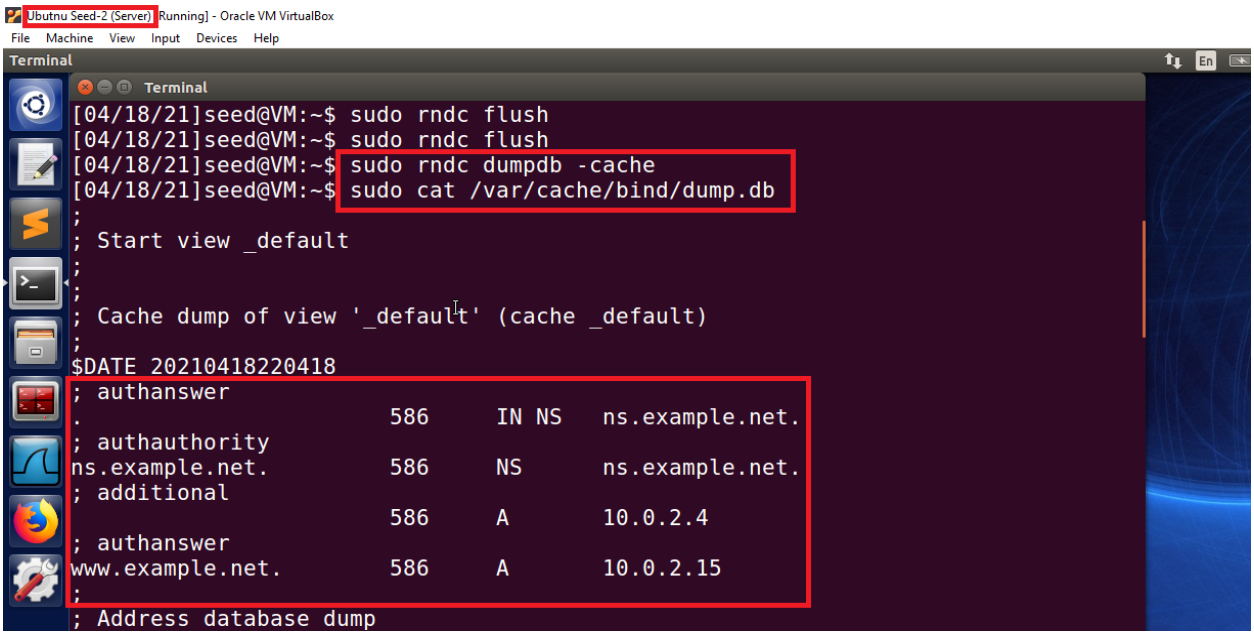

Figure 45: Server: Attackers spoofed response in DNS Cache

7. After that, I go back to the attack machine and see the DNS responses to the user request.

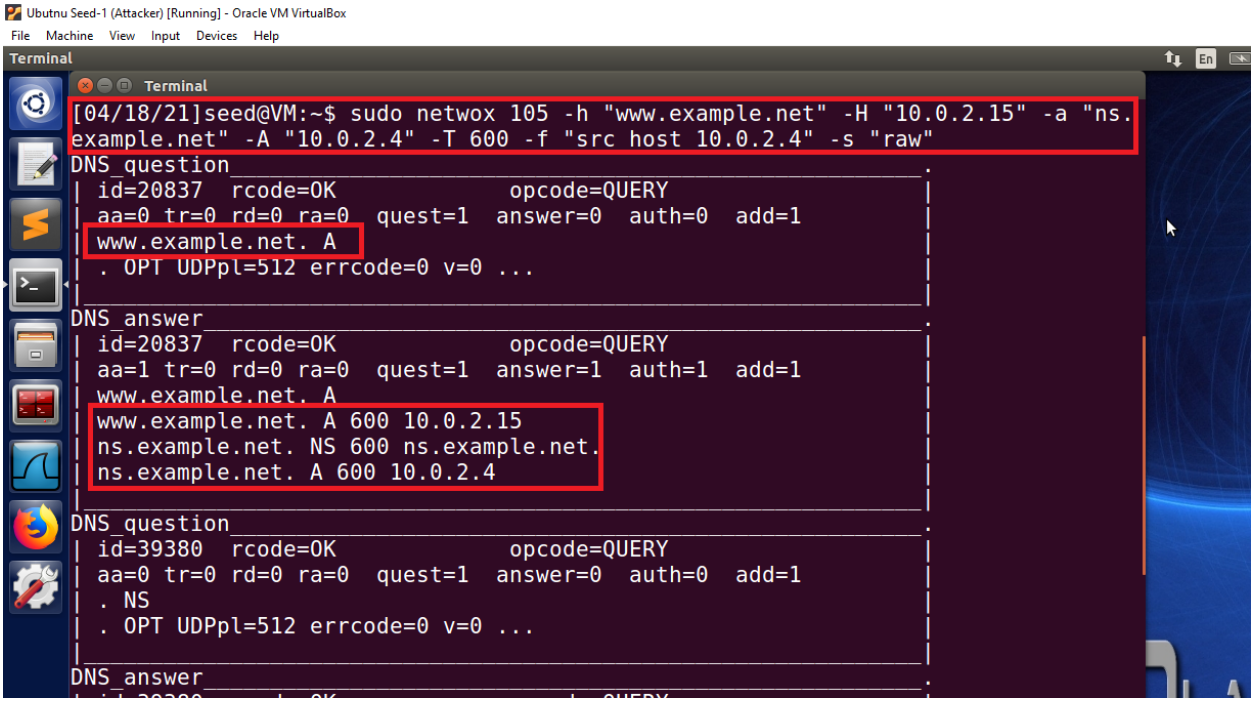

Figure 46: Attacker: DNS responses to the user request

Task-7 DNS Cache Poisoning: Targeting the Authority Section

1. In this task, I am going to attack the authority section of the DNS server replies.
2. For that first, I go to the attacker machine and then create the python file named "DNSCachePoisoningAttack.py".

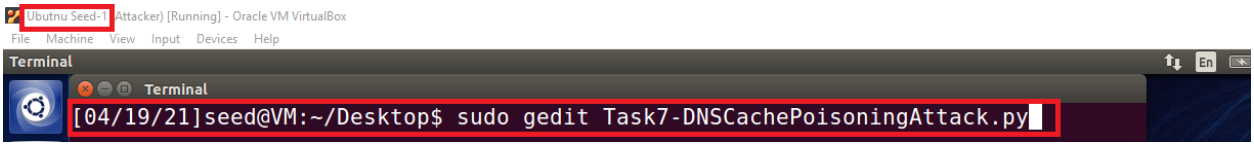

Figure 47: Attacker: Creating the python code file

3. Then add the code given in the guideline section of the lab manual. After that, first I change the IP address in the "answer" section to the attacker's IP address, and then in the "authority section nameserver section 1" I change the nameserver of www.example.net to atacker32.com. Then in the "Construct the DNS packet" section, I change the name server count value to "1" because in this task we require only one nameserver and then I change the additional section count to "0" because it is not required in this task.

Figure 48: Attacker: Python code used in this task

4. After that, I run the code.

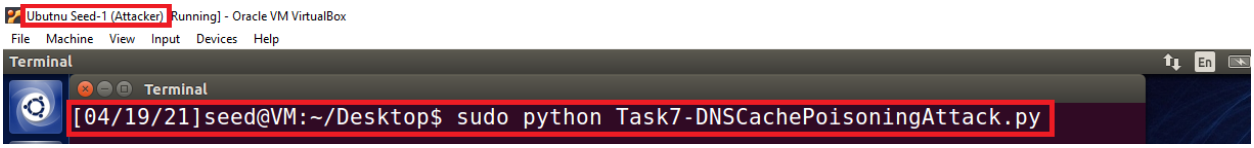

Figure 49: Attacker: Run the Python code

5. Then that I go to the server machine and flush the DNS server cache.

Figure 50: Server: Flush the DNS Server Cache

6. Then I go to the user machine, open the Wireshark and run it, then I open the terminal and ask the local DNS server for the IP address of www.example.net using the dig command. Now here I get the spoofed information in the "answer and authority section".

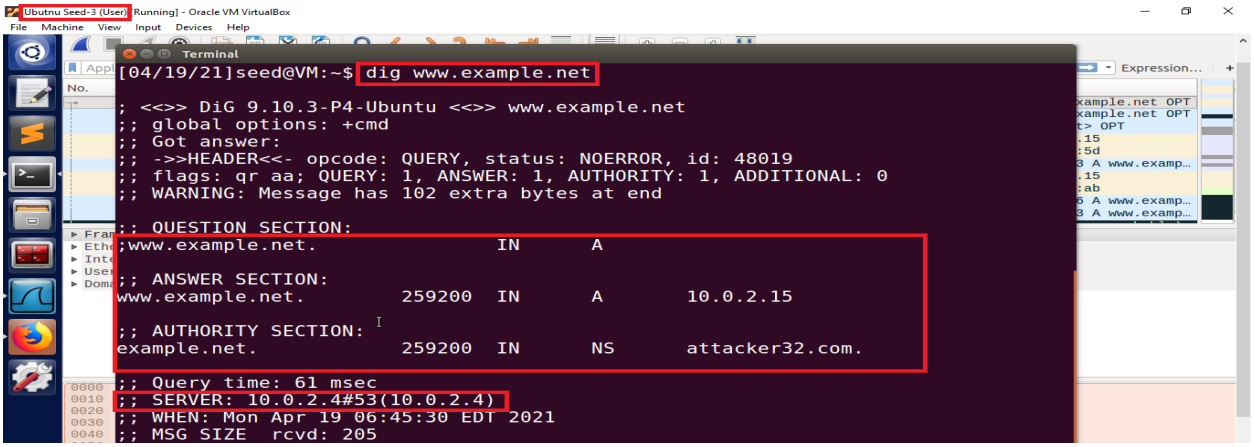

Figure 51: User: Dig response after the attack

7. Then I go to Wireshark to analyze the DNS traffic and found that on request of the user DNS server respond to the user with spoof information by the attacker.

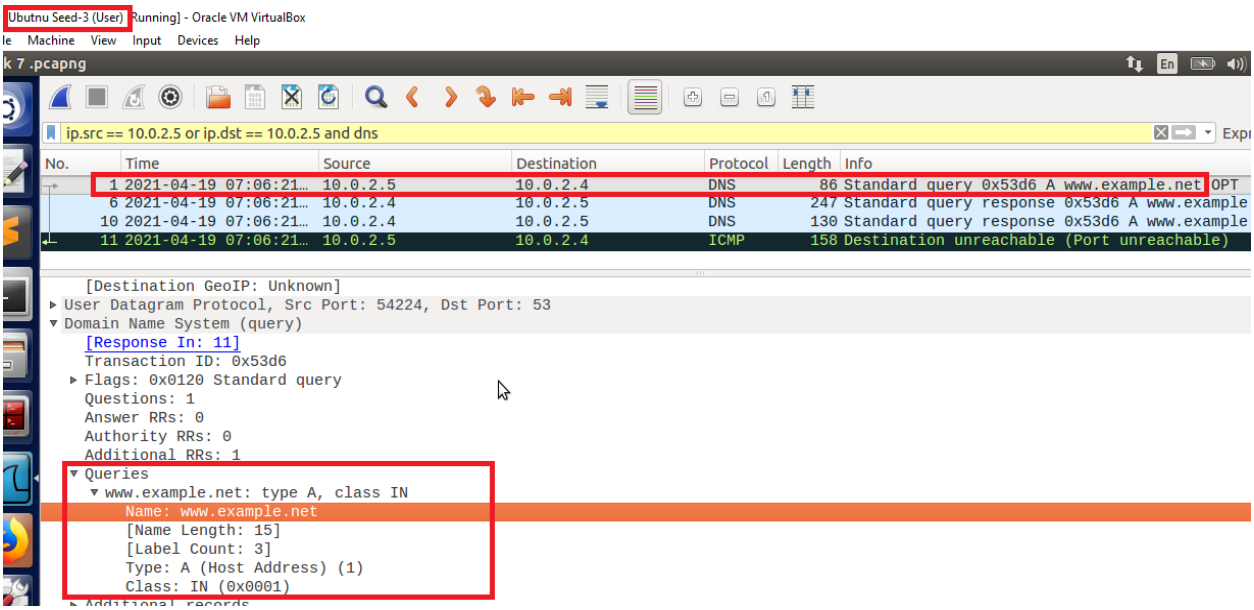

Figure 52: User: User request server for www.example.net IP and Nameserver

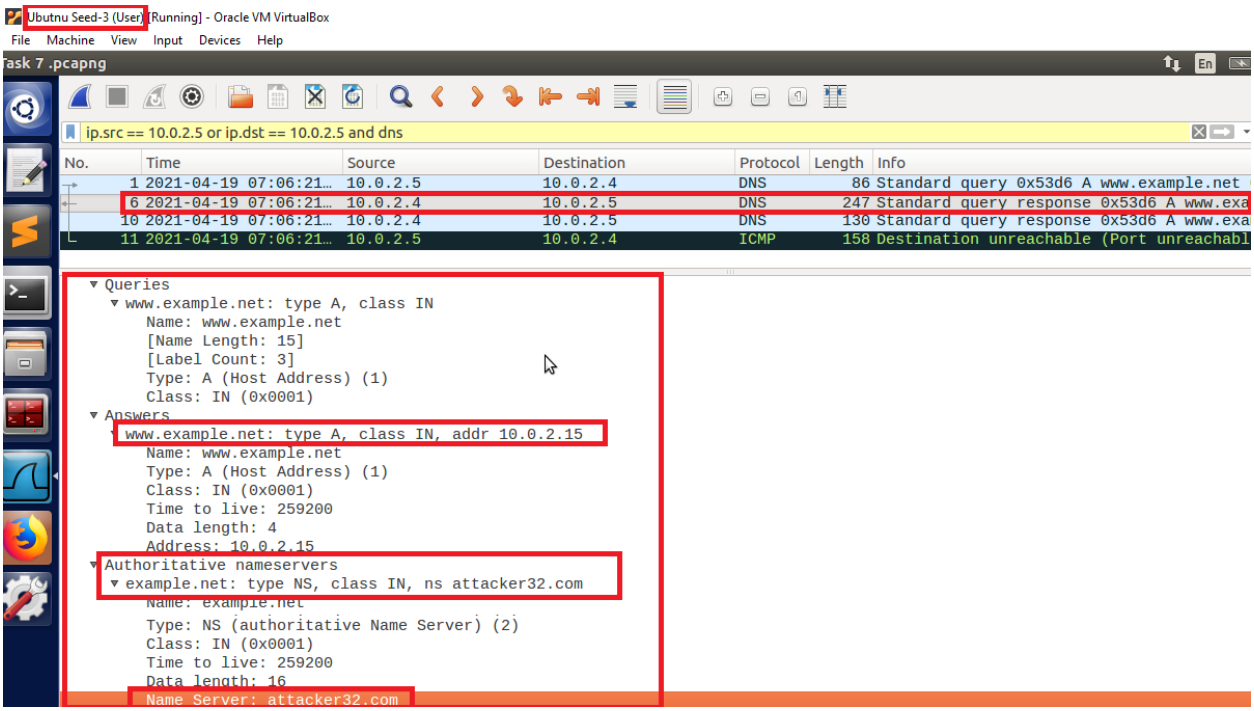

Figure 53: User: DNS server spoofed response to the user's query

8. Then I go back to the server machine and dump the cache in the dump.db file using the "rndc dumpdb -cache" command and then open the file using the "cat" command. Now here I found that the spoofed response by the attacker is also dumped in the DNS server's cache.

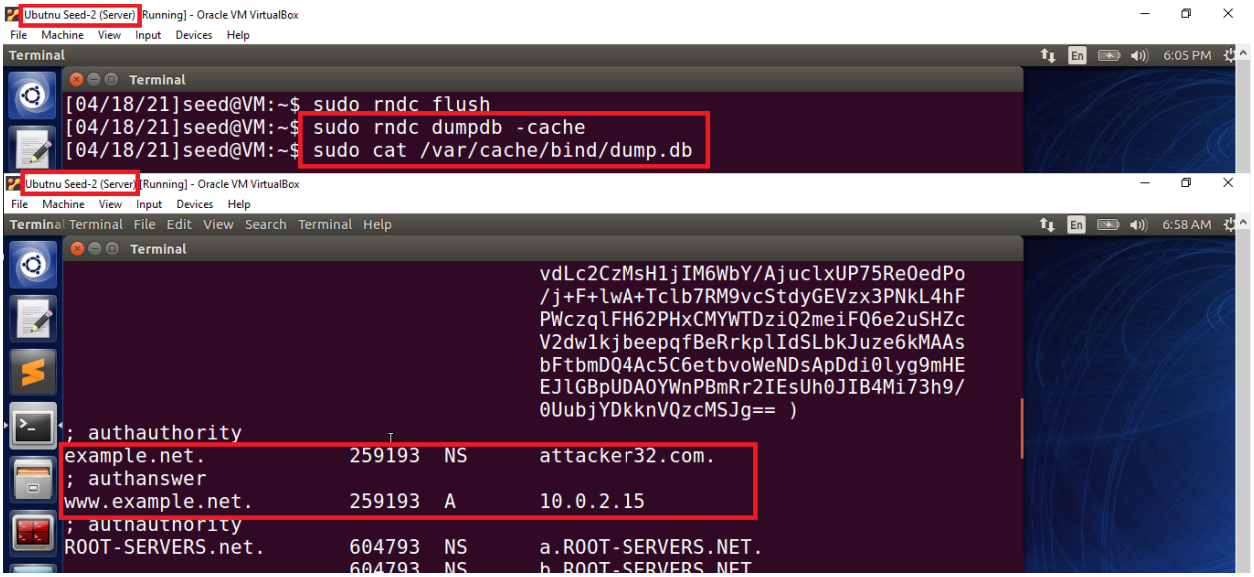

Figure 54: Attacker poisoned the DNS Cache successfully

Task-8 Targeting Another Domain

1. In the previous task, we are targeting only one domain but now in this task, we are targeting another domain by following the same steps.
2. First, I go to the attacker machine and then create the python file named "TargetingAontherDomain.py".

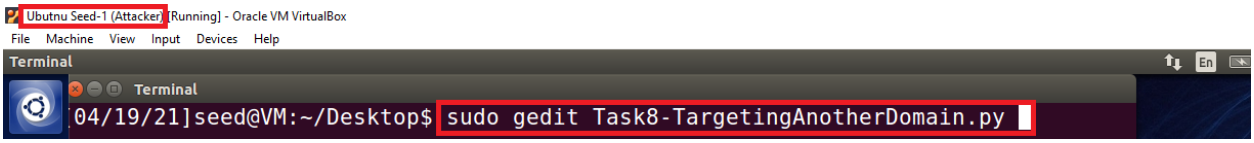

Figure 55: Attacker: Creating the python code file

3. Then add the code given in the guideline section of the lab manual. After that, first I change the IP address in the "answer" section to the attacker's IP address, and then in the "authority section nameserver section 2" I change the hostname to www.google.com and then change the nameserver to atacker32.com. Then in the "Construct the DNS packet" section, I change the name server count value to "2" because in this task we require both nameservers and then I change the additional section count to "0" because it is not required in this task.

Figure 56: Attacker: Python code used in this task

4. After that, I run the code.

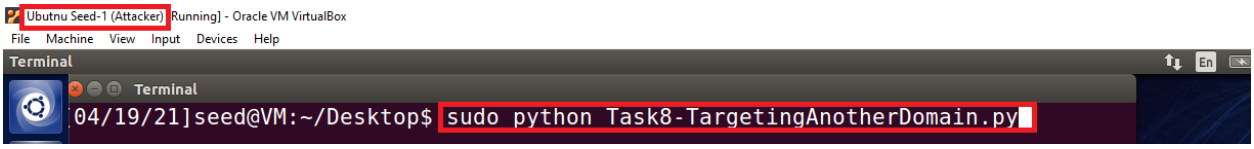

Figure 57: Attacker: Run the Python code

5. Then that I go to the server machine and flush the DNS server cache.

Figure 58: Server: Flush the DNS Server Cache

6. Then I go to the user machine, open the Wireshark and run it, then I open the terminal and ask the local DNS server for the IP address of www.example.net using the dig command. Now here I get the spoofed information of www.example.com and also the spoofed nameserver of www.google.com in the "answer and authority section".

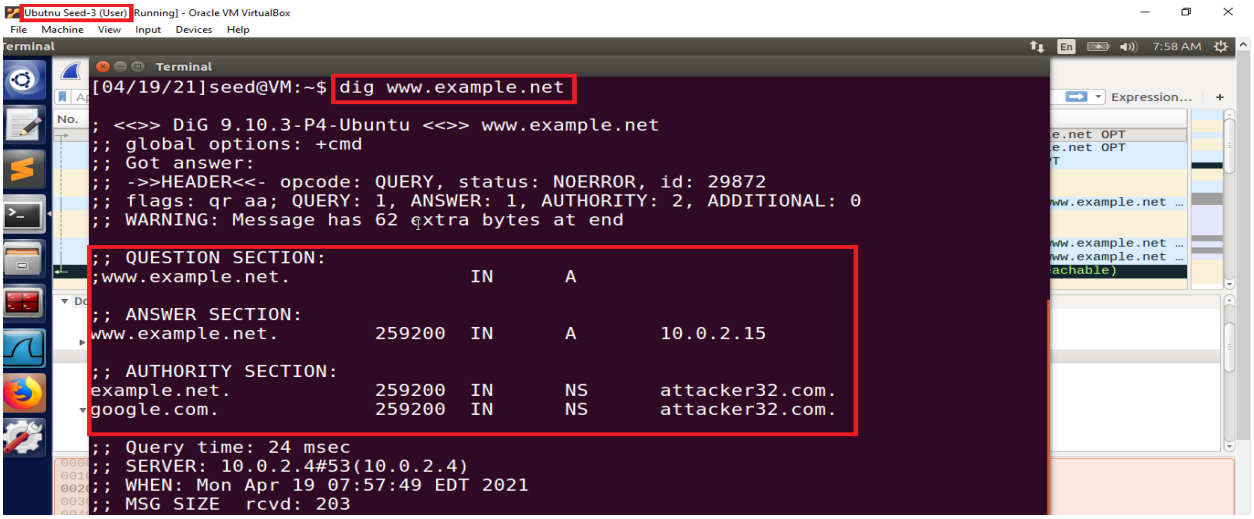

Figure 59: User: Dig response after the attack

7. Then I go to Wireshark to analyze the DNS traffic and found that on request of the user DNS server respond to the user with spoof information by the attacker. Here also we get the spoofed information of www.example.com and also the spoofed nameserver of www.google.com and also the details of the authority section.

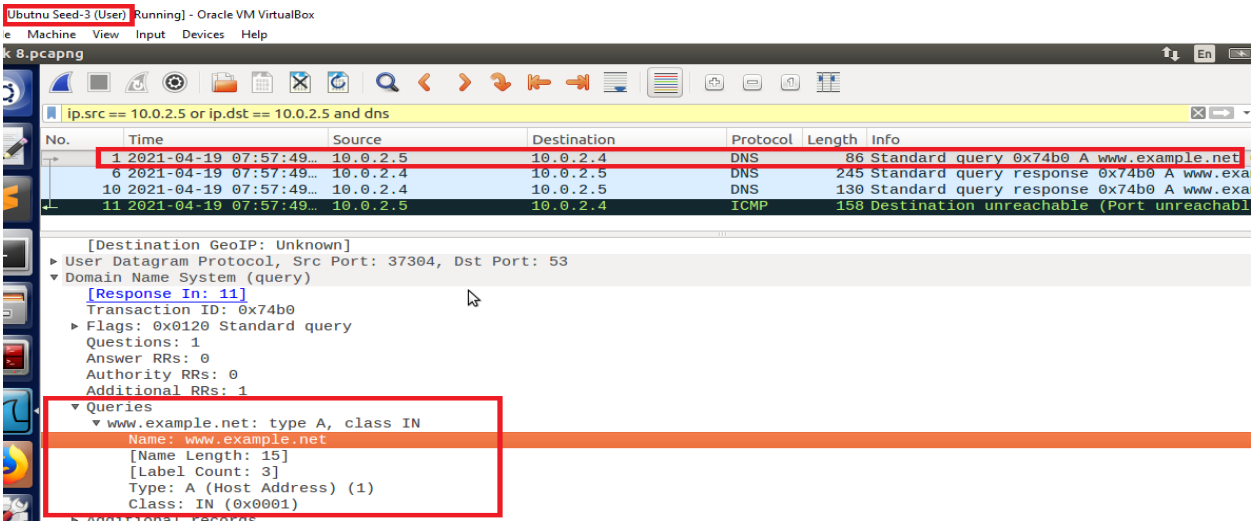

Figure 60: User: User request server for www.example.net IP and Nameserver

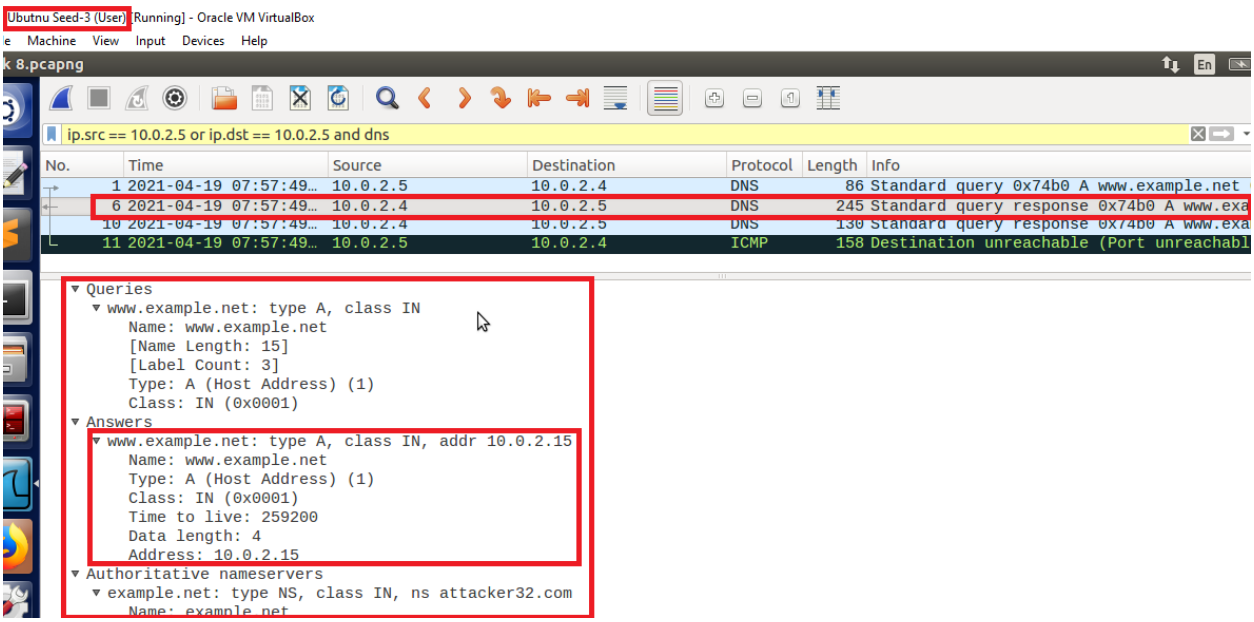

Figure 61: User: DNS server spoofed response to the user's query

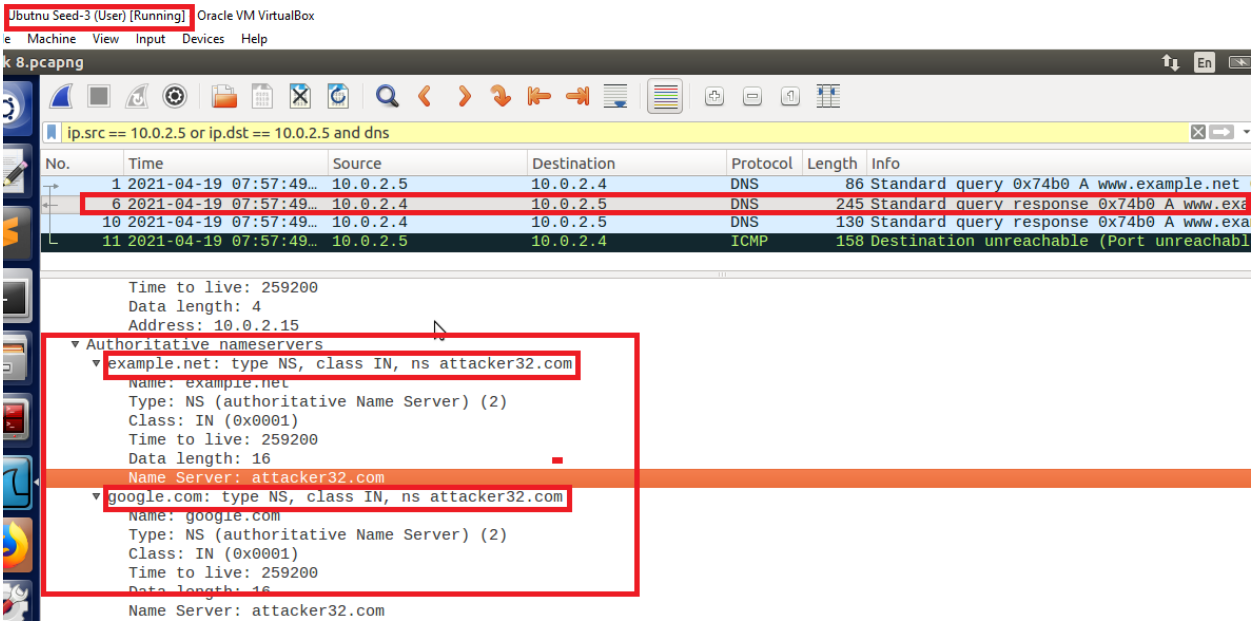

Figure 62: User: Authoritative Name server in the response

Task-9 Targeting the Additional Section

1. Till now we are only targeting the domain and authority section of the DNS. Now we are also going to target the additional section of the DNS. It provides the information of the IP addresses of the hostnames that are appearing in the authority section.

2. First, I go to the attacker machine and then create the python file named "TargetingtheAdditionalSection.py".

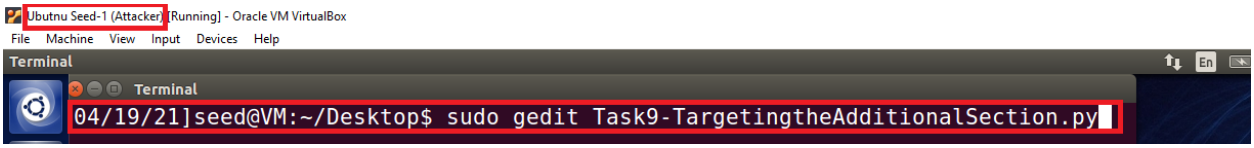

Figure 63: Attacker: Creating the python code file

3. Then add the code given in the guideline section of the lab manual. After that, first I change the IP address in the "answer" section to the attacker's IP address, and then in the "authority section nameserver section 1" I change the nameserver of www.example.net to atacker32.com and then in the "authority section nameserver section 2" I change the nameserver of www.example.net to ns.example.net. Then in the "Additional Section" I change the IP of the additional section 1 to "1.2.3.4" and hostname to atacker32.com and then in the additional section 2 I again change the IP to "5.6.7.8" and hostname to ns.example.net. After that, I add another entry in the additional section with the hostname www.facebook.com and IP "3.4.5.6". Then in the "Construct the DNS packet" section, I change the name server count value to "2" because in this task we require both nameservers and then I change the additional section count to "3" because there are three entries in this section and we have to use all three entries in this task.

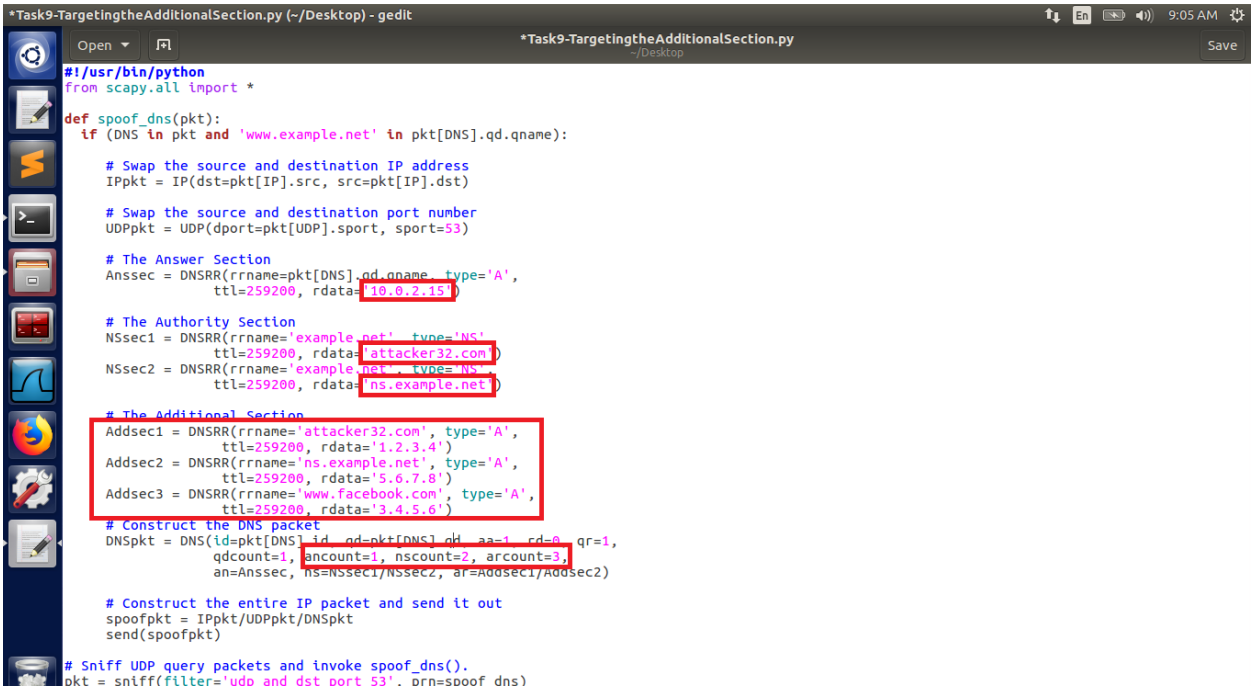

Figure 64: Attacker: Python code used in this task

4. After that, I run the code.

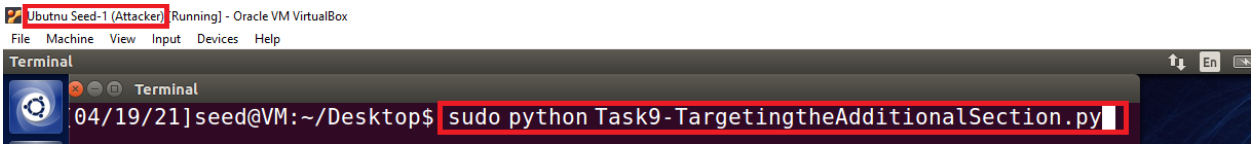

Figure 65: Attacker: Run the Python code

5. Then that I go to the server machine and flush the DNS server cache.

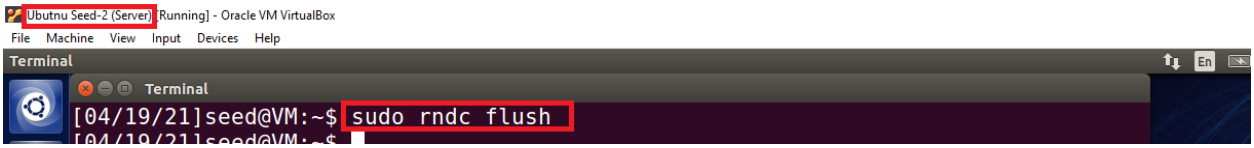

Figure 66: Server: Flush the DNS Server Cache

6. Then I go to the user machine, open the Wireshark and run it, then I open the terminal and ask the local DNS server for the IP address of www.example.net using the dig command. Now here I get the spoofed information of www.example.com in the "answer, authority and additional section".

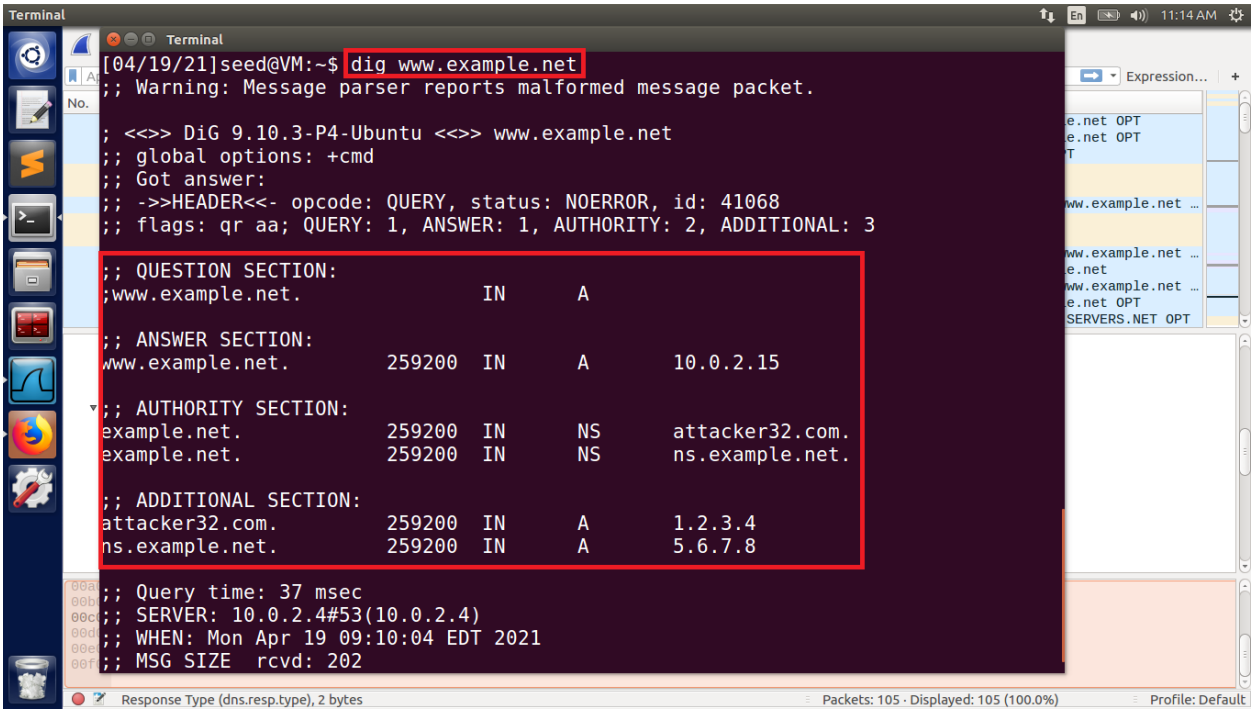

Figure 67: User: Dig response after the attack

7. Then I go to Wireshark to analyze the DNS traffic and found that on request of the user DNS server respond to the user with spoof information by the attacker. Here also we get the spoofed information of www.example.com and also the information of additional and authority section.

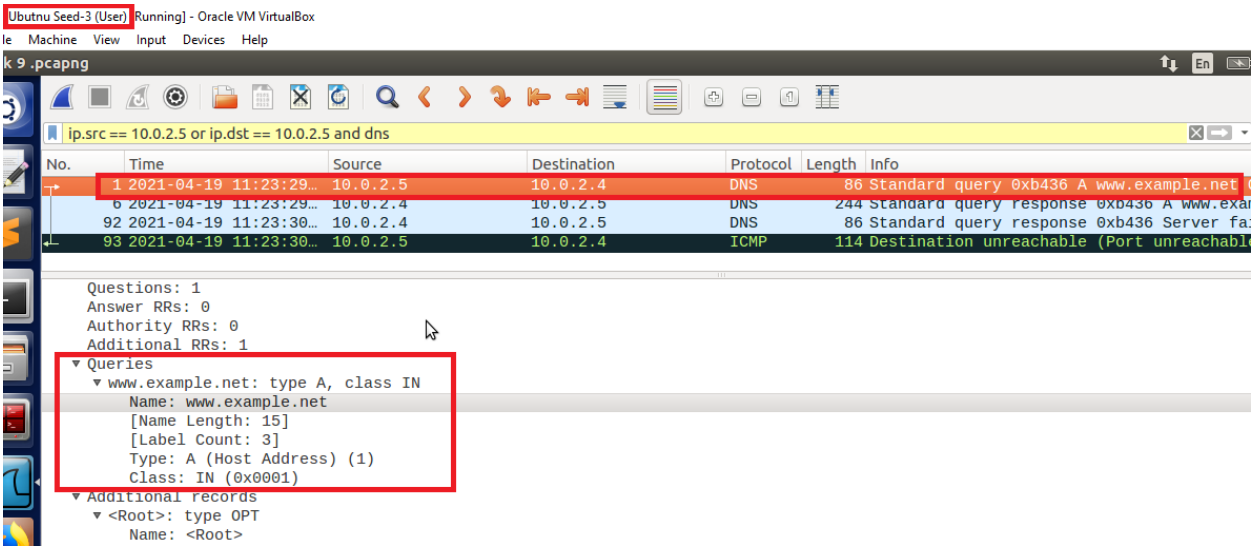

Figure 68: User: User request server for www.example.net IP and Nameserver

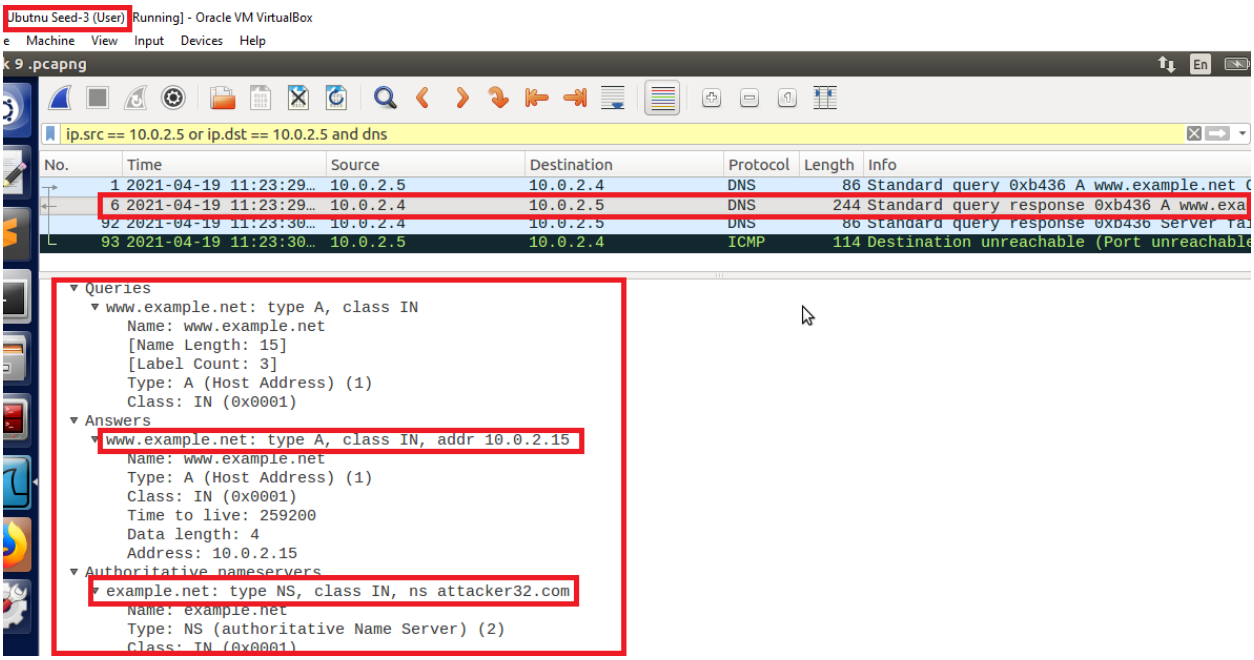

Figure 69: User: DNS server spoofed response to the user's query

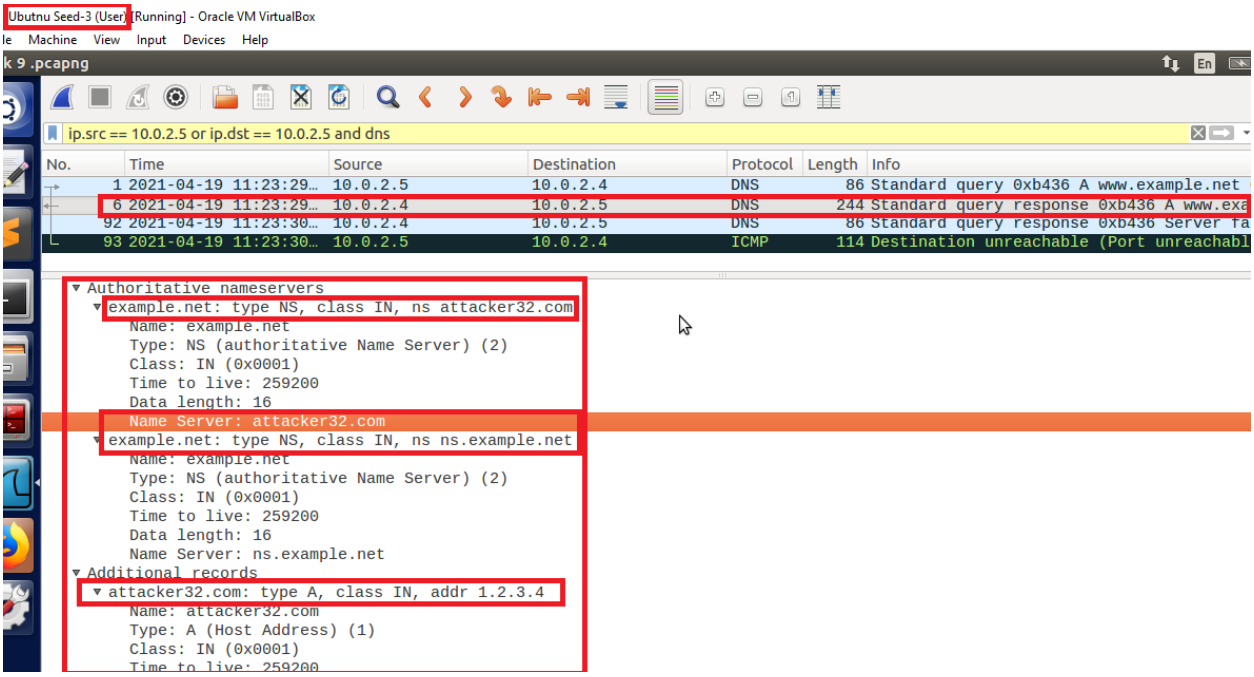

Figure 70: User: Authoritative Name server in the response

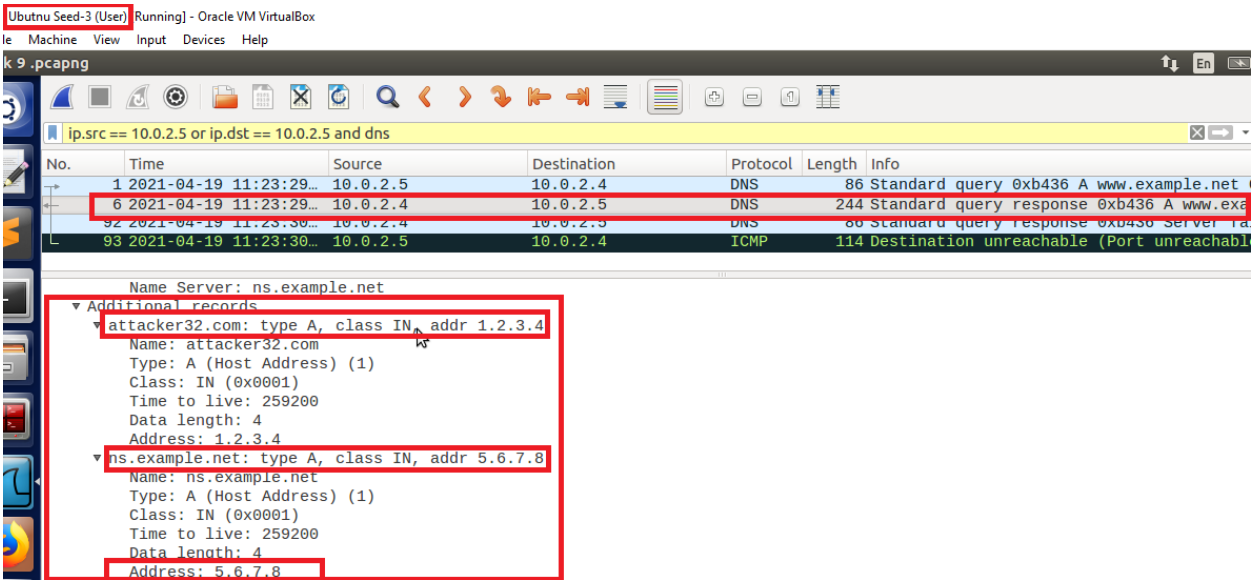

Figure 71: User: Additional section in the response