

Names: PJ Sangvong , Luisa Escosteguy

A. What is Kali's main interface's MAC address? (The main interface is probably called eth0, but check ifconfig to be sure.)

00:0c:29:66:26:86

B. What is Kali's main interface's IP address?

172.16.151.128

C. What is Metasploitable's main interface's MAC address?

00:0c:29:46:ba:a6

D. What is Metasploitable's main interface's IP address?

172.16.151.131

E. Show Kali's routing table. (Use "netstat -r" to see it with symbolic names, or "netstat -rn" to see it with numerical addresses.)

```
(kali㉿kali)-[~]
$ netstat -r
Kernel IP routing table
Destination        Gateway            Genmask           Flags     MSS Window  irtt Iface
default            172.16.151.2      0.0.0.0           UG        0 0        0 eth0
172.16.151.0       0.0.0.0           255.255.255.0     U         0 0        0 eth0
```

F. Show Kali's ARP cache. (Use "arp" or "arp -n".)

```
(kali㉿kali)-[~]
$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
172.16.151.2             ether   00:50:56:e2:87:36   C                    eth0
```

G. Show Metasploitable's routing table.

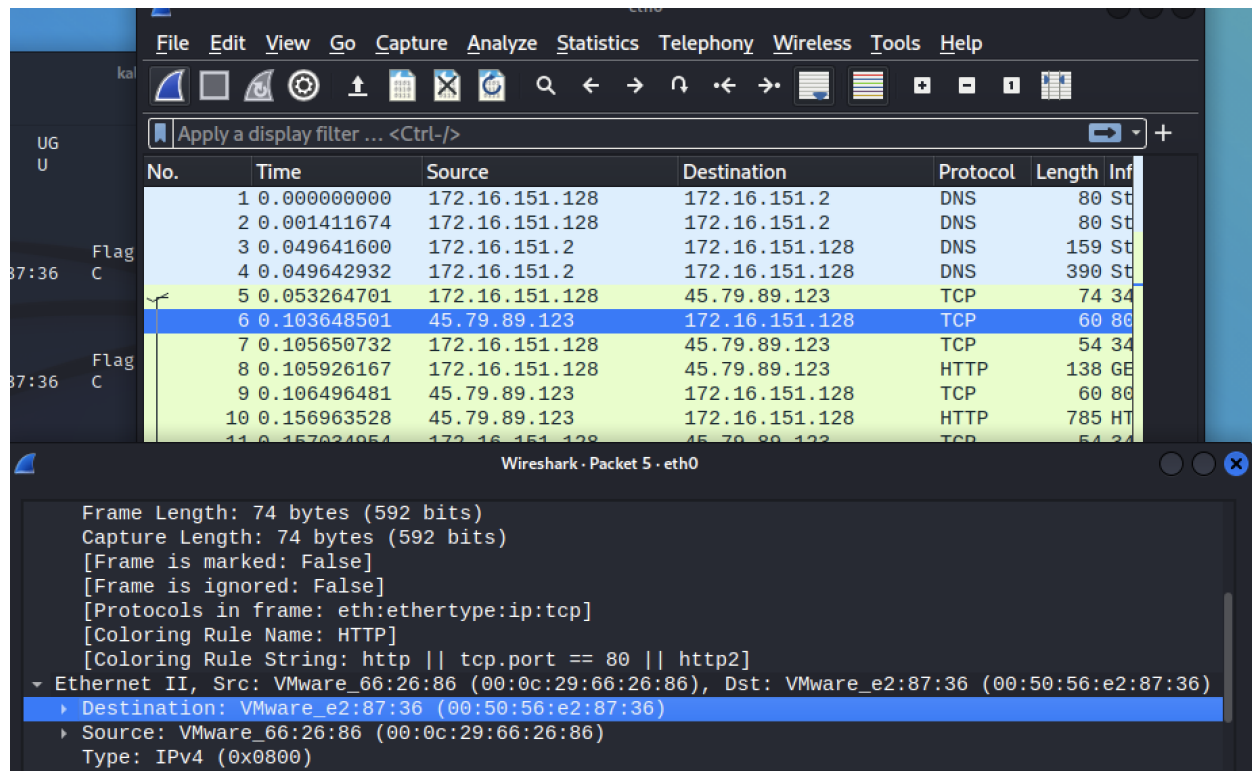
```
msfadmin@metasploitable:~$ netstat -r
Kernel IP routing table
Destination        Gateway            Genmask           Flags     MSS Window  irtt Iface
172.16.151.0       *                  255.255.255.0     U         0 0        0 eth0
default            172.16.151.2      0.0.0.0           UG        0 0        0 eth0
```

H. Show Metasploitable's ARP cache.

```
msfadmin@metasploitable:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
172.16.151.128         ether   00:0C:29:66:26:86   C                    eth0
172.16.151.254         ether   00:50:56:FD:20:82   C                    eth0
172.16.151.2           ether   00:50:56:E2:87:36   C                    eth0
```

- I. Suppose the user of Metasploitable wants to get the CS338 sandbox page via the command "curl http://cs338.jeffondich.com/". To which MAC address should Metasploitable send the TCP SYN packet to get the whole HTTP query started? Explain why.

Metasploitable will send the TCP SYN packet to VMware MAC (00:50:56:e2:87:36), as this is our "gateway" to the internet. We tested this by running curl on kali with wireshark opened.



- J. Fire up Wireshark on Kali. Start capturing packets for "tcp port http". On Metasploitable, execute "curl http://cs338.jeffondich.com/". On Kali, stop capturing. Do you see an HTTP response on Metasploitable? Do you see any captured packets in Wireshark on Kali?

We see an HTTP response on Metasploitable, which is the HTML source code of <http://cs338.jeffondich.com/>. We did not see any captured packets in Wireshark on Kali.

- K. Now, it's time to be Mal (who will, today, merely eavesdrop). Use Ettercap to do ARP spoofing (also known as ARP Cache Poisoning) with Metasploitable as your target. There are many online tutorials on how to do this ([here's one](#)). Find one you like, and start spoofing your target. NOTE: most of these tutorials are showing an old user interface for Ettercap, which may make them confusing. The steps you're trying to take within Ettercap are:
- Start sniffing (not bridged sniffing) on eth0

- b. Scan for Hosts
- c. View the Hosts list
- d. Select your Metasploit VM from the Host List
- e. Add that host as Target 1
- f. Start ARP Poisoning (including Sniff Remote Connections)
- g. Do your stuff with wireshark and Metasploit
- h. Stop ARP Poisoning

Done! See question O for screenshots.

L. Show Metasploitable's ARP cache. How has it changed?

We now have 172.16.151.1 added into the list of addresses. All the MAC addresses also changed to Kali's MAC address.

```
msfadmin@metasploitable:~$ arp
```

Address	HWtype	HWaddress	Flags	Mask	Iface
172.16.151.1	ether	00:0C:29:66:26:86	C		eth0
172.16.151.254	ether	00:0C:29:66:26:86	C		eth0
172.16.151.2	ether	00:0C:29:66:26:86	C		eth0
172.16.151.128	ether	00:0C:29:66:26:86	C		eth0

M. Without actually doing it yet, predict what will happen if you execute "curl http://cs338.jeffondich.com/" on Metasploitable now. Specifically, to what MAC address will Metasploitable send the TCP SYN packet? Explain why.

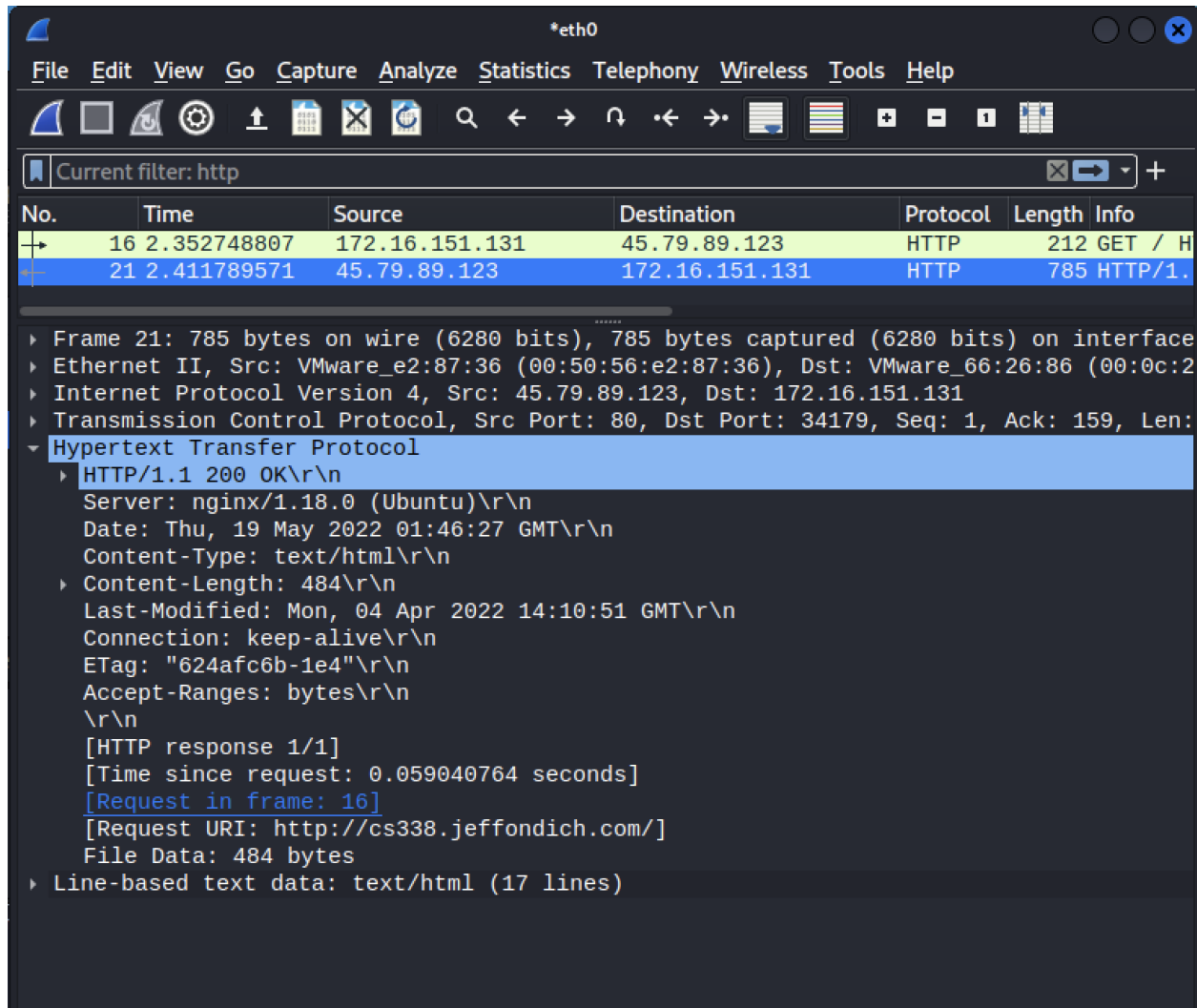
We predict Metasploitable will send the TCP SYN packet to 00:0c:29:66:26:86, which is Kali's MAC address. This will happen because we introduced a person in the middle between Metasploitable and the gateway.

N. Start Wireshark capturing "tcp port http" again.

OK

O. Execute "curl http://cs338.jeffondich.com/" on Metasploitable. On Kali, stop capturing. Do you see an HTTP response on Metasploitable? Do you see captured packets in Wireshark? Can you tell from Kali what messages went back and forth between Metasploitable and cs338.jeffondich.com?

Now that we have Ettercap running, we can see packets sent between Metasploitable and cs338.jeffondich.com. We can see everything: Metasploitable sent an HTTP request to <http://cs338.jeffondich.com/>, then <http://cs338.jeffondich.com/> sent back an HTTP response to Metasploitable. Looking at the screenshot, we can see everything that is in the HTTP response (the content of the page, the length of the page, the type of request, etc.).



- P. Explain in detail what happened. How did Kali change Metasploitable's ARP cache? (If you want to watch the attack in action, try stopping the PITM/MITM attack by selecting "Stop mitm attack(s)" from Ettercap's Mitm menu, starting a Wireshark capture for "arp", and restarting the ARP poisoning attack in Ettercap.)

Basically, Kali MAC address was linked to the legitimate gateway IP address.

From the screenshot, the first 6 packets were when we had Ettercap Mitm on, and starting from the 7th packet is when we had Ettercap turned off. When it was on, Kali sent an ARP packet to the 4 different MAC addresses (sent to 4c:ba:a6 3 times) and told those addresses where the MAC addresses of the IPs 172.16.151.254, 172.16.151.131, 172.16.151.2, and/or 172.16.151.1 are at. In the info, we see that the ARP packets are telling those addresses that their MAC address is 00:0c:29:66:26:86 which is Kali's MAC address.

When we closed Ettercap Mitm, Kali repeated the same protocol, but instead of telling all the addresses to look at Kali's MAC address, it told the addresses to look at their own MAC addresses instead.

Wireshark - Capturing from eth0 (arp)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	VMware_66:26:86	VMware_4c:ba:a6	ARP	42	172.16.151.254 is at 00:0c:29:66:26:86
2	0.000058207	VMware_66:26:86	VMware_fd:20:82	ARP	42	172.16.151.131 is at 00:0c:29:66:26:86 (duplicate)
3	0.010252693	VMware_66:26:86	VMware_4c:ba:a6	ARP	42	172.16.151.2 is at 00:0c:29:66:26:86
4	0.010303410	VMware_66:26:86	VMware_e2:87:36	ARP	42	172.16.151.131 is at 00:0c:29:66:26:86 (duplicate)
5	0.020460822	VMware_66:26:86	VMware_4c:ba:a6	ARP	42	172.16.151.1 is at 00:0c:29:66:26:86
6	0.020511375	VMware_66:26:86	f2:18:98:c1:1f:65	ARP	42	172.16.151.131 is at 00:0c:29:66:26:86 (duplicate)
7	4.127285912	VMware_66:26:86	VMware_4c:ba:a6	ARP	42	172.16.151.254 is at 00:50:56:fd:20:82
8	4.127429496	VMware_66:26:86	VMware_fd:20:82	ARP	42	172.16.151.131 is at 00:0c:29:4c:ba:a6 (duplicate)
9	4.137658524	VMware_66:26:86	VMware_4c:ba:a6	ARP	42	172.16.151.2 is at 00:50:56:e2:87:36
10	4.137729633	VMware_66:26:86	VMware_e2:87:36	ARP	42	172.16.151.131 is at 00:0c:29:4c:ba:a6 (duplicate)
11	4.147975778	VMware_66:26:86	VMware_4c:ba:a6	ARP	42	172.16.151.1 is at f2:18:98:c1:1f:65
12	4.148114400	VMware_66:26:86	f2:18:98:c1:1f:65	ARP	42	172.16.151.131 is at 00:0c:29:4c:ba:a6 (duplicate)
13	5.158512789	VMware_66:26:86	VMware_4c:ba:a6	ARP	42	172.16.151.254 is at 00:50:56:fd:20:82
14	5.158611716	VMware_66:26:86	VMware_fd:20:82	ARP	42	172.16.151.131 is at 00:0c:29:4c:ba:a6 (duplicate)
15	5.168828056	VMware_66:26:86	VMware_4c:ba:a6	ARP	42	172.16.151.2 is at 00:50:56:e2:87:36
16	5.168895194	VMware_66:26:86	VMware_e2:87:36	ARP	42	172.16.151.131 is at 00:0c:29:4c:ba:a6 (duplicate)
17	5.179066479	VMware_66:26:86	VMware_4c:ba:a6	ARP	42	172.16.151.1 is at f2:18:98:c1:1f:65
18	5.179124910	VMware_66:26:86	f2:18:98:c1:1f:65	ARP	42	172.16.151.131 is at 00:0c:29:4c:ba:a6 (duplicate)
19	6.189433400	VMware_66:26:86	VMware_4c:ba:a6	ARP	42	172.16.151.254 is at 00:50:56:fd:20:82
20	6.189511551	VMware_66:26:86	VMware_fd:20:82	ARP	42	172.16.151.131 is at 00:0c:29:4c:ba:a6 (duplicate)
21	6.199669123	VMware_66:26:86	VMware_4c:ba:a6	ARP	42	172.16.151.2 is at 00:50:56:e2:87:36
22	6.199721448	VMware_66:26:86	VMware_e2:87:36	ARP	42	172.16.151.131 is at 00:0c:29:4c:ba:a6 (duplicate)
23	6.209883501	VMware_66:26:86	VMware_4c:ba:a6	ARP	42	172.16.151.1 is at f2:18:98:c1:1f:65
24	6.209951296	VMware_66:26:86	f2:18:98:c1:1f:65	ARP	42	172.16.151.131 is at 00:0c:29:4c:ba:a6 (duplicate)

Frame 7: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth0, id 0
 Ethernet II, Src: VMware_66:26:86 (00:0c:29:66:26:86), Dst: VMware_4c:ba:a6 (00:0c:29:4c:ba:a6)
 Address Resolution Protocol (reply)

0000 00 0c 29 4c ba a6 00 0c 29 66 26 86 08 06 00 01 ...L...)f& ...
 0010 08 00 06 04 00 02 00 50 56 fd 20 82 ac 10 97 feP V.....

Expert Info [ws.expert] Packets: 24 · Displayed: 24 (100.0%) Profile: Default

Q. If you wanted to design an ARP spoofing detector, what would you have your detector do? (As you think about this, consider under what circumstances your detector might generate false positives.)

The ARP cache is a good indication that we are undergoing a spoofing attack. Therefore, if we detect more than one IP address mapped to the same MAC in the cache (see questions H and L) we can say that an attack is most likely happening. This method will generate false positives when the network configurations are outdated, such that there may be old devices and old IPs/MACs in the cache. We can also detect this kind of attack by monitoring a specific pattern of ARP packets, which are trying to modify the sources and destinations. We do this by looking at the Info field (see screenshot).