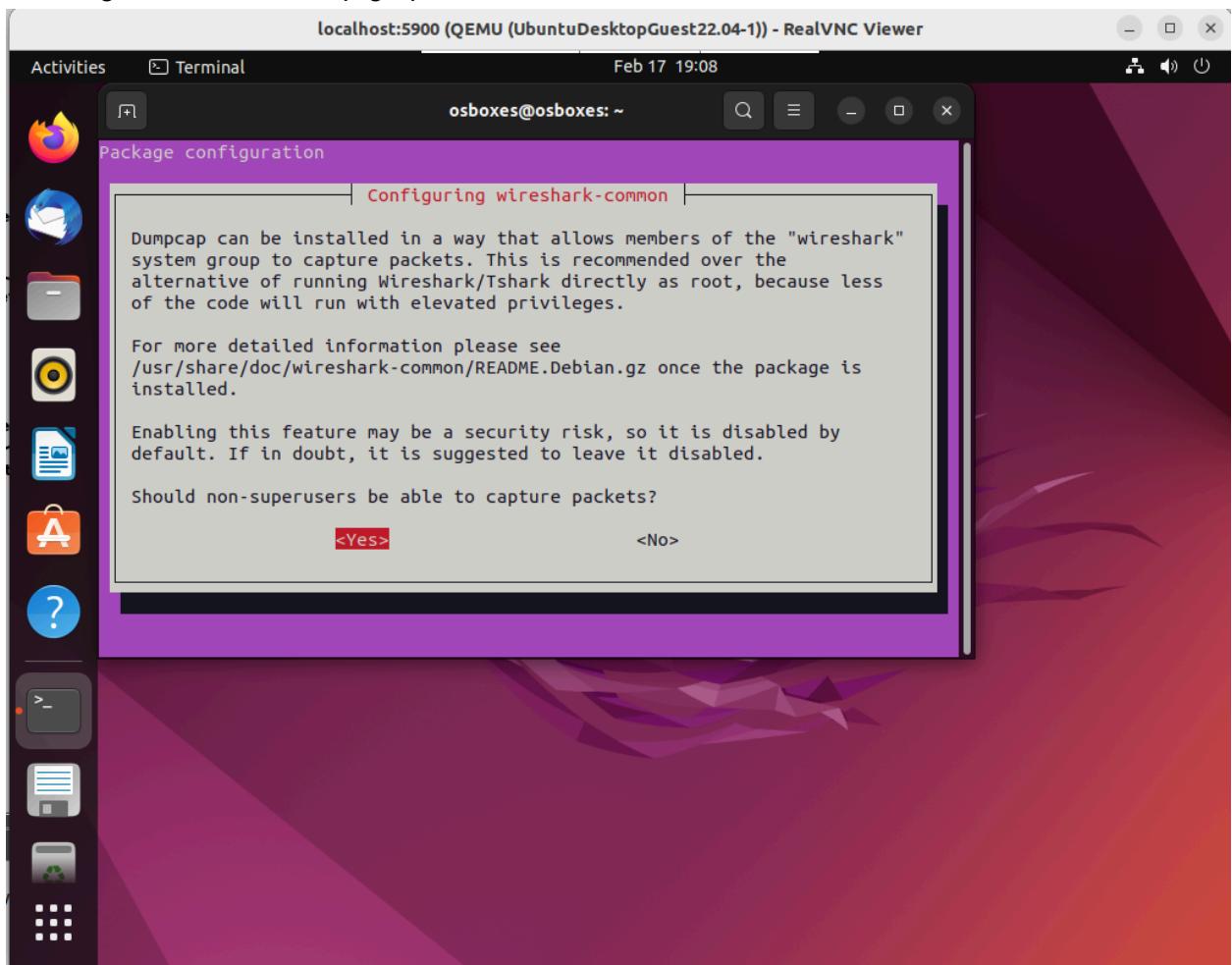


## Installing wireshark ettercap-graphical

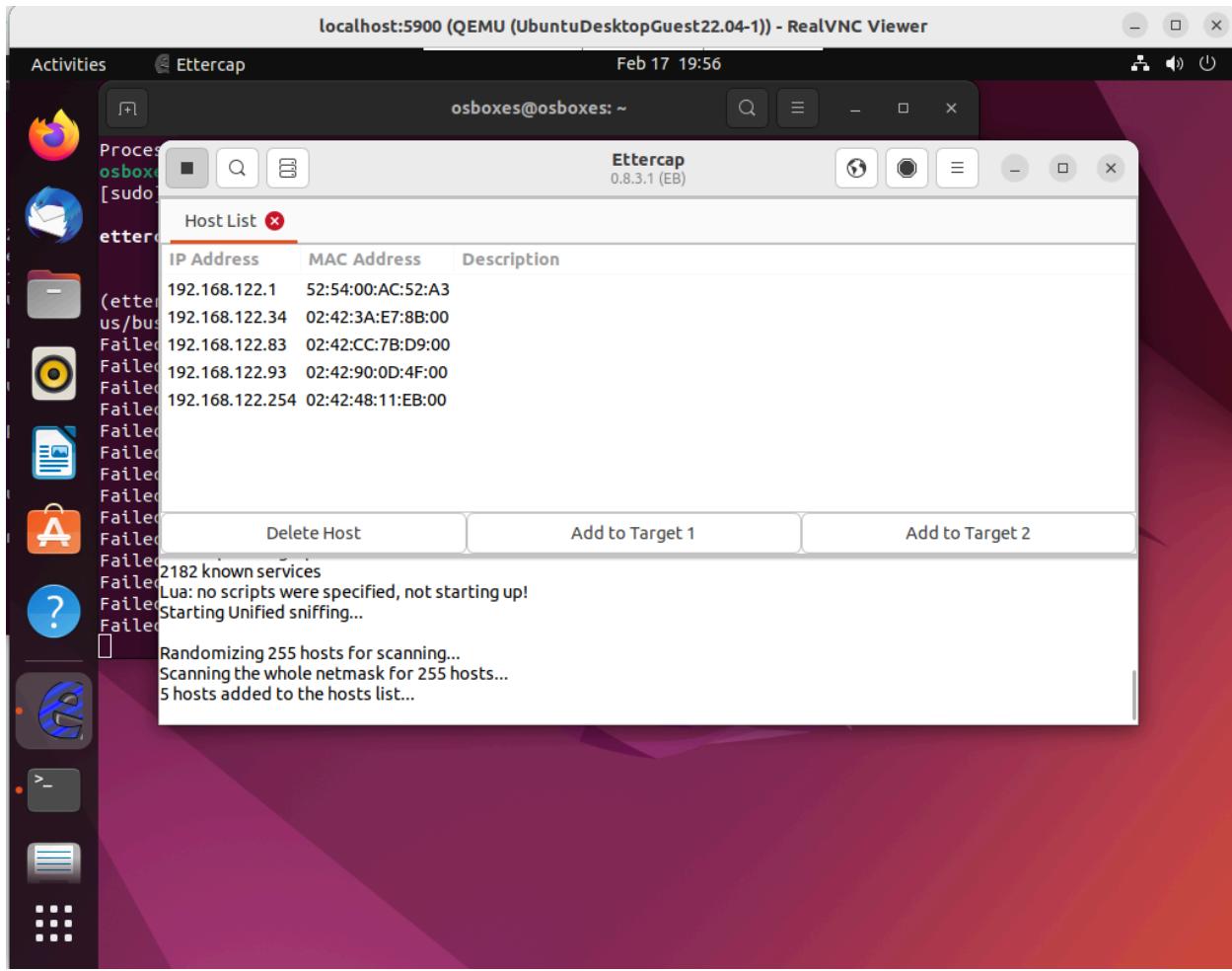


1. What are the IP addresses of LinuxHost1 and LinuxHost2?

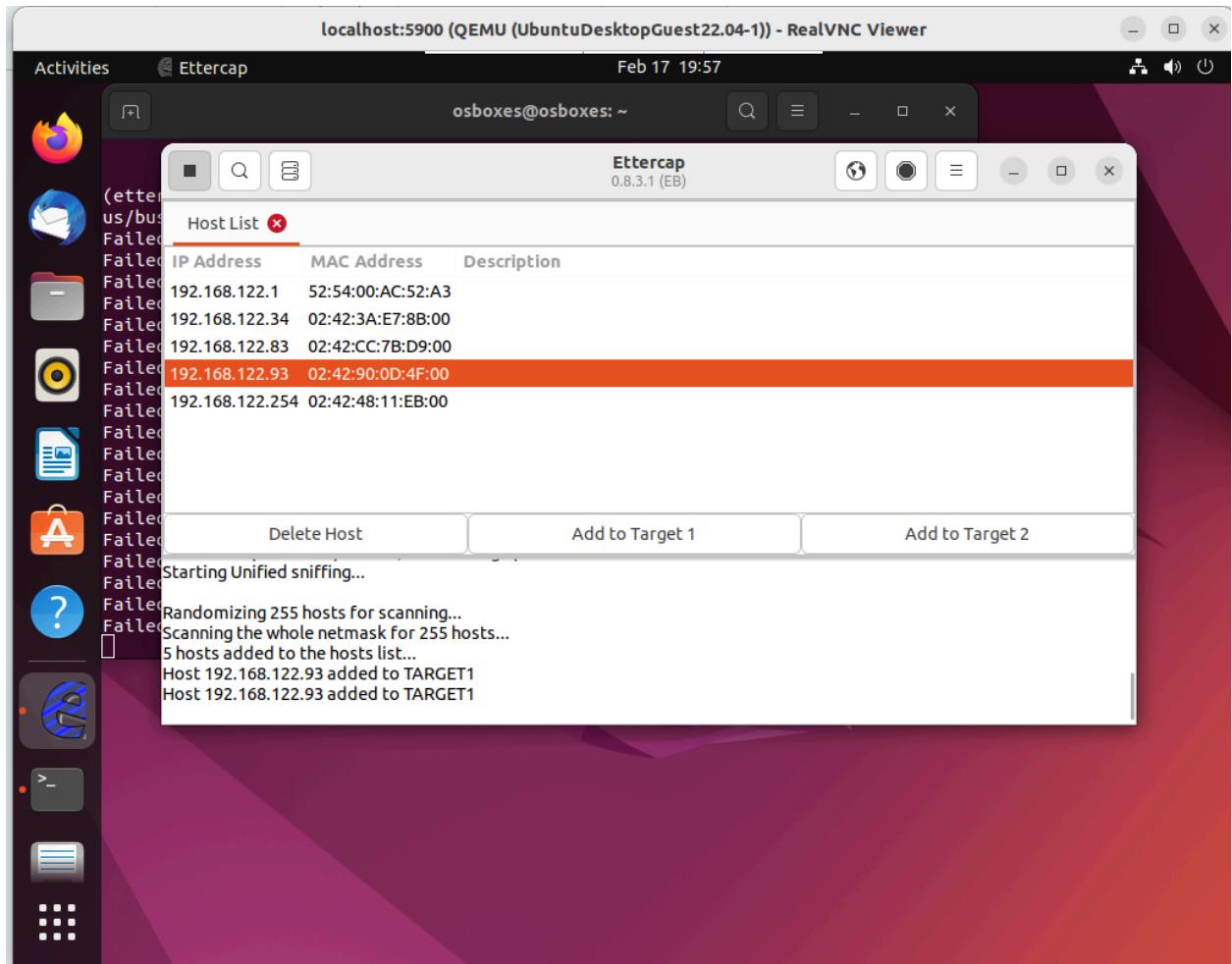
LinuxHost1 : 192.168.122.93

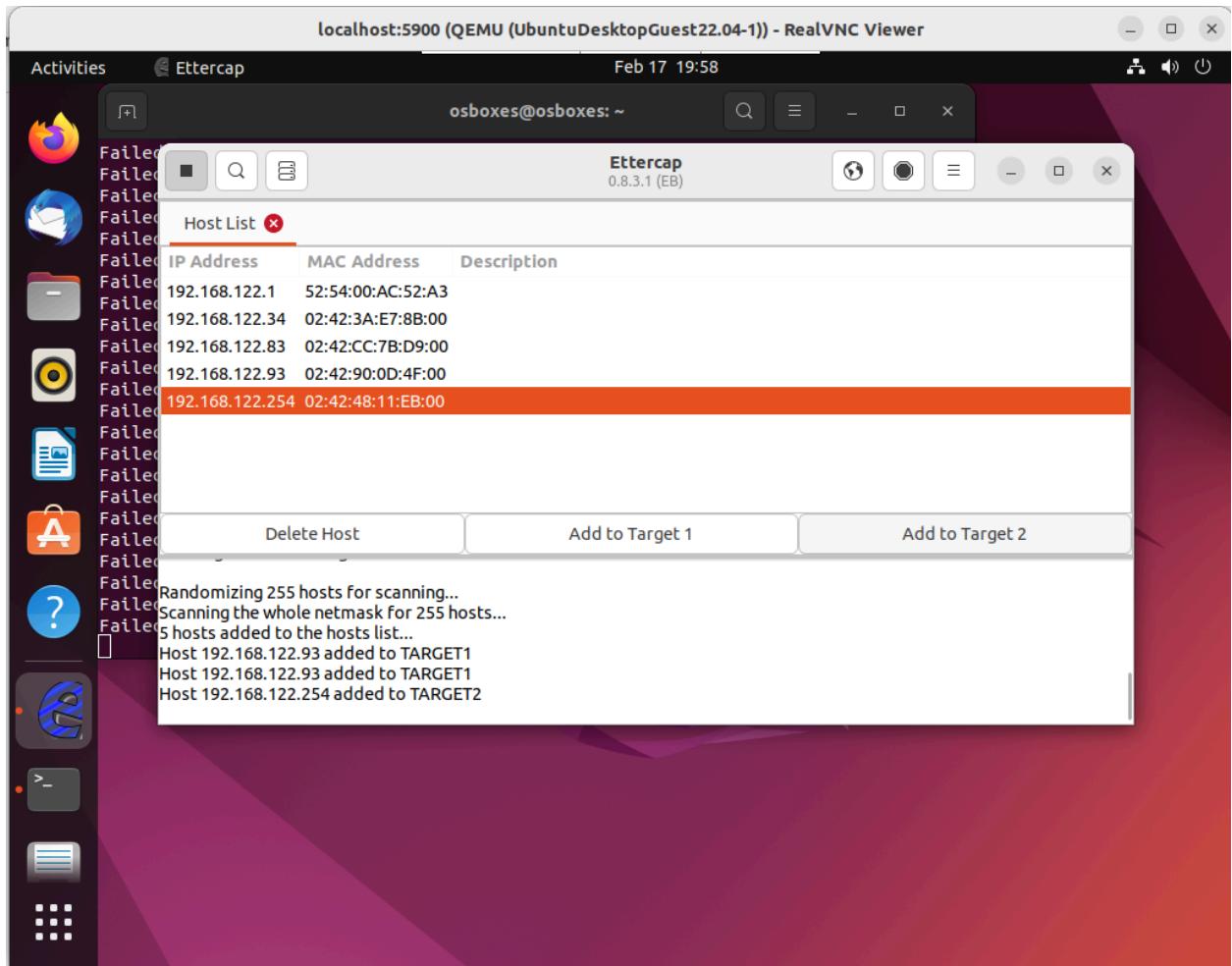
LinuxHost2 : 192.168.122.254

2. Next, we will use ettercap to conduct a MAC spoofing attack such that LinuxHost1 is deceived into believing that the MAC associated with the IP of LinuxHost1 is the MAC of Ubuntu Desktop and to deceive that LinuxHost2 that the IP associated with the MAC of LinuxHost1 is that of the Ubuntu Desktop.



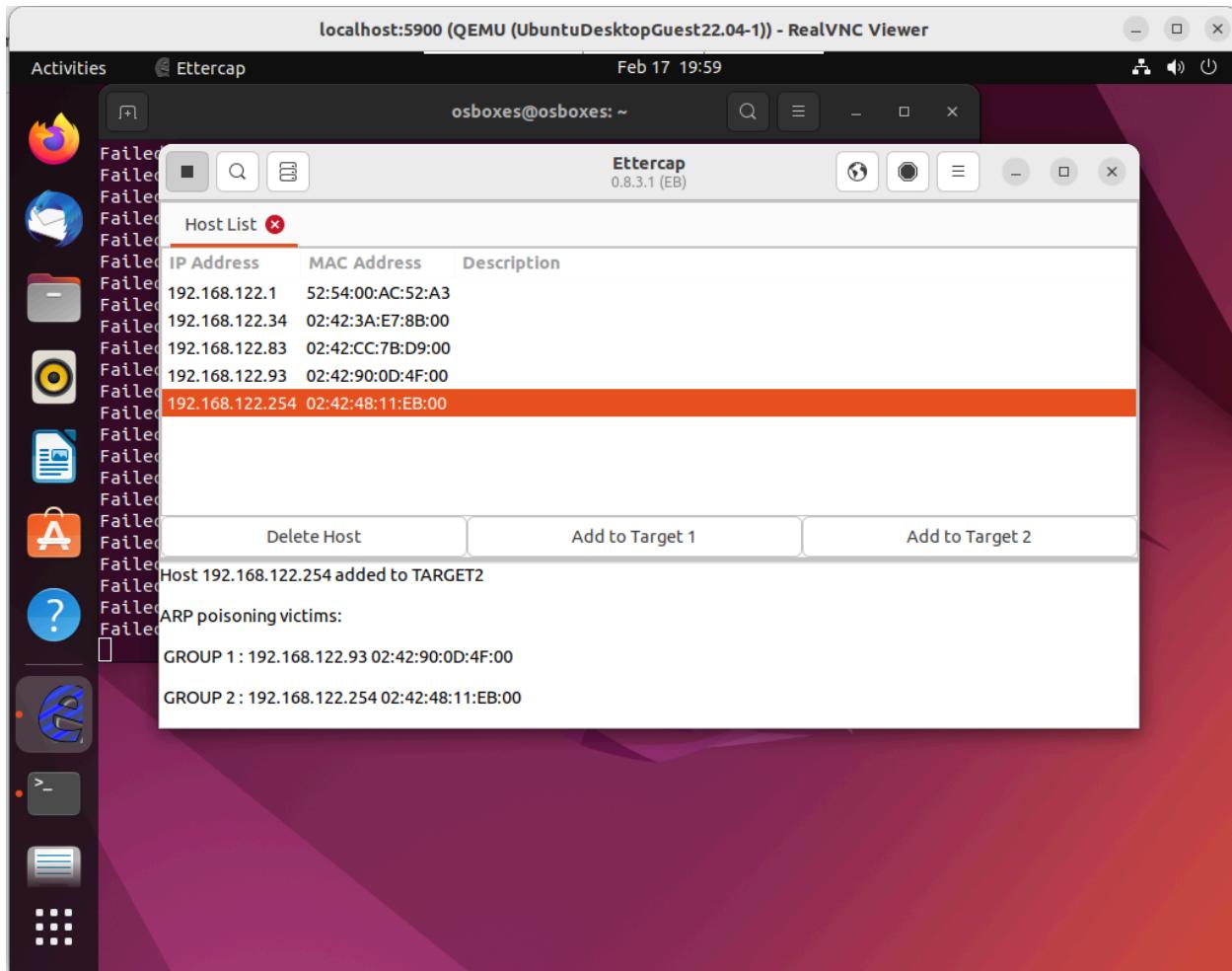
3. From the resulting list that appears, click on the IP and MAC of LinuxHost1 and click on "Add to Target 1" and then select LinuxHost2 and select "Add to Target 2". These will be the systems on which we will use ARP poisoning to achieve a Man-in-the-Middle attack. Please include screenshots of all of your screens.



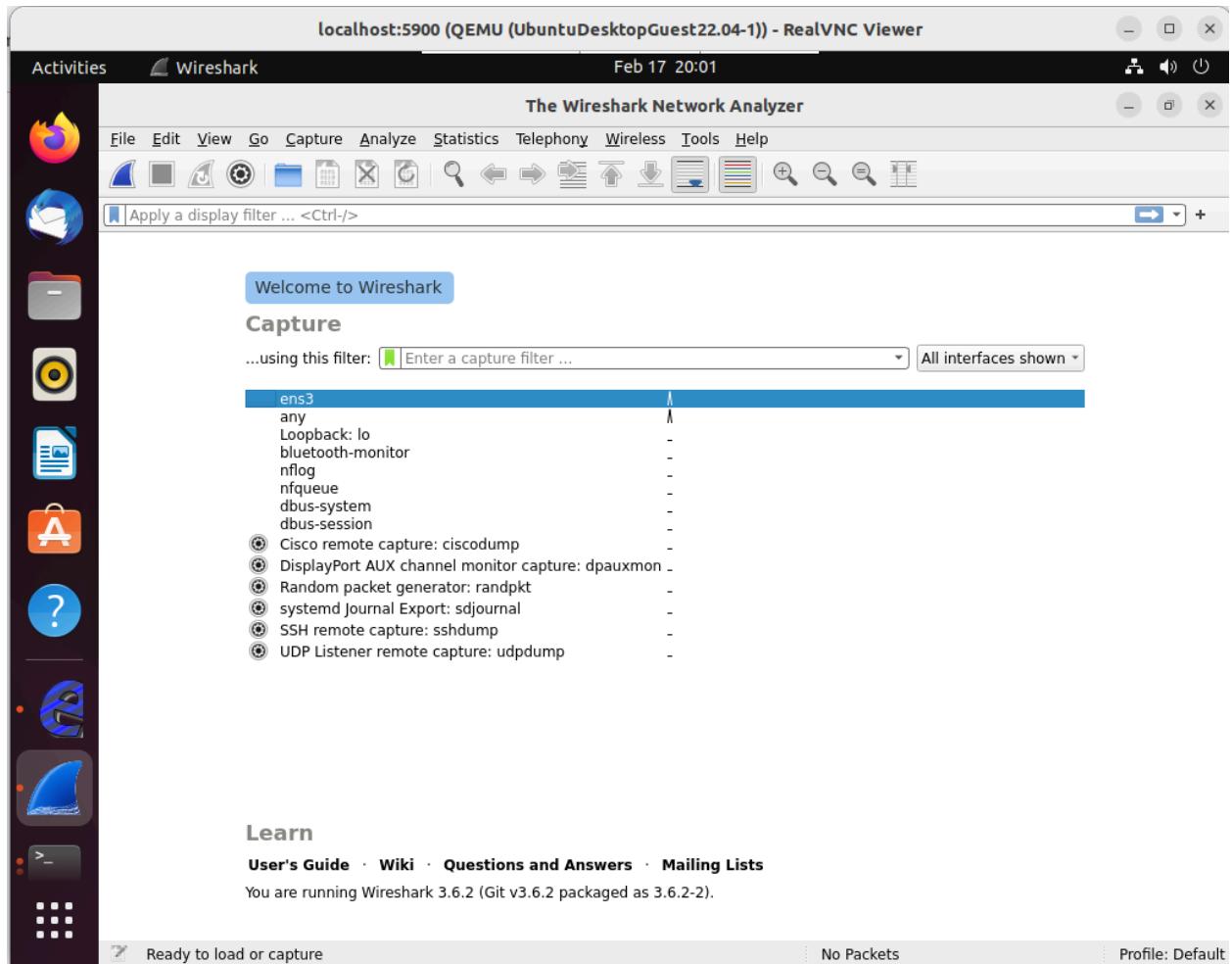


Adding LinuxHost1 and LinuxHost2 to their respective targets.

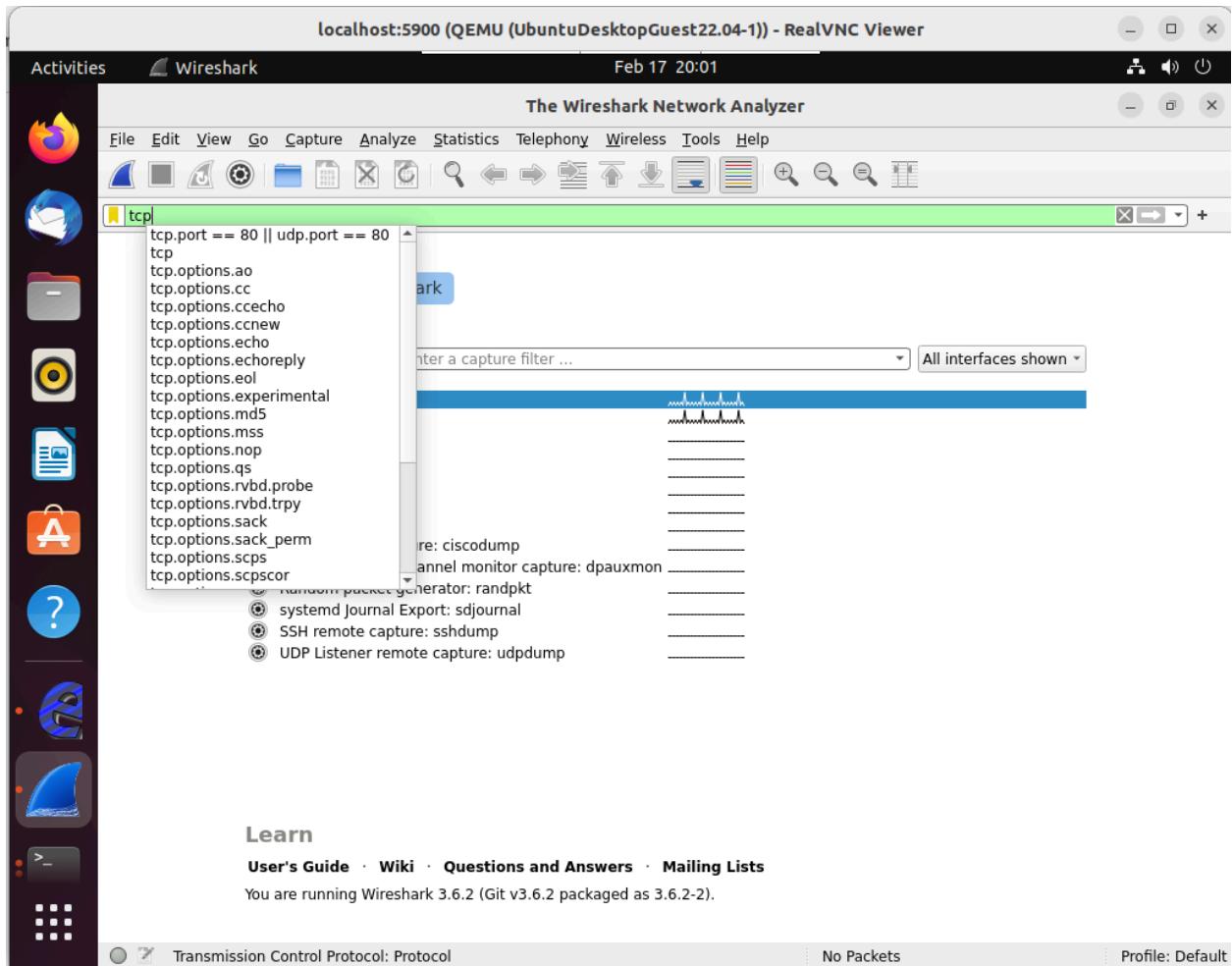
4. From the MITM attack menu, select "ARP Poisoning":



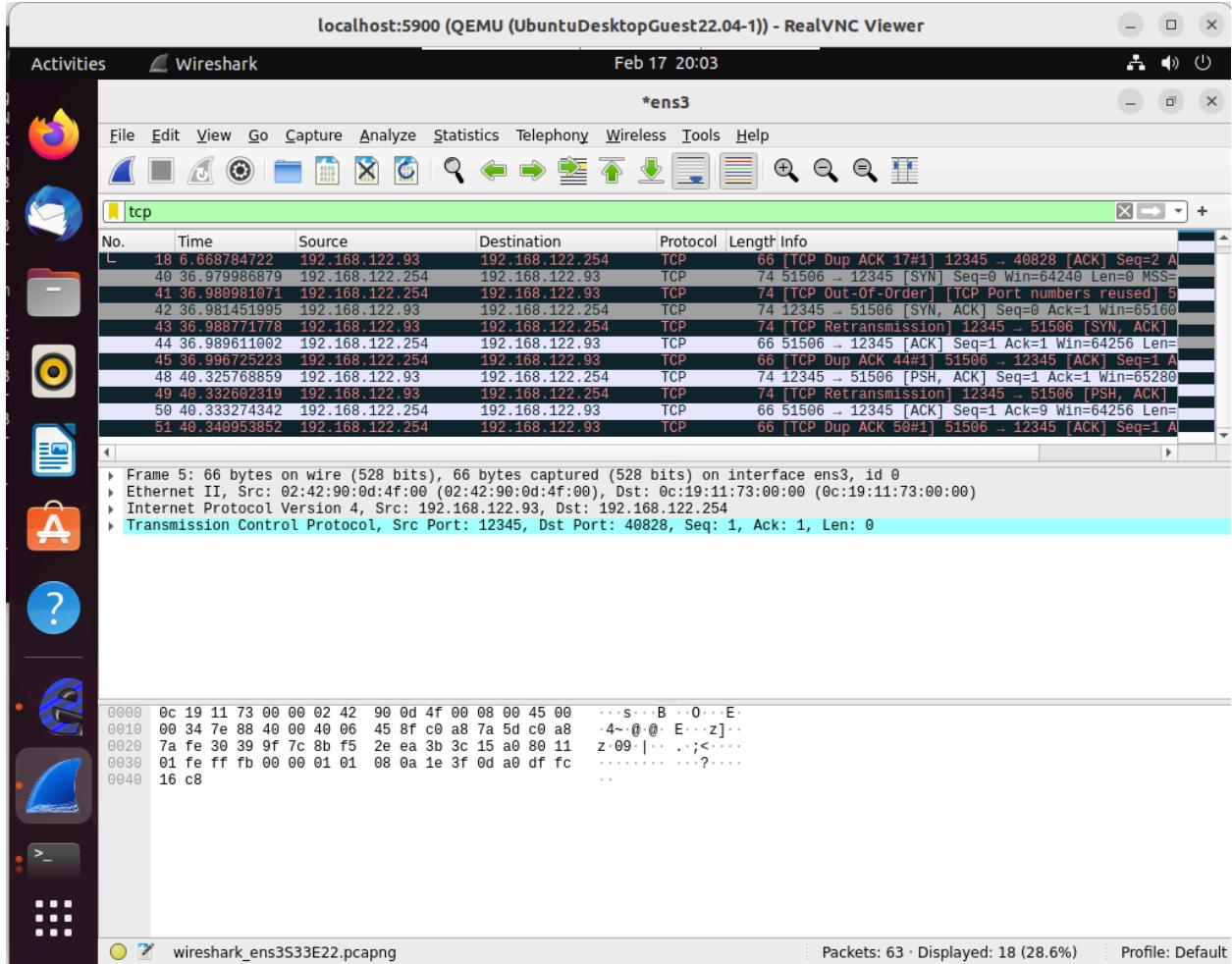
5. Next, start Wireshark by running the command "sudo wireshark" in another terminal. Select the same interface that you told ettercap to sniff on (in the above example it was "ens3". We will be using Wireshark to see the data transiting between the two systems.



6. In the Wireshark filter window type "tcp" to filter out all packets except TCP.



7. Repeat the netcat experiment in the last question.



8. Why does the above happen? In your explanation please include screenshots of arp tables from both victim systems (you can use the arp -n command to view the arp tables).

```
root@UbuntuDockerGuest-1:~# ifconfig
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 0 bytes 0 (0.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 0 bytes 0 (0.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@UbuntuDockerGuest-1:~# netcat -l -p 12345
test
^C
root@UbuntuDockerGuest-1:~# netcat -l -p 12345
testing
arp -n
^C
root@UbuntuDockerGuest-1:~# arp -n
Address      Hwtype  Hwaddress      Flags Mask      Iface
192.168.122.254   ether  0c:19:11:73:00:00  C          eth0
192.168.122.1    ether  52:54:00:ac:52:a3  C          eth0
192.168.122.63   ether  0c:19:11:73:00:00  C          eth0
root@UbuntuDockerGuest-1:~# 

root@UbuntuDockerGuest-2:~# ifconfig
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 0 bytes 0 (0.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 0 bytes 0 (0.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@UbuntuDockerGuest-2:~# netcat 192.168.122.93 12345
test
^C
root@UbuntuDockerGuest-2:~# netcat 192.168.122.93 12345
testing
arp -n
^C
root@UbuntuDockerGuest-2:~# arp -n
Address      Hwtype  Hwaddress      Flags Mask      Iface
192.168.122.1    ether  52:54:00:ac:52:a3  C          eth0
192.168.122.63   ether  0c:19:11:73:00:00  C          eth0
root@UbuntuDockerGuest-2:~# 
```

From our pictures, on LinuxHost1, we see that all the IP addresses are mapped to the same MAC address which suggests that LinuxHost1 believes all of these IP addresses are originating from a single device which may be the attacker's machine. From the second image with LinuxHost2, we see that there are multiple IP addresses linked to the same MAC address which

shows that this machine believes these IPs belong to the same system. For this reason it can be inferred that the ARP poisoning was successful as we are able to impersonate hosts.

#### 9. Stop the attack by choosing the "Stop MITM Attacks" from the "MITM menu":

