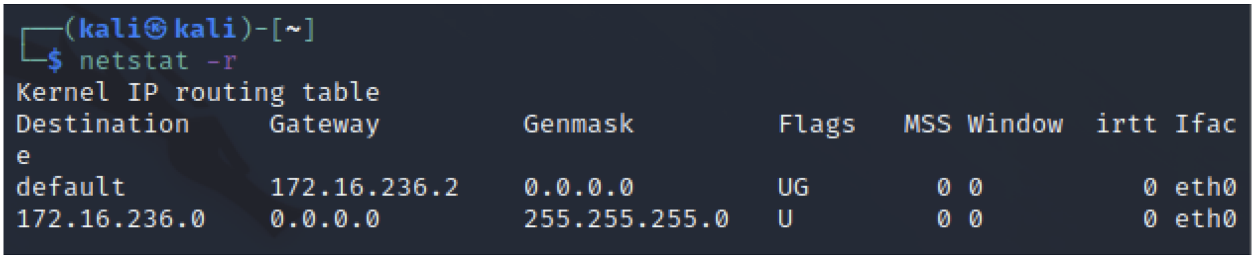
Responses

a) 00:0c:29:b3:59:dd

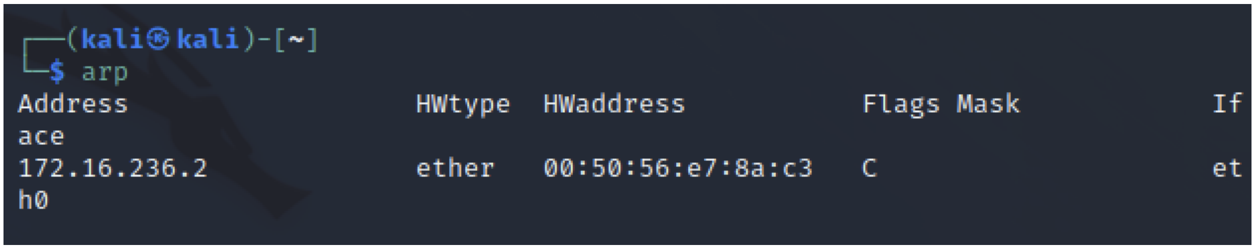
b) 172.16.236.128

c) 00:0c:29:46:e5:0f

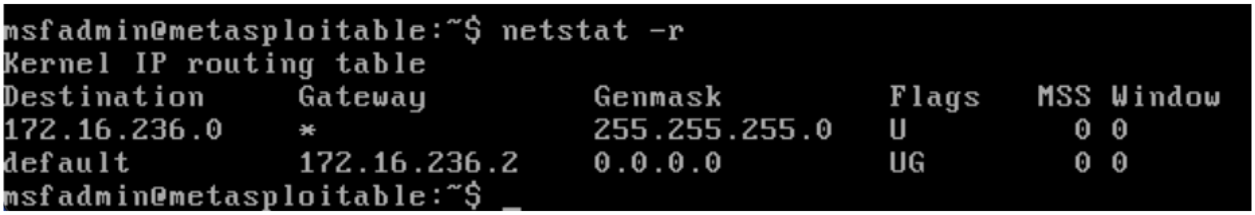
d) 172.16.236.129

e) 

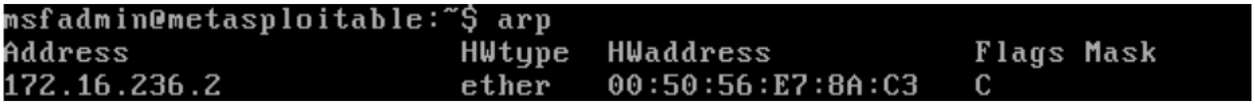
```
┌──(kali㉿kali)-[~]
└─$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Ifac
e
default         172.16.236.2    0.0.0.0         UG        0 0          0 eth0
172.16.236.0    0.0.0.0         255.255.255.0   U         0 0          0 eth0
```

f) 

```
┌──(kali㉿kali)-[~]
└─$ arp
Address                  HWtype  HWaddress           Flags Mask            If
ace
172.16.236.2             ether   00:50:56:e7:8a:c3   C                     et
h0
```

g) 

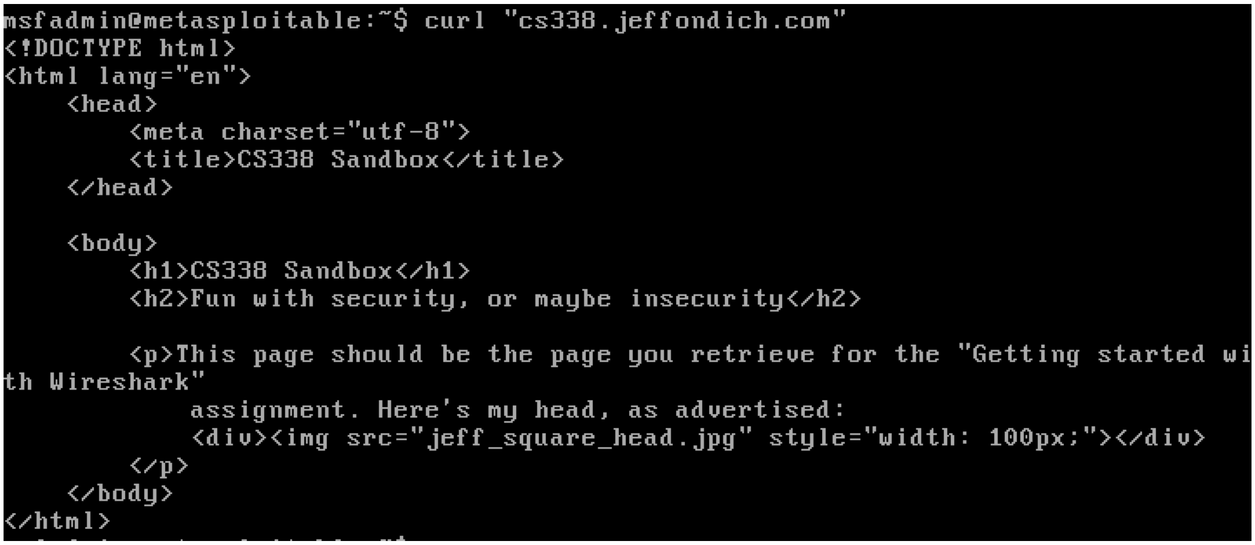
```
msfadmin@metasploitable:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window
172.16.236.0    *               255.255.255.0   U         0 0
default         172.16.236.2    0.0.0.0         UG        0 0
msfadmin@metasploitable:~$ _
```

h) 

```
msfadmin@metasploitable:~$ arp
Address                  HWtype  HWaddress           Flags Mask
172.16.236.2             ether   00:50:56:E7:8A:C3   C
```

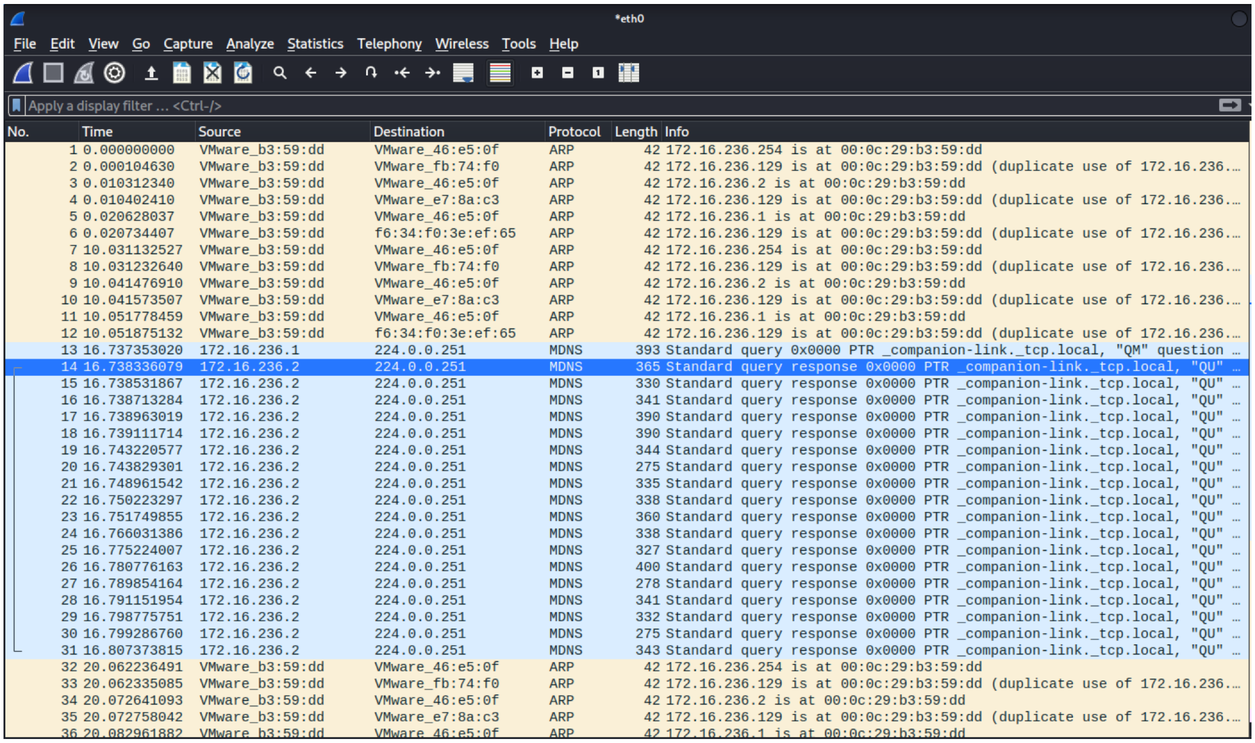
i) The MAC address we are sending our TCP SYN packet to is 00:50:56:E0:CF:B1. This MAC address is associated with the IP address 172.16.64.2. We identified the aforementioned IP address by looking up the IP address of cs338.jeffondich.com using nslookup and identifying 172.16.64.2 as the first hop for our packets towards the final destination of the jeffondich server (we confirmed this by identifying 172.16.64.2 as a gateway on our routing table).

j) While we received an HTTP response on Metasploitable, we did not capture any packets with Wireshark.



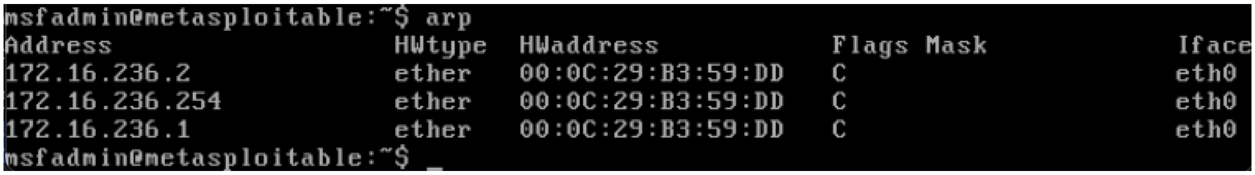
```
msfadmin@metasploitable:~$ curl "cs338.jeffondich.com"
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>CS338 Sandbox</title>
    </head>

    <body>
        <h1>CS338 Sandbox</h1>
        <h2>Fun with security, or maybe insecurity</h2>

        <p>This page should be the page you retrieve for the "Getting started wi
th Wireshark"
            assignment. Here's my head, as advertised:
            <div><img src="jeff_square_head.jpg" style="width: 100px;"></div>
        </p>
    </body>
</html>
```

k)



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	VMware_b3:59:dd	VMware_46:e5:0f	ARP	42	172.16.236.254 is at 00:0c:29:b3:59:dd
2	0.000104630	VMware_b3:59:dd	VMware_fb:74:f0	ARP	42	172.16.236.129 is at 00:0c:29:b3:59:dd (duplicate use of 172.16.236.…
3	0.010312340	VMware_b3:59:dd	VMware_46:e5:0f	ARP	42	172.16.236.2 is at 00:0c:29:b3:59:dd
4	0.010402410	VMware_b3:59:dd	VMware_e7:8a:c3	ARP	42	172.16.236.129 is at 00:0c:29:b3:59:dd (duplicate use of 172.16.236.…
5	0.020628037	VMware_b3:59:dd	VMware_46:e5:0f	ARP	42	172.16.236.1 is at 00:0c:29:b3:59:dd
6	0.020734407	VMware_b3:59:dd	f6:34:f0:3e:ef:65	ARP	42	172.16.236.129 is at 00:0c:29:b3:59:dd (duplicate use of 172.16.236.…
7	10.031132527	VMware_b3:59:dd	VMware_46:e5:0f	ARP	42	172.16.236.254 is at 00:0c:29:b3:59:dd
8	10.031232640	VMware_b3:59:dd	VMware_fb:74:f0	ARP	42	172.16.236.129 is at 00:0c:29:b3:59:dd (duplicate use of 172.16.236.…
9	10.041476910	VMware_b3:59:dd	VMware_46:e5:0f	ARP	42	172.16.236.2 is at 00:0c:29:b3:59:dd
10	10.041573507	VMware_b3:59:dd	VMware_e7:8a:c3	ARP	42	172.16.236.129 is at 00:0c:29:b3:59:dd (duplicate use of 172.16.236.…
11	10.051778459	VMware_b3:59:dd	VMware_46:e5:0f	ARP	42	172.16.236.1 is at 00:0c:29:b3:59:dd
12	10.051875132	VMware_b3:59:dd	f6:34:f0:3e:ef:65	ARP	42	172.16.236.129 is at 00:0c:29:b3:59:dd (duplicate use of 172.16.236.…
13	16.737353020	172.16.236.1	224.0.0.251	MDNS	393	Standard query 0x0000 PTR _companion-link._tcp.local, "QM" question …
14	16.738336079	172.16.236.2	224.0.0.251	MDNS	365	Standard query response 0x0000 PTR _companion-link._tcp.local, "QU" …
15	16.738531867	172.16.236.2	224.0.0.251	MDNS	330	Standard query response 0x0000 PTR _companion-link._tcp.local, "QU" …
16	16.738713284	172.16.236.2	224.0.0.251	MDNS	341	Standard query response 0x0000 PTR _companion-link._tcp.local, "QU" …
17	16.738963019	172.16.236.2	224.0.0.251	MDNS	390	Standard query response 0x0000 PTR _companion-link._tcp.local, "QU" …
18	16.739111714	172.16.236.2	224.0.0.251	MDNS	390	Standard query response 0x0000 PTR _companion-link._tcp.local, "QU" …
19	16.743220577	172.16.236.2	224.0.0.251	MDNS	344	Standard query response 0x0000 PTR _companion-link._tcp.local, "QU" …
20	16.743829301	172.16.236.2	224.0.0.251	MDNS	275	Standard query response 0x0000 PTR _companion-link._tcp.local, "QU" …
21	16.748961542	172.16.236.2	224.0.0.251	MDNS	335	Standard query response 0x0000 PTR _companion-link._tcp.local, "QU" …
22	16.750223297	172.16.236.2	224.0.0.251	MDNS	338	Standard query response 0x0000 PTR _companion-link._tcp.local, "QU" …
23	16.751749855	172.16.236.2	224.0.0.251	MDNS	360	Standard query response 0x0000 PTR _companion-link._tcp.local, "QU" …
24	16.766031386	172.16.236.2	224.0.0.251	MDNS	338	Standard query response 0x0000 PTR _companion-link._tcp.local, "QU" …
25	16.775224007	172.16.236.2	224.0.0.251	MDNS	327	Standard query response 0x0000 PTR _companion-link._tcp.local, "QU" …
26	16.780776163	172.16.236.2	224.0.0.251	MDNS	400	Standard query response 0x0000 PTR _companion-link._tcp.local, "QU" …
27	16.789854164	172.16.236.2	224.0.0.251	MDNS	278	Standard query response 0x0000 PTR _companion-link._tcp.local, "QU" …
28	16.791151954	172.16.236.2	224.0.0.251	MDNS	341	Standard query response 0x0000 PTR _companion-link._tcp.local, "QU" …
29	16.798775751	172.16.236.2	224.0.0.251	MDNS	332	Standard query response 0x0000 PTR _companion-link._tcp.local, "QU" …
30	16.799286760	172.16.236.2	224.0.0.251	MDNS	275	Standard query response 0x0000 PTR _companion-link._tcp.local, "QU" …
31	16.807373815	172.16.236.2	224.0.0.251	MDNS	343	Standard query response 0x0000 PTR _companion-link._tcp.local, "QU" …
32	20.062236491	VMware_b3:59:dd	VMware_46:e5:0f	ARP	42	172.16.236.254 is at 00:0c:29:b3:59:dd
33	20.062335085	VMware_b3:59:dd	VMware_fb:74:f0	ARP	42	172.16.236.129 is at 00:0c:29:b3:59:dd (duplicate use of 172.16.236.…
34	20.072641093	VMware_b3:59:dd	VMware_46:e5:0f	ARP	42	172.16.236.2 is at 00:0c:29:b3:59:dd
35	20.072758042	VMware_b3:59:dd	VMware_e7:8a:c3	ARP	42	172.16.236.129 is at 00:0c:29:b3:59:dd (duplicate use of 172.16.236.…
36	20.082961882	VMware_b3:59:dd	VMware_46:e5:0f	ARP	42	172.16.236.1 is at 00:0c:29:b3:59:dd

l) Two new IP addresses were added with identical MAC addresses

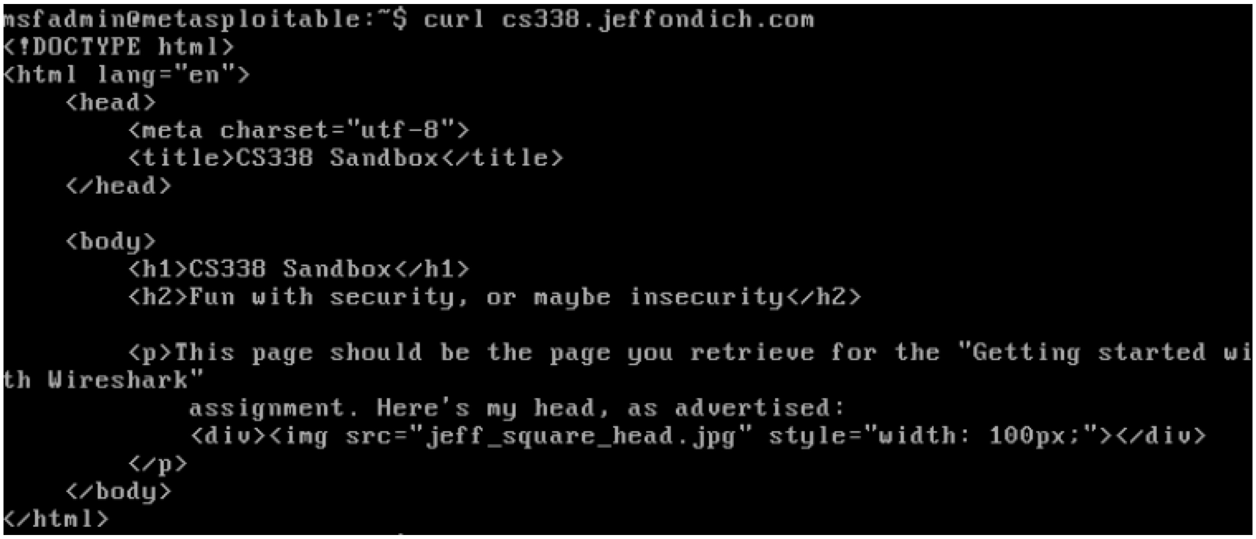


```
msfadmin@metasploitable:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
172.16.236.2             ether   00:0C:29:B3:59:DD   C                     eth0
172.16.236.254           ether   00:0C:29:B3:59:DD   C                     eth0
172.16.236.1             ether   00:0C:29:B3:59:DD   C                     eth0
msfadmin@metasploitable:~$ _
```

m) We believe that the MAC address that Metasploitable will now send the TCP SYN packet to will be Kali's MAC address. This is because Ettercap is now intercepting and reading communication intended for the gateway with Kali's MAC address, before then sending the communication onto the gateway.

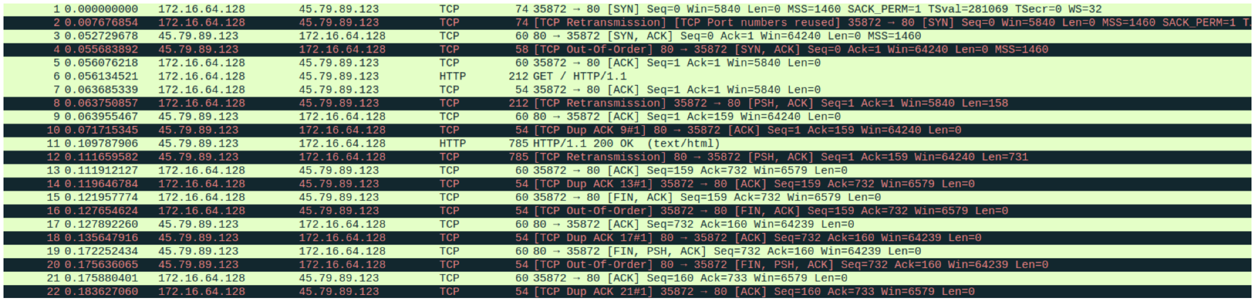
n) Done :smileyface:

o) As indicated in the screenshot below, we received an identical HTTP response on Metasploitable to the one before.



```
msfadmin@metasploitable:~$ curl cs338.jeffondich.com
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>CS338 Sandbox</title>
    </head>

    <body>
        <h1>CS338 Sandbox</h1>
        <h2>Fun with security, or maybe insecurity</h2>

        <p>This page should be the page you retrieve for the "Getting started wi
th Wireshark"
            assignment. Here's my head, as advertised:
            <div><img src="jeff_square_head.jpg" style="width: 100px;"></div>
        </p>
    </body>
</html>
```

However, we are now able to read the packets (both TCP and HTTP) being sent between Metasploitable and the Jeff domain. We can see that a TCP handshake was established and an HTTP GET request was made and fulfilled.



```
 1 0.000000000   172.16.64.128    45.79.89.123     TCP        74 35872 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=281069 TSecr=0 WS=32
 2 0.007676854   172.16.64.128    45.79.89.123     TCP        74 [TCP Retransmission] [TCP Port numbers reused] 35872 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 T
 3 0.052729678   45.79.89.123     172.16.64.128    TCP        60 80 → 35872 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
 4 0.055683892   45.79.89.123     172.16.64.128    TCP        58 [TCP Out-Of-Order] 80 → 35872 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
 5 0.056076218   172.16.64.128    45.79.89.123     TCP        60 35872 → 80 [ACK] Seq=1 Ack=1 Win=5840 Len=0
 6 0.056134521   172.16.64.128    45.79.89.123     HTTP      212 GET / HTTP/1.1
 7 0.063685339   172.16.64.128    45.79.89.123     TCP        54 35872 → 80 [ACK] Seq=1 Ack=1 Win=5840 Len=0
 8 0.063750857   172.16.64.128    45.79.89.123     TCP       212 [TCP Retransmission] 35872 → 80 [PSH, ACK] Seq=1 Ack=1 Win=5840 Len=158
 9 0.063955467   45.79.89.123     172.16.64.128    TCP        60 80 → 35872 [ACK] Seq=1 Ack=159 Win=64240 Len=0
10 0.071715345   45.79.89.123     172.16.64.128    TCP        54 [TCP Dup ACK 9#1] 80 → 35872 [ACK] Seq=1 Ack=159 Win=64240 Len=0
11 0.109787906   45.79.89.123     172.16.64.128    HTTP      785 HTTP/1.1 200 OK  (text/html)
12 0.111659582   45.79.89.123     172.16.64.128    TCP       785 [TCP Retransmission] 80 → 35872 [PSH, ACK] Seq=1 Ack=159 Win=64240 Len=731
13 0.111912127   172.16.64.128    45.79.89.123     TCP        60 35872 → 80 [ACK] Seq=159 Ack=732 Win=6579 Len=0
14 0.119646784   172.16.64.128    45.79.89.123     TCP        54 [TCP Dup ACK 13#1] 35872 → 80 [ACK] Seq=159 Ack=732 Win=6579 Len=0
15 0.121957774   172.16.64.128    45.79.89.123     TCP        60 35872 → 80 [FIN, ACK] Seq=159 Ack=732 Win=6579 Len=0
16 0.127654624   172.16.64.128    45.79.89.123     TCP        54 [TCP Out-Of-Order] 35872 → 80 [FIN, ACK] Seq=159 Ack=732 Win=6579 Len=0
17 0.127892260   45.79.89.123     172.16.64.128    TCP        60 80 → 35872 [ACK] Seq=732 Ack=160 Win=64239 Len=0
18 0.135647916   45.79.89.123     172.16.64.128    TCP        54 [TCP Dup ACK 17#1] 80 → 35872 [ACK] Seq=732 Ack=160 Win=64239 Len=0
19 0.172252434   45.79.89.123     172.16.64.128    TCP        60 80 → 35872 [FIN, PSH, ACK] Seq=732 Ack=160 Win=64239 Len=0
20 0.175636065   45.79.89.123     172.16.64.128    TCP        54 [TCP Out-Of-Order] 80 → 35872 [FIN, PSH, ACK] Seq=732 Ack=160 Win=64239 Len=0
21 0.175880401   172.16.64.128    45.79.89.123     TCP        60 35872 → 80 [ACK] Seq=160 Ack=733 Win=6579 Len=0
22 0.183627060   172.16.64.128    45.79.89.123     TCP        54 [TCP Dup ACK 21#1] 35872 → 80 [ACK] Seq=160 Ack=733 Win=6579 Len=0
```

p) Ettercap says that its own MAC address is associated with the IP address that Metasploitable intends to send its packets to; this causes the MAC address associated with the gateway's IP address to change in Metasploitable's ARP cache. Because of this, Metasploitable sends its packets to Ettercap (Kali), instead of the actual gateway. Ettercap then facilitates communication between Metasploitable and the gateway by first receiving all the packets coming and going from Metasploitable.

q) One way of detecting possible ARP poisoning is checking the ARP cache to see if there are multiple IP addresses associated with one MAC address. This may generate false positives however if there are multiple physical devices acting as the gateway, each with their own IP addresses, while sharing the same MAC address.