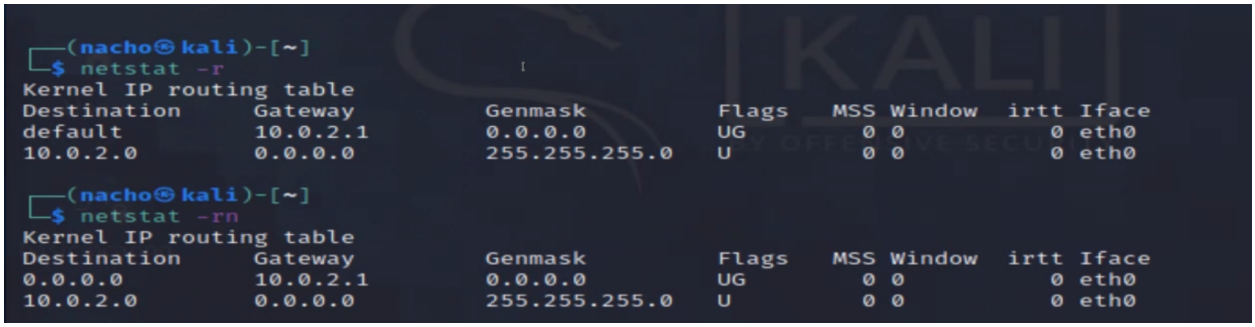
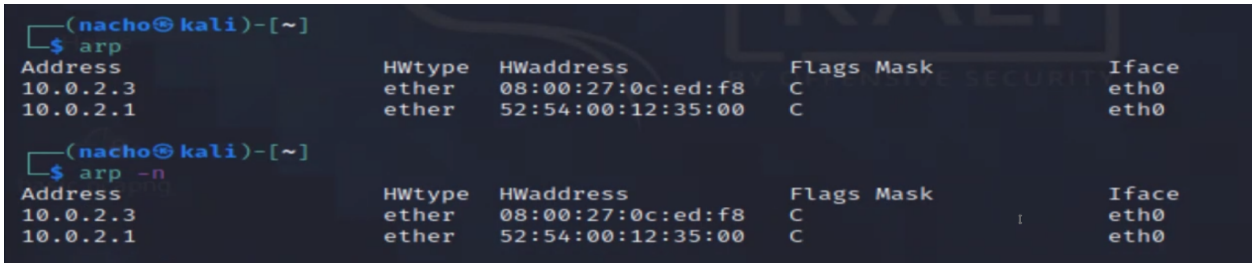
Nacho Rodriguez-Cortes and Kitty Tyree

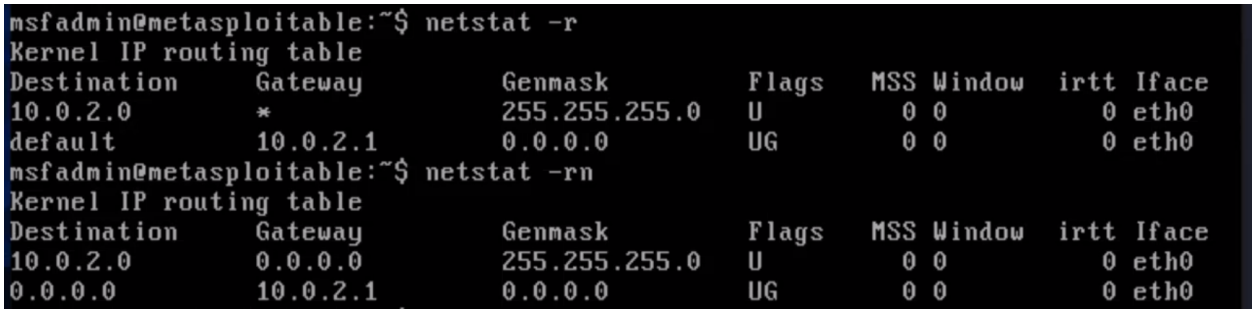
1. What is Kali's main interface's MAC address? (The main interface is probably called eth0, but check ifconfig to be sure.)
    a. 08:00:27:43:dd:00
2. What is Kali's main interface's IP address?
    a. 10.0.2.15
3. What is Metasploitable's main interface's MAC address?
    a. 08:00:27:b1:74:00
4. What is Metasploitable's main interface's IP address?
    a. 10.0.2.4
5. Show Kali's routing table. (Use "netstat -r" to see it with symbolic names, or "netstat -rn" to see it with numerical addresses.)



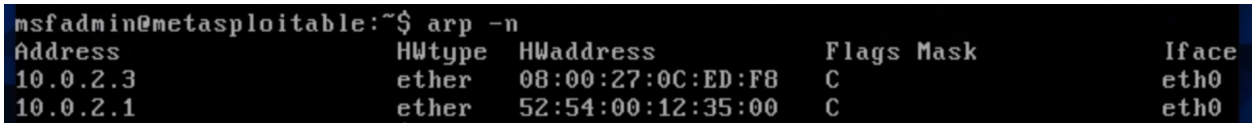
6. Show Kali's ARP cache. (Use "arp" or "arp -n".)



7. Show Metasploitable's routing table.



8. Show Metasploitable's ARP cache.



9. Suppose the user of Metasploitable wants to get the CS231 sandbox page via the command "curl http://cs231.jeffondich.com/". To which MAC address should Metasploitable send the TCP SYN packet to get the whole HTTP query started? Explain why.
    a. 52:54:00:12:35:00. This is the MAC Address for the IP 10.0.2.1. We are using this IP because it is present in both the routing table and the arp cache (10.0.2.3 is an IP address that is only present in the arp cache but not the routing table). Because of the addresses in the routing table, we think that the 10.0.2.1 IP is connected to the 10.0.2.0 address through 0.0.0.0.

10. Fire up Wireshark on Kali. Start capturing packets for "tcp port http". On Metasploitable, execute "curl http://cs231.jeffondich.com/". On Kali, stop capturing. Do you see an HTTP response on Metasploitable? Do you see any captured packets in Wireshark on Kali?
    a. We did get an HTTP response on Metasploitable:

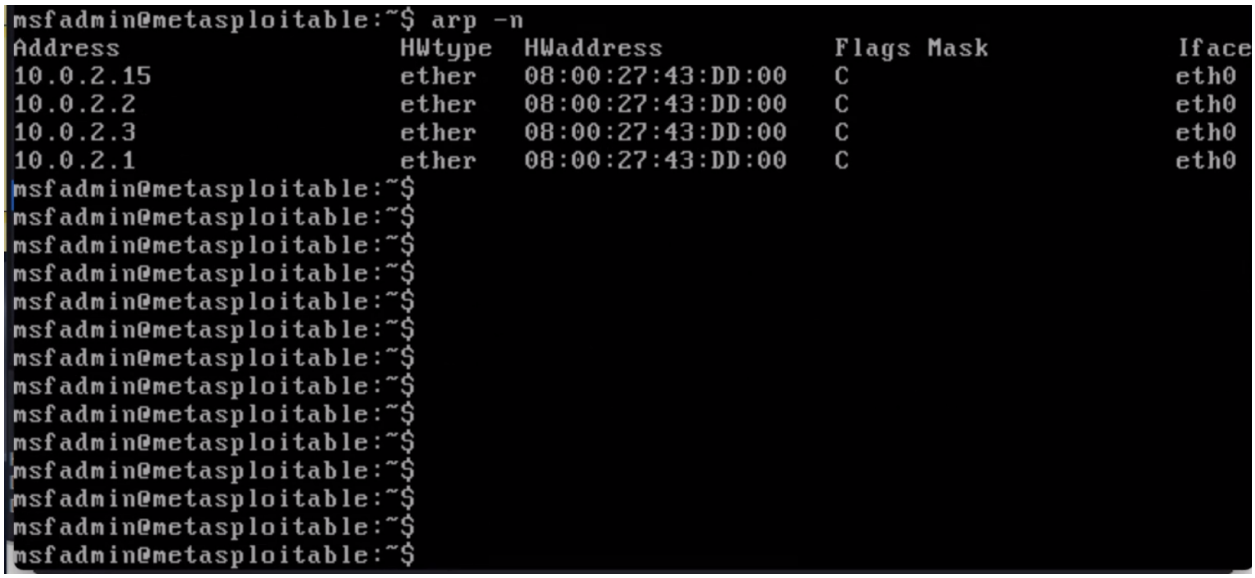
```
msfadmin@metasploitable:~$ curl http://cs231.jeffondich.com/
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>CS231 Sandbox</title>
    </head>

    <body>
        <h1>CS231 Sandbox</h1>
        <h2>Fun with security, or maybe insecurity</h2>

        <ul>
            <li>This page should be the page you retrieve for the "Getting start
ed with Wireshark"
                assignment. Here's my head, as advertised:
                <div><img src="jeff_square_head.jpg" style="width: 100px;"></div
>
            </li>
            <li>The <a href="/basicauth/">secrets</a> for the Basic Authenticati
on Story exercise</li>
        </ul>
    </body>
</html>
```

    Nothing came through on the Wireshark capture that we were running on Kali.

11. *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:*
    a. *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

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



Well, as you can see, there are 2 additional IP addresses listed, the first one being 10.0.2.15 (Kali's main interface IP address) and 10.0.2.2. Also notice that all of the HW addresses are the same and are for the Kali MAC address. Sneaky!

13. If you execute "curl http://cs231.jeffondich.com/" on Metasploitable now, to what MAC address will Metasploitable send the TCP SYN packet? Explain why.
    a. Metasploitable sends the TCP SYN packet to 08:00:27:43:DD:00 because it has no other choice now that Ettercap has poisoned the system and rerouted the paths.

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

15. Execute "curl http://cs231.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 cs231.jeffondich.com?

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.4	45.79.89.123	TCP	74	38130 → 80 [SYN] Seq=0 Wi
2	0.007095612	10.0.2.4	45.79.89.123	TCP	74	[TCP Retransmission] 3813
3	0.053605819	45.79.89.123	10.0.2.4	TCP	60	80 → 38130 [SYN, ACK] Seq
4	0.055423517	45.79.89.123	10.0.2.4	TCP	58	[TCP Out-Of-Order] 80 → 3
5	0.056165661	10.0.2.4	45.79.89.123	TCP	60	38130 → 80 [ACK] Seq=1 Ac
6	0.056165744	10.0.2.4	45.79.89.123	HTTP	212	GET / HTTP/1.1
7	0.063117061	10.0.2.4	45.79.89.123	TCP	54	38130 → 80 [ACK] Seq=1 Ac
8	0.063247959	10.0.2.4	45.79.89.123	TCP	212	[TCP Retransmission] 3813
9	0.109473849	45.79.89.123	10.0.2.4	HTTP	933	HTTP/1.1 200 OK (text/ht
10	0.115527390	45.79.89.123	10.0.2.4	TCP	933	[TCP Retransmission] 80 →
11	0.116279168	10.0.2.4	45.79.89.123	TCP	60	38130 → 80 [ACK] Seq=159
12	0.123108186	10.0.2.4	45.79.89.123	TCP	54	[TCP Dup ACK 11#1] 38130
13	0.124126472	10.0.2.4	45.79.89.123	TCP	60	38130 → 80 [FIN, ACK] Seq
14	0.131095529	10.0.2.4	45.79.89.123	TCP	54	[TCP Out-Of-Order] 38130
15	0.131312080	45.79.89.123	10.0.2.4	TCP	60	80 → 38130 [ACK] Seq=880
16	0.139142239	45.79.89.123	10.0.2.4	TCP	54	[TCP Dup ACK 15#1] 80 → 3
17	0.178409512	45.79.89.123	10.0.2.4	TCP	60	80 → 38130 [FIN, ACK] Seq

a. There is still an HTTP response on Metasploitable. Now, we can see captured packets in Wireshark (because now, the packets are being sent to Kali's MAC address). We can see the TCP handshake (frames 1-5) as well as seeing the GET request (frame 6) with an OK response (frame 9). Notice that the Source IP address is 10.0.2.4, which is Metasploitable's main interface IP.

16. 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.)

Below is the Wireshark capture for the poisoning

00	PcsCompu_b1:74:00	ARP	42 Who has 10.0.2.4? Tell 10.0.2.15
00	PcsCompu_43:dd:00	ARP	60 10.0.2.4 is at 08:00:27:b1:74:00
00	PcsCompu_b1:74:00	ARP	42 10.0.2.3 is at 08:00:27:43:dd:00
00	PcsCompu_0c:ed:f8	ARP	42 10.0.2.4 is at 08:00:27:43:dd:00 (duplicate
00	PcsCompu_b1:74:00	ARP	42 10.0.2.2 is at 08:00:27:43:dd:00
00	RealtekU_12:35:00	ARP	42 10.0.2.4 is at 08:00:27:43:dd:00 (duplicate
00	PcsCompu_b1:74:00	ARP	42 10.0.2.1 is at 08:00:27:43:dd:00
00	RealtekU_12:35:00	ARP	42 10.0.2.4 is at 08:00:27:43:dd:00 (duplicate

Below is the specifics for "Who has 10.0.2.4?..."

```
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender MAC address: PcsCompu_b1:74:00 (08:00:27:b1:74:00)
Sender IP address: 10.0.2.4
Target MAC address: PcsCompu_43:dd:00 (08:00:27:43:dd:00)
Target IP address: 10.0.2.15
```

Below is the packet immediately after "Who has 10.0.2.4?..."

```
Hardware size: 6
Protocol size: 4
Opcode: request (1)
Sender MAC address: PcsCompu_43:dd:00 (08:00:27:43:dd:00)
Sender IP address: 10.0.2.15
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
Target IP address: 10.0.2.4
```

First off, Kali sends a request to see who is at the IP Address 10.0.2.4 (Metasploitable's IP) : [Who has 10.0.2.4? Tell 10.0.2.15]. After they get a response from Metasploitable, Kali repeatedly tells Metasploitable that their IP address is at Kali's own MAC address (seen in the third frame in the first picture

and all following frames). Basically, this tricks Metasploitable into believing that their MAC address is 08:00:27:43:dd:00 when really that address belongs to Kali.

17. 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.)
    a. We were able to see a difference from the arp cache taken before the poisoning and the one taken during because of the added IP addresses and the changed MAC Addresses. One thing to note is that all of the MAC addresses were the same in the table when without spoofing they were different. Our spoofer would flare on any change in MAC addresses (since they should remain unchanged without any sneakiness going on). False positives could occur with software that happens to update and change its MAC address. We could also consider having our spoofer flaring on multiple IP addresses that have the same MAC address. This would only generate a false positive if for some reason a piece of software was running on two different IP addresses but using the same MAC address.