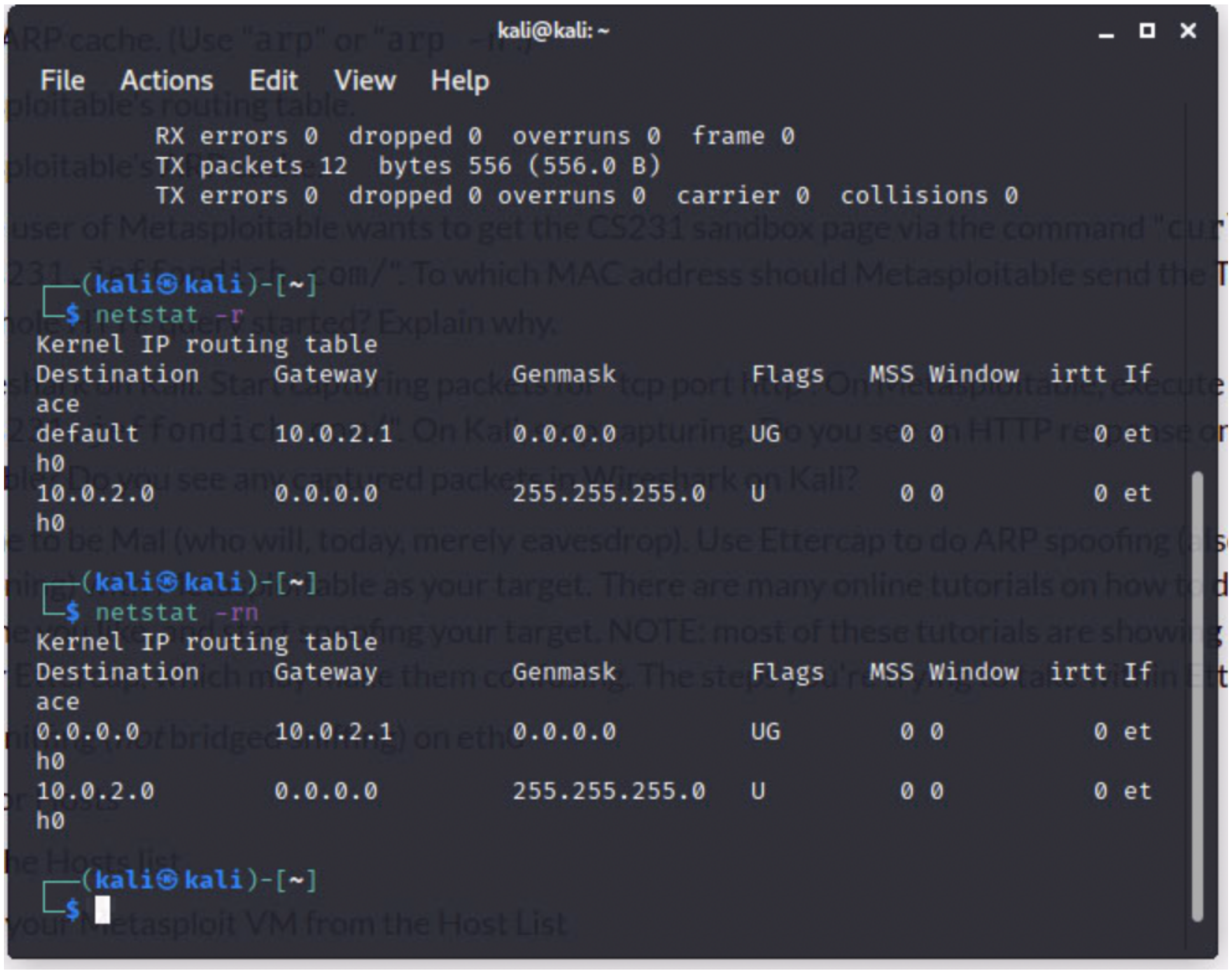
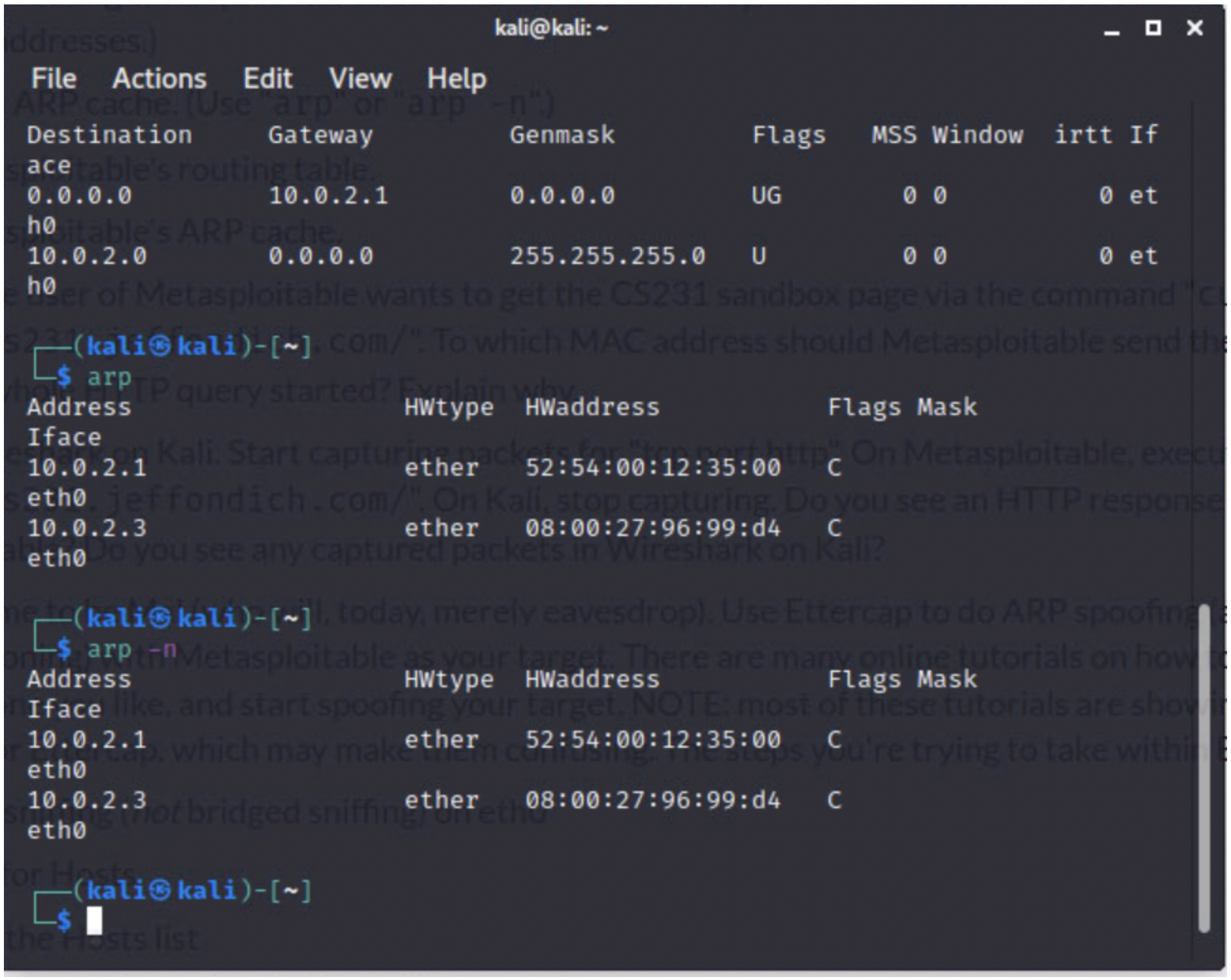
Jake Martens and Josiah Misplon

a. 10.0.2.15
b. 08:00:27:11:cf:53
c. 10.0.2.4
d. 08:00:27:1a:45:12
e. This is the screenshot:



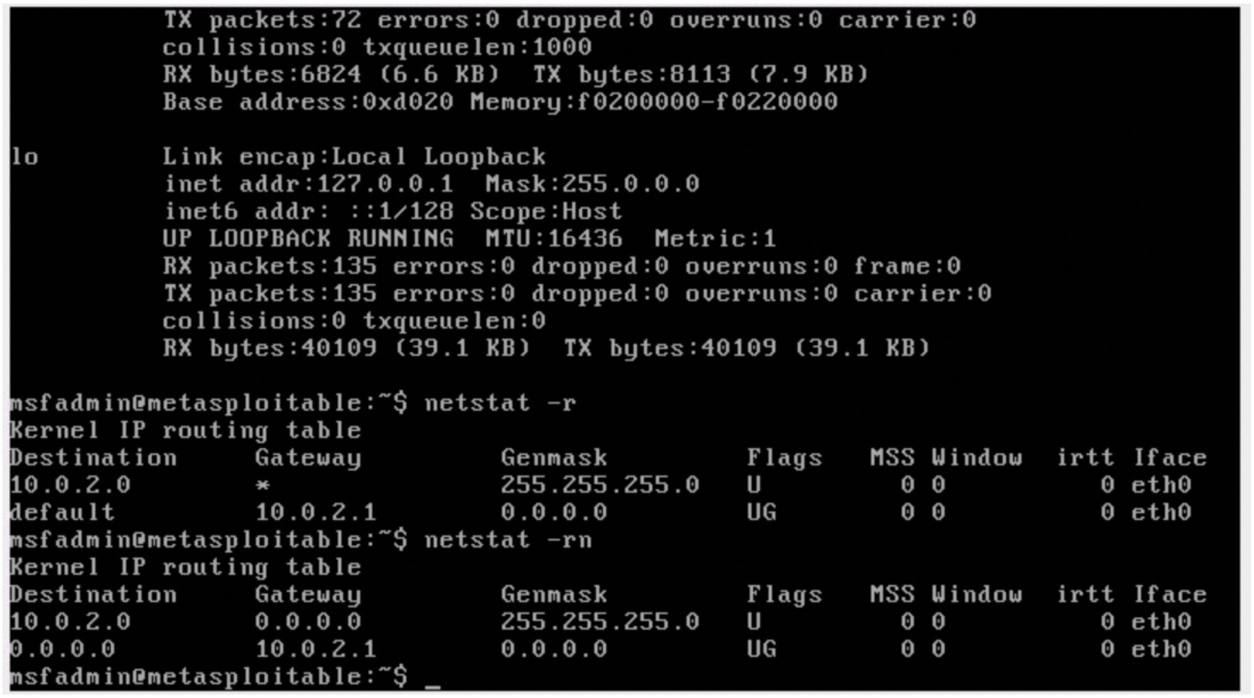
```
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 556 (556.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(kali㉿kali)-[~]
└─$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt If
ace
default         10.0.2.1        0.0.0.0         UG        0 0          0 et
h0
10.0.2.0        0.0.0.0         255.255.255.0   U         0 0          0 et
h0

┌──(kali㉿kali)-[~]
└─$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt If
ace
0.0.0.0         10.0.2.1        0.0.0.0         UG        0 0          0 et
h0
10.0.2.0        0.0.0.0         255.255.255.0   U         0 0          0 et
h0

┌──(kali㉿kali)-[~]
└─$ 
```

f. This is the screenshot:



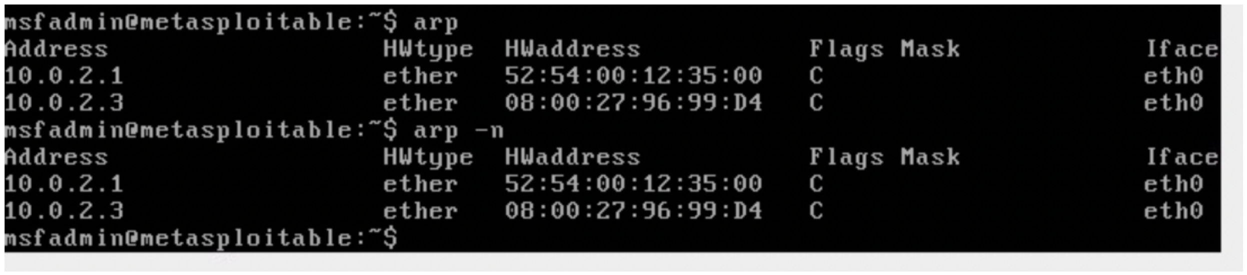
```
Destination     Gateway         Genmask         Flags   MSS Window  irtt If
ace
0.0.0.0         10.0.2.1        0.0.0.0         UG        0 0          0 et
h0
10.0.2.0        0.0.0.0         255.255.255.0   U         0 0          0 et
h0

┌──(kali㉿kali)-[~]
└─$ arp
Address                  HWtype  HWaddress           Flags Mask
Iface
10.0.2.1                 ether   52:54:00:12:35:00   C
eth0
10.0.2.3                 ether   08:00:27:96:99:d4   C
eth0

┌──(kali㉿kali)-[~]
└─$ arp -n
Address                  HWtype  HWaddress           Flags Mask
Iface
10.0.2.1                 ether   52:54:00:12:35:00   C
eth0
10.0.2.3                 ether   08:00:27:96:99:d4   C
eth0

┌──(kali㉿kali)-[~]
└─$ 
```

g. This is the screenshot:



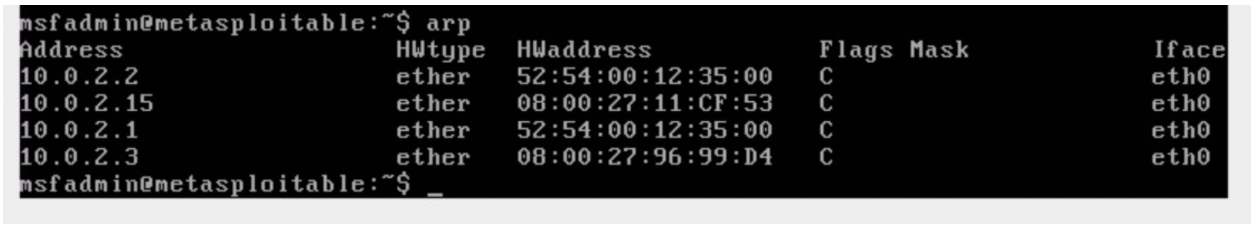
```
          TX packets:72 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6824 (6.6 KB)  TX bytes:8113 (7.9 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:135 errors:0 dropped:0 overruns:0 frame:0
          TX packets:135 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:40109 (39.1 KB)  TX bytes:40109 (39.1 KB)

msfadmin@metasploitable:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.0.2.0        *               255.255.255.0   U         0 0          0 eth0
default         10.0.2.1        0.0.0.0         UG        0 0          0 eth0
msfadmin@metasploitable:~$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.0.2.0        0.0.0.0         255.255.255.0   U         0 0          0 eth0
0.0.0.0         10.0.2.1        0.0.0.0         UG        0 0          0 eth0
msfadmin@metasploitable:~$ _
```

h. This is the screenshot:



```
msfadmin@metasploitable:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.2.1                 ether   52:54:00:12:35:00   C                     eth0
10.0.2.3                 ether   08:00:27:96:99:D4   C                     eth0
msfadmin@metasploitable:~$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.2.1                 ether   52:54:00:12:35:00   C                     eth0
10.0.2.3                 ether   08:00:27:96:99:D4   C                     eth0
msfadmin@metasploitable:~$
```

i. 52:54:00:12:35:00. 10.0.2.1 is listed in the routing table as a gateway for the default case when we don't have the destination IP in the routing table already. Then in the ARP cache, we see that 10.02.01 is paired with the MAC address that we listed as our answer.
j. We do see the HTTP response on Metasploitable. Do not see any captured packets in Wireshark on Kali.
k. Completed
l. This is the screenshot:



```
msfadmin@metasploitable:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.2.2                 ether   52:54:00:12:35:00   C                     eth0
10.0.2.15                ether   08:00:27:11:CF:53   C                     eth0
10.0.2.1                 ether   52:54:00:12:35:00   C                     eth0
10.0.2.3                 ether   08:00:27:96:99:D4   C                     eth0
msfadmin@metasploitable:~$ _
```

The IP addresses 10.0.2.2 and 10.0.2.15 are added, where 10.0.2.2 has the same MAC address as 10.0.2.1 and 10.0.2.15 with the correct MAC address for the Kali machine.

m. Metasploitable will still send the TCP SYN packet to the same MAC address as before: 52:54:00:12:35:00. However, there are now two IP addresses in the Metasploitable ARP cache with that MAC address associated. We looked at the routing table - it still said to direct the packet to the MAC address for 10.0.2.1.
n. Completed
o. We still see the HTTP response on Metasploitable. We do see captured packets in Wireshark - we can see the whole exchange between the Metasploitable machine and the website, including the HTTP response, with the packets labeled such that we can clearly see the intended source, destinations, and contents of the packets.
p. Kali repeatedly sends out messages to Metasploitable claiming that its MAC address is that which Metasploitable might want to request. Metasploitable then sends its packets to the Kali machine, as it believes that this is the proper destination. For this exercise, Kali completes each request and then sends the results back to Metasploitable. This retransmission of Metasploitable's requests explains why we see "TCP Retransmission" in the info tab in Wireshark.
q. A starting point for a spoofing detector would be to notice when two distinct IP addresses claim to have the same MAC address. There might be quite legitimate reasons for two IP addresses having the same MAC address, so there could be frequent false positives with that criteria. We might also be suspicious that ARP spoofing is occurring if we frequently receive ARP replies we never requested. It might be possible to add an encryption / certificate layer to the system that would make ARP spoofing more difficult or make validating correct MAC addresses easier.