Person in the Middle via ARP Spoofing
Author: Ashok Khare

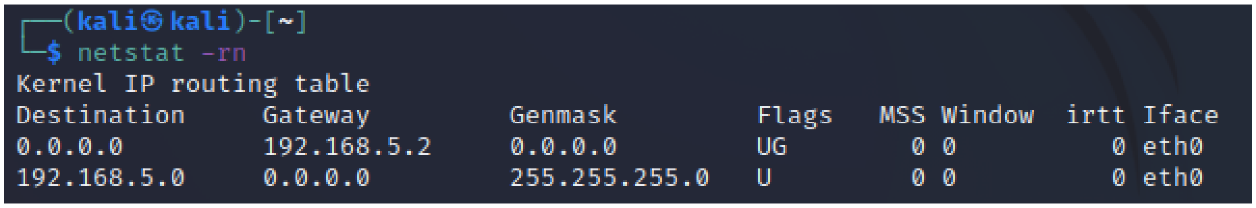
a) The MAC address of Kali's main interface (eth0) is 00:0c:29:51:85:e8

b) The IP address of Kali's main interface (eth0) is 192.168.5.130

c) The MAC address of Metasploitable's main interface (eth0) is 00:0c:29:70:a0:68

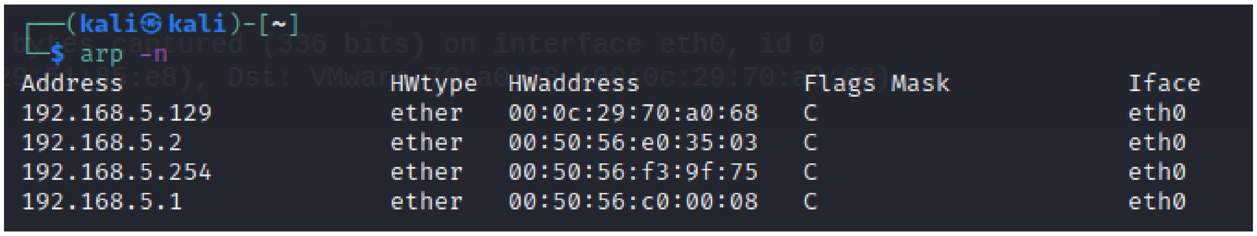
d) The IP address of Metasploitable's main interface (eth0) is 192.168.5.129

e) Kali's routing table is:



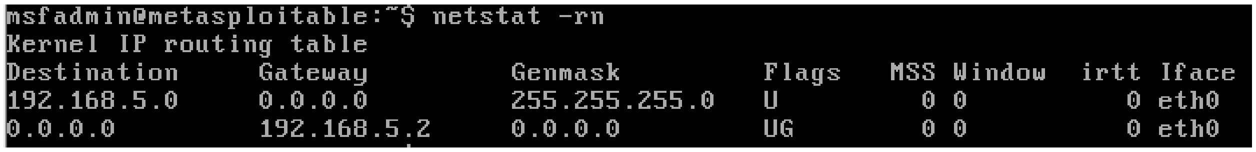
```
┌──(kali㉿kali)-[~]
└─$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         192.168.5.2     0.0.0.0         UG        0 0          0 eth0
192.168.5.0     0.0.0.0         255.255.255.0   U         0 0          0 eth0
```

f) Kali's ARP cache is:



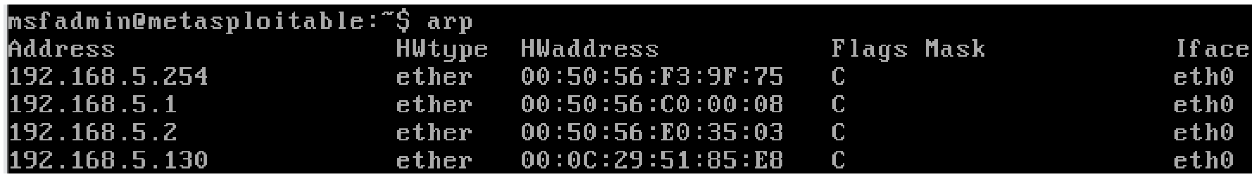
```
┌──(kali㉿kali)-[~]
└─$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.5.129            ether   00:0c:29:70:a0:68   C                     eth0
192.168.5.2              ether   00:50:56:e0:35:03   C                     eth0
192.168.5.254            ether   00:50:56:f3:9f:75   C                     eth0
192.168.5.1              ether   00:50:56:c0:00:08   C                     eth0
```

g) Metasploitable's routing table is:



```
msfadmin@metasploitable:~$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.168.5.0     0.0.0.0         255.255.255.0   U         0 0          0 eth0
0.0.0.0         192.168.5.2     0.0.0.0         UG        0 0          0 eth0
```

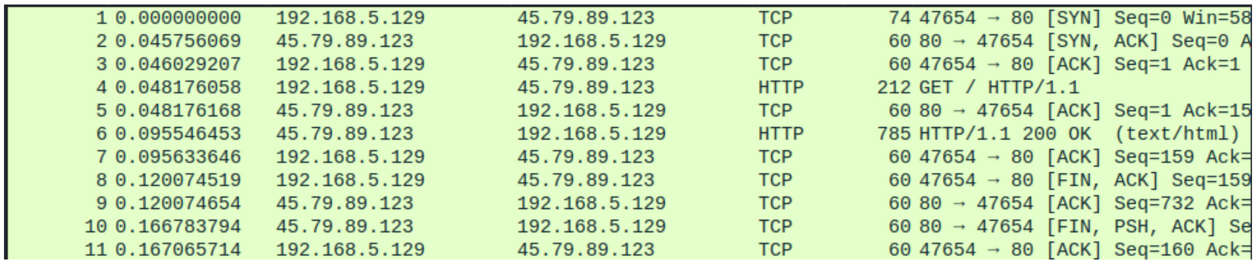
h) Metasploitable's ARP cache is:



```
msfadmin@metasploitable:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.5.254            ether   00:50:56:F3:9F:75   C                     eth0
192.168.5.1              ether   00:50:56:C0:00:08   C                     eth0
192.168.5.2              ether   00:50:56:E0:35:03   C                     eth0
192.168.5.130            ether   00:0C:29:51:85:E8   C                     eth0
```

i) To get the CS338 Sandbox page via the given command, Metasploitable should send the TCP SYN packet to the MAC address 00:50:56:e0:35:03. This address is the address of the gateway. Metasploitable must send the packet through the gateway because the IP address of the system hosting the Sandbox page is not of the form 192.168.5.* - in other words, because the system is not on the local network.

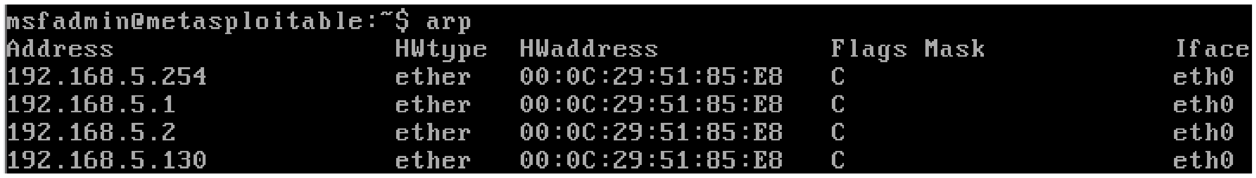
j) There is an HTTP response on Metasploitable. It won't let me scroll up to see the specific details of the response, but it is showing the HTTP content of the page, leading me to believe it is showing a 200 OK response. The packets from the exchange were also captured on Kali's Wireshark. The packets consist of the TCP handshake, the HTTP query and response, and the TCP FIN packets. The exact packets are shown below:



```
 1 0.000000000   192.168.5.129        45.79.89.123         TCP        74 47654 → 80 [SYN] Seq=0 Win=58
 2 0.045756069   45.79.89.123         192.168.5.129        TCP        60 80 → 47654 [SYN, ACK] Seq=0 A
 3 0.046029207   192.168.5.129        45.79.89.123         TCP        60 47654 → 80 [ACK] Seq=1 Ack=1
 4 0.048176058   192.168.5.129        45.79.89.123         HTTP      212 GET / HTTP/1.1
 5 0.048176168   45.79.89.123         192.168.5.129        TCP        60 80 → 47654 [ACK] Seq=1 Ack=15
 6 0.095546453   45.79.89.123         192.168.5.129        HTTP      785 HTTP/1.1 200 OK  (text/html)
 7 0.095633646   192.168.5.129        45.79.89.123         TCP        60 47654 → 80 [ACK] Seq=159 Ack=
 8 0.120074519   192.168.5.129        45.79.89.123         TCP        60 47654 → 80 [FIN, ACK] Seq=159
 9 0.120074654   45.79.89.123         192.168.5.129        TCP        60 80 → 47654 [ACK] Seq=732 Ack=
10 0.166783794   45.79.89.123         192.168.5.129        TCP        60 80 → 47654 [FIN, PSH, ACK] Se
11 0.167065714   192.168.5.129        45.79.89.123         TCP        60 47654 → 80 [ACK] Seq=160 Ack=
```

k) ARP poisoning performed

l) Metasploitable's ARP cache has been changed. Now the MAC address listed for each of the registered IP addresses is the same as the MAC address for Kali's IP address (192.168.5.130). The new ARP cache is:

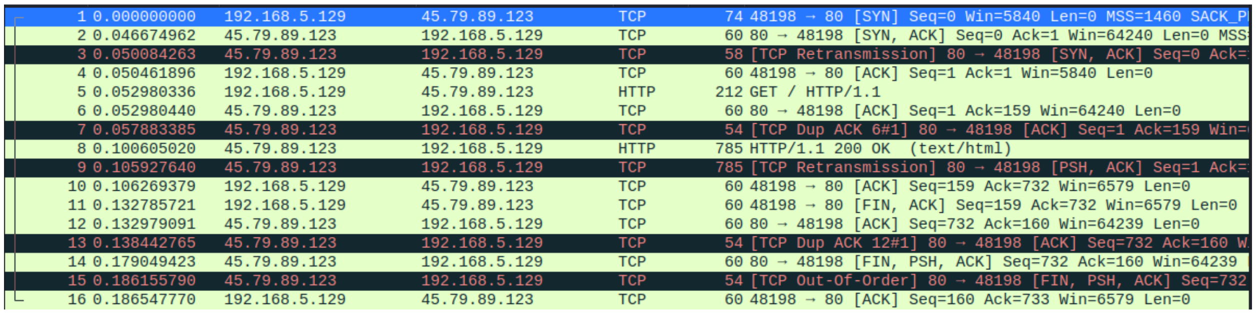


```
msfadmin@metasploitable:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.5.254            ether   00:0C:29:51:85:E8   C                     eth0
192.168.5.1              ether   00:0C:29:51:85:E8   C                     eth0
192.168.5.2              ether   00:0C:29:51:85:E8   C                     eth0
192.168.5.130            ether   00:0C:29:51:85:E8   C                     eth0
```

m) Now, if I try to execute 'curl http://cs338.jeffondich.com/' on Metasploitable, I think that Metasploitable will send the initial TCP SYN packet to Kali's MAC address. Although Metasploitable would still (at least to its knowledge) be sending the packet to the gateway for the reasons described in part (i), because the gateway's MAC address has been changed to Kali's MAC address in Metasploitable's ARP cache, the packet will actually be sent to Kali.

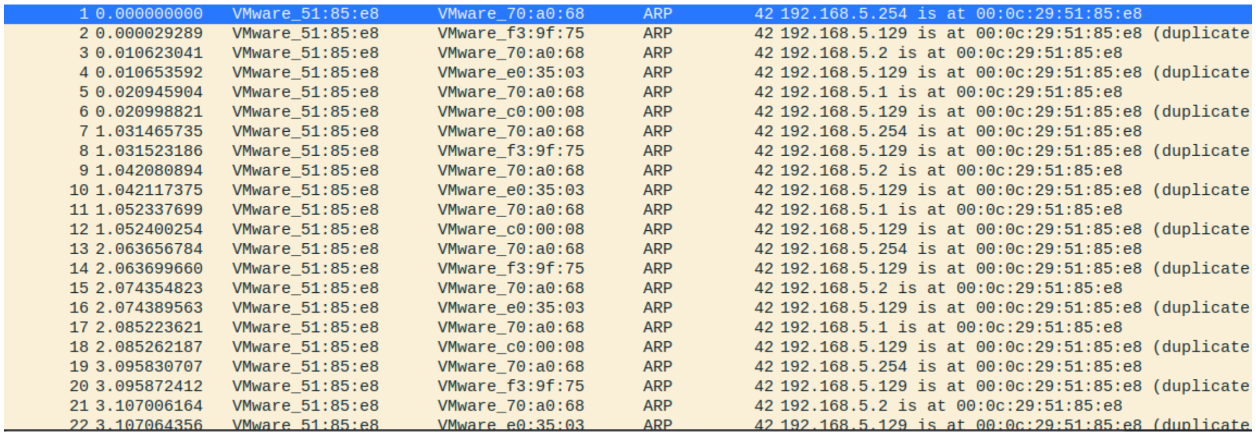
n) New capture performed on Wireshark

o) Metasploitable once again got an HTTP response; similarly to part (j), I can't scroll up to see the full response, but the page's HTML is displayed on Metasploitable, so I assume that Metasploitable got a 200 OK response. Wireshark also captured the packets in the exchange. From these packets, we can see the TCP handshake, the HTTP query and response, and the FIN packets. However, as shown below, there are some additional packets, as shown below. The TCP Retransmissions are sent if acknowledgement of a prior packet is not received after a certain amount of time, according to https://accedian.com/blog/network-packet-loss-retransmissions-and-duplicate-acknowledgements/. The Dup ACK packets are sent if the receiver notices that packets were not received in numerical order, or if there was a gap, according to https://networkengineering.stackexchange.com/questions/38471/what-does-tcp-dup-ack-mean. Finally, the TCP Out of Order packet indicates that the packet was received in a different order than it was sent, according to https://networkengineering.stackexchange.com/questions/38471/what-does-tcp-dup-ack-mean. The presence of these three new kinds of packets likely indicate some network delays or disruptions in packet order, which could be due to the fact that the packets are now being sent through Kali as an intermediary.



```
  1 0.000000000   192.168.5.129       45.79.89.123        TCP         74 48198 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_P
  2 0.046674962   45.79.89.123        192.168.5.129       TCP         60 80 → 48198 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS
  3 0.050084263   45.79.89.123        192.168.5.129       TCP         58 [TCP Retransmission] 80 → 48198 [SYN, ACK] Seq=0 Ack=
  4 0.050461896   192.168.5.129       45.79.89.123        TCP         60 48198 → 80 [ACK] Seq=1 Ack=1 Win=5840 Len=0
  5 0.052980336   192.168.5.129       45.79.89.123        HTTP       212 GET / HTTP/1.1
  6 0.052980440   45.79.89.123        192.168.5.129       TCP         60 80 → 48198 [ACK] Seq=1 Ack=159 Win=64240 Len=0
  7 0.057883385   45.79.89.123        192.168.5.129       TCP         54 [TCP Dup ACK 6#1] 80 → 48198 [ACK] Seq=1 Ack=159 Win=
  8 0.100605020   45.79.89.123        192.168.5.129       HTTP       785 HTTP/1.1 200 OK  (text/html)
  9 0.105927640   45.79.89.123        192.168.5.129       TCP        785 [TCP Retransmission] 80 → 48198 [PSH, ACK] Seq=1 Ack=
 10 0.106269379   192.168.5.129       45.79.89.123        TCP         60 48198 → 80 [ACK] Seq=159 Ack=732 Win=6579 Len=0
 11 0.132785721   192.168.5.129       45.79.89.123        TCP         60 48198 → 80 [FIN, ACK] Seq=159 Ack=732 Win=6579 Len=0
 12 0.132979091   45.79.89.123        192.168.5.129       TCP         60 80 → 48198 [ACK] Seq=732 Ack=160 Win=64239 Len=0
 13 0.138442765   45.79.89.123        192.168.5.129       TCP         54 [TCP Dup ACK 12#1] 80 → 48198 [ACK] Seq=732 Ack=160 W
 14 0.179049423   45.79.89.123        192.168.5.129       TCP         60 80 → 48198 [FIN, PSH, ACK] Seq=732 Ack=160 Win=64239
 15 0.186155790   45.79.89.123        192.168.5.129       TCP         54 [TCP Out-Of-Order] 80 → 48198 [FIN, PSH, ACK] Seq=732
 16 0.186547770   192.168.5.129       45.79.89.123        TCP         60 48198 → 80 [ACK] Seq=160 Ack=733 Win=6579 Len=0
```

p) It would appear that Kali changed Metasploitable's ARP cache by sending out ARP broadcasts that claimed the corresponding MAC addresses for each of the IPs on the local server are the same as Kali's MAC address. This is shown in the image below. The source is likely Kali Ettercap, the destination is likely Metasploitable, and the ARP packets captured show the source broadcasting that all open addresses can be found at Kali's MAC address (00:0c:29:51:85:e8). Because all IP addresses now correspond to Kali's MAC address, whenever Metasploitable tries to send something anywhere on the network, it will end up going to Kali.



```
  1 0.000000000   VMware_51:85:e8     VMware_70:a0:68     ARP         42 192.168.5.254 is at 00:0c:29:51:85:e8
  2 0.000029289   VMware_51:85:e8     VMware_f3:9f:75     ARP         42 192.168.5.129 is at 00:0c:29:51:85:e8 (duplicate
  3 0.010623041   VMware_51:85:e8     VMware_70:a0:68     ARP         42 192.168.5.2 is at 00:0c:29:51:85:e8
  4 0.010653592   VMware_51:85:e8     VMware_e0:35:03     ARP         42 192.168.5.129 is at 00:0c:29:51:85:e8 (duplicate
  5 0.020945904   VMware_51:85:e8     VMware_70:a0:68     ARP         42 192.168.5.1 is at 00:0c:29:51:85:e8
  6 0.020998821   VMware_51:85:e8     VMware_c0:00:08     ARP         42 192.168.5.129 is at 00:0c:29:51:85:e8 (duplicate
  7 1.031465735   VMware_51:85:e8     VMware_70:a0:68     ARP         42 192.168.5.254 is at 00:0c:29:51:85:e8
  8 1.031523186   VMware_51:85:e8     VMware_f3:9f:75     ARP         42 192.168.5.129 is at 00:0c:29:51:85:e8 (duplicate
  9 1.042080894   VMware_51:85:e8     VMware_70:a0:68     ARP         42 192.168.5.2 is at 00:0c:29:51:85:e8
 10 1.042117375   VMware_51:85:e8     VMware_e0:35:03     ARP         42 192.168.5.129 is at 00:0c:29:51:85:e8 (duplicate
 11 1.052337699   VMware_51:85:e8     VMware_70:a0:68     ARP         42 192.168.5.1 is at 00:0c:29:51:85:e8
 12 1.052400254   VMware_51:85:e8     VMware_c0:00:08     ARP         42 192.168.5.129 is at 00:0c:29:51:85:e8 (duplicate
 13 2.063656784   VMware_51:85:e8     VMware_70:a0:68     ARP         42 192.168.5.254 is at 00:0c:29:51:85:e8
 14 2.063699660   VMware_51:85:e8     VMware_f3:9f:75     ARP         42 192.168.5.129 is at 00:0c:29:51:85:e8 (duplicate
 15 2.074354823   VMware_51:85:e8     VMware_70:a0:68     ARP         42 192.168.5.2 is at 00:0c:29:51:85:e8
 16 2.074389563   VMware_51:85:e8     VMware_e0:35:03     ARP         42 192.168.5.129 is at 00:0c:29:51:85:e8 (duplicate
 17 2.085223621   VMware_51:85:e8     VMware_70:a0:68     ARP         42 192.168.5.1 is at 00:0c:29:51:85:e8
 18 2.085262187   VMware_51:85:e8     VMware_c0:00:08     ARP         42 192.168.5.129 is at 00:0c:29:51:85:e8 (duplicate
 19 3.095830707   VMware_51:85:e8     VMware_70:a0:68     ARP         42 192.168.5.254 is at 00:0c:29:51:85:e8
 20 3.095872412   VMware_51:85:e8     VMware_f3:9f:75     ARP         42 192.168.5.129 is at 00:0c:29:51:85:e8 (duplicate
 21 3.107006164   VMware_51:85:e8     VMware_70:a0:68     ARP         42 192.168.5.2 is at 00:0c:29:51:85:e8
 22 3.107064356   VMware_51:85:e8     VMware_e0:35:03     ARP         42 192.168.5.129 is at 00:0c:29:51:85:e8 (duplicate
```

q) If I were to design an ARP spoofing detector, I would probably have it check the ARP cache for IP addresses with duplicate MAC addresses. I would probably have it sound an alarm if the IP address had the same MAC address as even just one other IP address, since there's probably a way to change a smaller selection of MAC addresses so it's not so obvious that the whole ARP cache has been messed with. I don't believe that this would cause any false positives, since MAC addresses on a local network must always be unique. Therefore, there should never be duplicates unless the MAC addresses or ARP caches were tampered with.