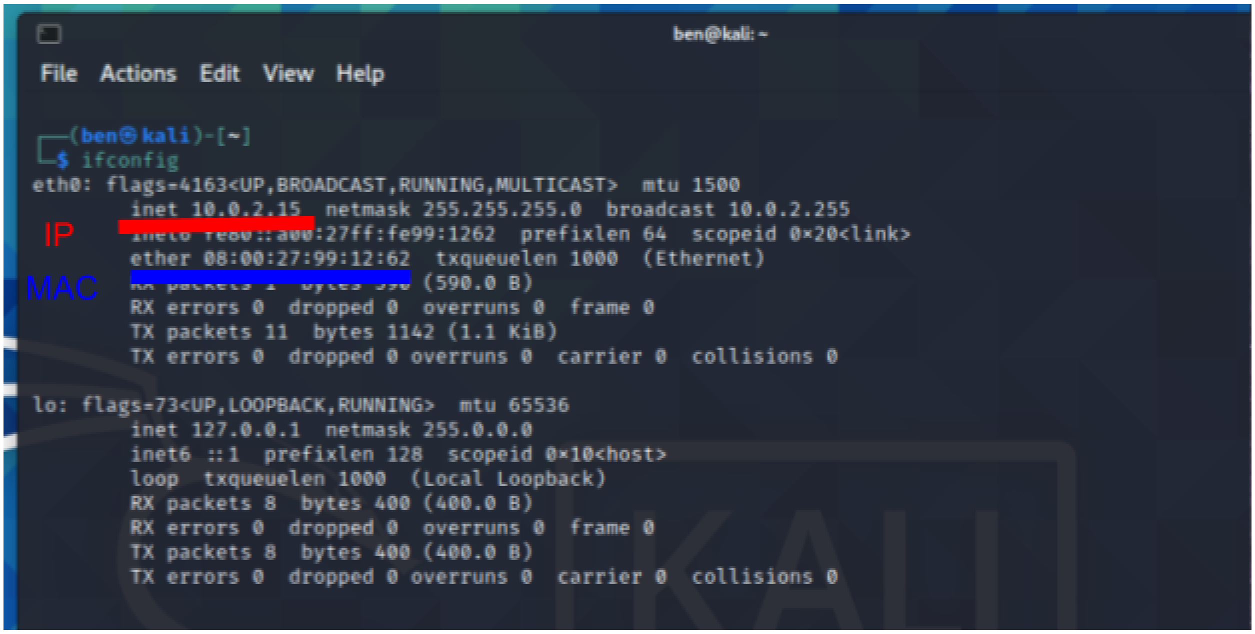
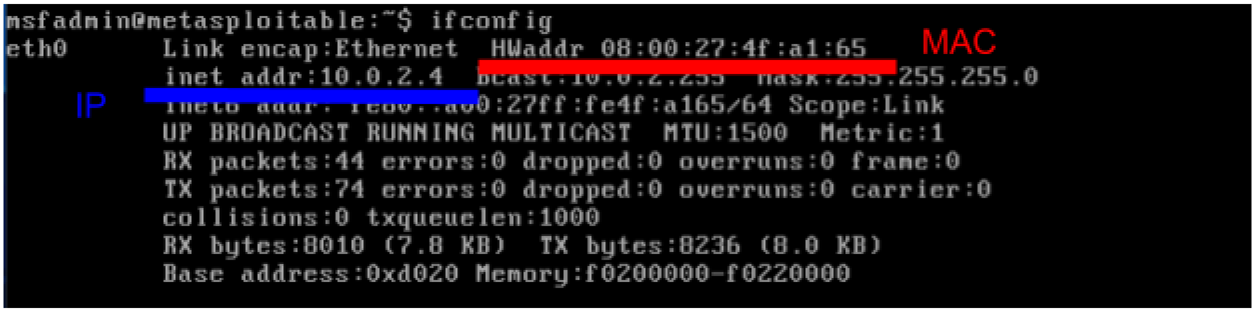
Person-in-the-middle via ARP
Ben Aoki-Sherwood, Kenyon Nystrom, Avery Watts

a. The main interface’s (eth0) MAC (ethernet) address is 08:00:27:99:12:62 for this Kali machine.
b. The Kali main interface’s IP (inet) address is 10.0.2.15.



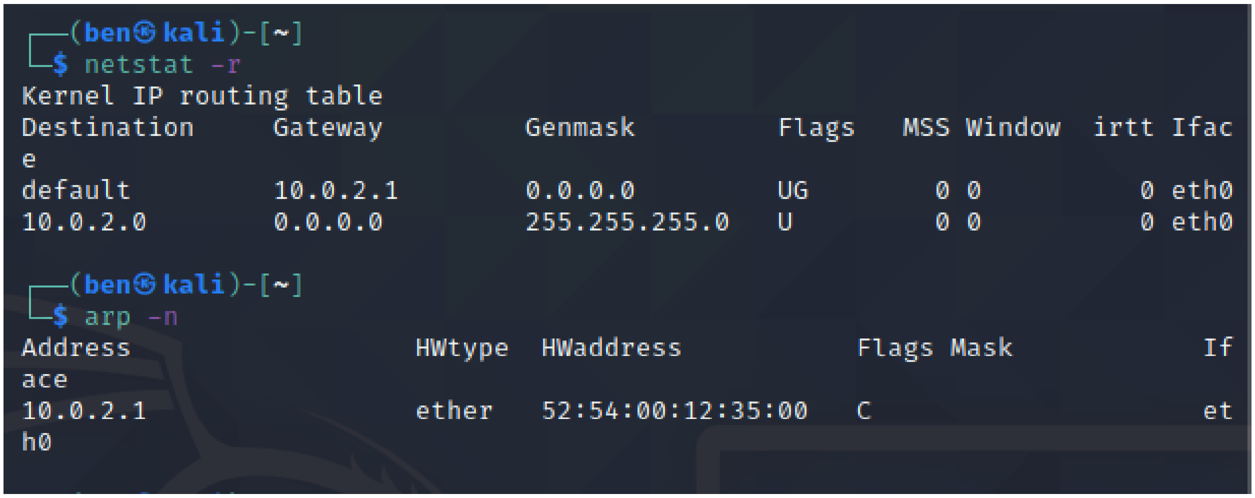
```
┌──(ben㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe99:1262  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:99:12:62  txqueuelen 1000  (Ethernet)
        RX packets 1  bytes 590 (590.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 11  bytes 1142 (1.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 400 (400.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 400 (400.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

c. The Metasploitable main interface MAC (hardware) address is 08:00:27:4f:a1:65
d. The Metasploitable main interface IP (inet) address is 10.0.2.15. This is the same as Kali.



```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:4f:a1:65
          inet addr:10.0.2.4  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe4f:a165/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:44 errors:0 dropped:0 overruns:0 frame:0
          TX packets:74 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8010 (7.8 KB)  TX bytes:8236 (8.0 KB)
          Base address:0xd020 Memory:f0200000-f0220000
```

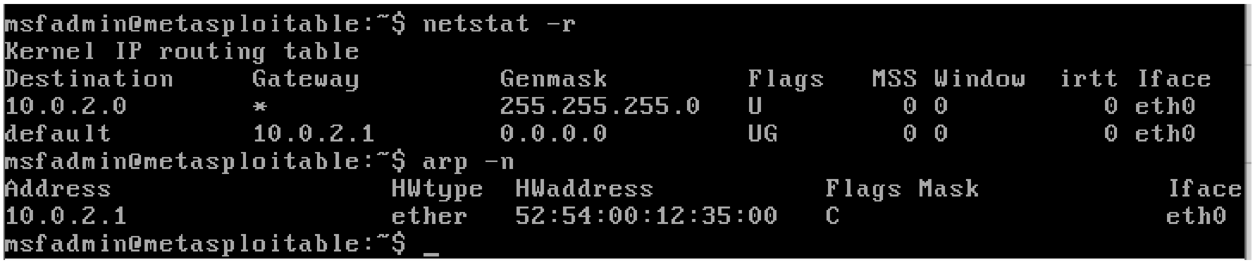
e. Kali Netstat, and
f. Kali arp:



```
┌──(ben㉿kali)-[~]
└─$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Ifac
e
default         10.0.2.1        0.0.0.0         UG        0 0          0 eth0
10.0.2.0        0.0.0.0         255.255.255.0   U         0 0          0 eth0

┌──(ben㉿kali)-[~]
└─$ arp -n
Address                  HWtype  HWaddress           Flags Mask            If
ace
10.0.2.1                 ether   52:54:00:12:35:00   C                     et
h0
```

g. Metasploitable Netstat, and

h. Metasploitable arp:



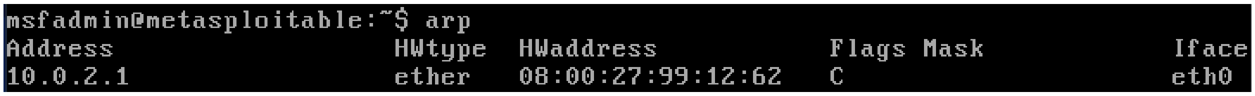
```
msfadmin@metasploitable:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.0.2.0        *               255.255.255.0   U         0 0          0 eth0
default         10.0.2.1        0.0.0.0         UG        0 0          0 eth0
msfadmin@metasploitable:~$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.2.1                 ether   52:54:00:12:35:00   C                     eth0
msfadmin@metasploitable:~$ _
```

i. The gateway address is the router address, which we saw was 10.0.2.2 using netstat. We can see that the hardware/MAC address for this IP address is the only entry in the arp table: the address is 52:54:00:12:35:02.

j. Wireshark didn’t intercept any packets from tcp port http, but Metasploitable did get an HTTP response containing the html content of cs231.jeffondich.com. This is because Wireshark captures from it’s main eth0 interface, which is different than the Metasploitable main eth0 interface: we know this because they have different MAC addresses.

k. Success!

l. The spoofed Metasploitable arp cache is shown below. The gateway/router IP address is the same on the left side, but the MAC address has changed from the MAC address of the router to the MAC address of the Kali main interface (this MAC is the same as the one we found in part a).

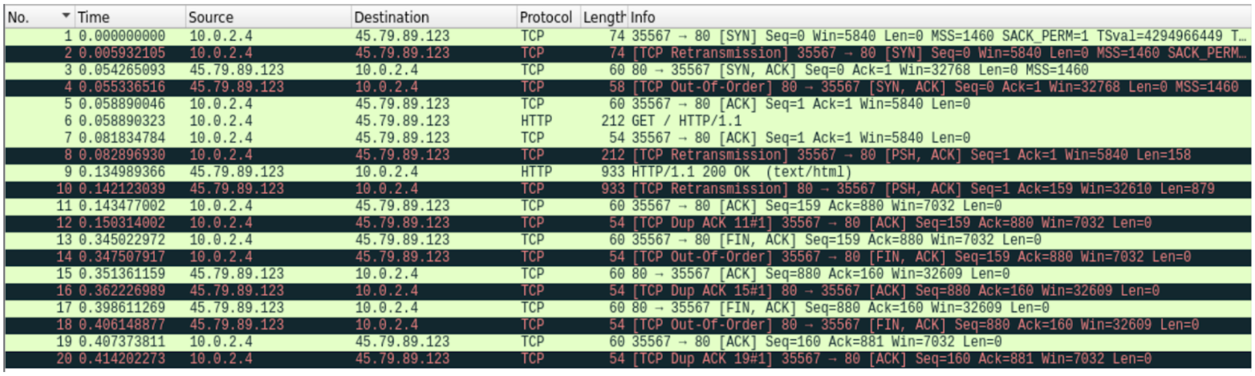


```
msfadmin@metasploitable:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.2.1                 ether   08:00:27:99:12:62   C                     eth0
```

m. Because Metasploitable now thinks that the router MAC address is 08:00:27:99:12:62 (which is actually Kali’s MAC address), it will send the TCP SYN packet to this address.

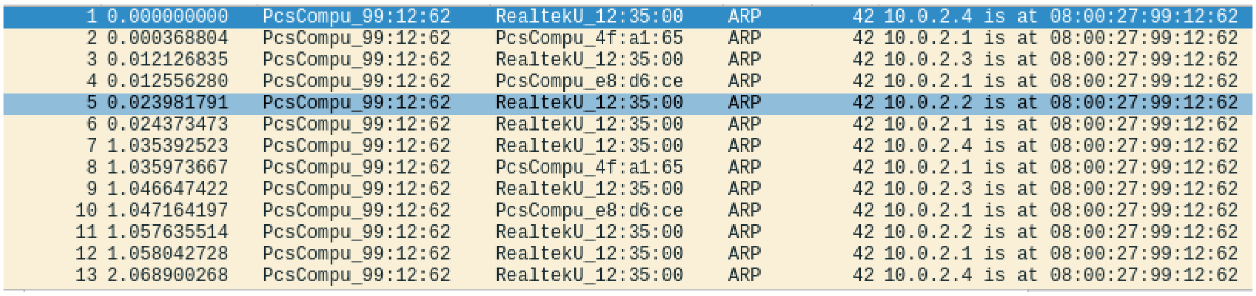
n. Success!

o. We did see an HTTP response on Metasploitable. This was captured in Wireshark, and all the individual information is easily retrievable there. We can easily see the TCP handshake between Metasploitable and the server, and also an HTTP GET request from Metasploitable with a 200 OK response from the server.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.4	45.79.89.123	TCP	74	35567 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=4294966449 T…
2	0.005932105	10.0.2.4	45.79.89.123	TCP	74	[TCP Retransmission] 35567 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM…
3	0.054265093	45.79.89.123	10.0.2.4	TCP	60	80 → 35567 [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0 MSS=1460
4	0.055336516	45.79.89.123	10.0.2.4	TCP	58	[TCP Out-Of-Order] 80 → 35567 [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0 MSS=1460
5	0.058890046	10.0.2.4	45.79.89.123	TCP	60	35567 → 80 [ACK] Seq=1 Ack=1 Win=5840 Len=0
6	0.058890323	10.0.2.4	45.79.89.123	HTTP	212	GET / HTTP/1.1
7	0.081834784	10.0.2.4	45.79.89.123	TCP	54	35567 → 80 [ACK] Seq=1 Ack=1 Win=5840 Len=0
8	0.082896930	10.0.2.4	45.79.89.123	TCP	212	[TCP Retransmission] 35567 → 80 [PSH, ACK] Seq=1 Ack=1 Win=5840 Len=158
9	0.134989366	45.79.89.123	10.0.2.4	HTTP	933	HTTP/1.1 200 OK (text/html)
10	0.142123039	45.79.89.123	10.0.2.4	TCP	933	[TCP Retransmission] 80 → 35567 [PSH, ACK] Seq=1 Ack=159 Win=32610 Len=879
11	0.143477002	10.0.2.4	45.79.89.123	TCP	60	35567 → 80 [ACK] Seq=159 Ack=880 Win=7032 Len=0
12	0.150314002	10.0.2.4	45.79.89.123	TCP	54	[TCP Dup ACK 11#1] 35567 → 80 [ACK] Seq=159 Ack=880 Win=7032 Len=0
13	0.345022972	10.0.2.4	45.79.89.123	TCP	60	35567 → 80 [FIN, ACK] Seq=159 Ack=880 Win=7032 Len=0
14	0.347507917	10.0.2.4	45.79.89.123	TCP	54	[TCP Out-Of-Order] 35567 → 80 [FIN, ACK] Seq=159 Ack=880 Win=7032 Len=0
15	0.351361159	45.79.89.123	10.0.2.4	TCP	60	80 → 35567 [ACK] Seq=880 Ack=160 Win=32609 Len=0
16	0.362226989	45.79.89.123	10.0.2.4	TCP	54	[TCP Dup ACK 15#1] 80 → 35567 [ACK] Seq=880 Ack=160 Win=32609 Len=0
17	0.398611269	45.79.89.123	10.0.2.4	TCP	60	80 → 35567 [FIN, ACK] Seq=880 Ack=160 Win=32609 Len=0
18	0.406148877	45.79.89.123	10.0.2.4	TCP	54	[TCP Out-Of-Order] 80 → 35567 [FIN, ACK] Seq=880 Ack=160 Win=32609 Len=0
19	0.407373811	10.0.2.4	45.79.89.123	TCP	60	35567 → 80 [ACK] Seq=160 Ack=881 Win=7032 Len=0
20	0.414202273	10.0.2.4	45.79.89.123	TCP	54	[TCP Dup ACK 19#1] 35567 → 80 [ACK] Seq=160 Ack=881 Win=7032 Len=0

p. Once the Ettercap ARP poisoning attack started, Kali began repeatedly sending out ARP replies saying that the Ettercap target IP addresses were actually at Kali's own MAC address, instead of the target (router's) MAC address. This prompts Metasploitable to change the MAC associated with its gateway IP in its ARP cache, because Kali is claiming that there is a new MAC associated with this IP (which was the Ettercap target). Basically, Kali is yelling to everybody in the room, "hey, I'm the router now!" and Metasploitable believes Kali, so it changes its cache. Below are the captured ARP packets:



1	0.000000000	PcsCompu_99:12:62	RealtekU_12:35:00	ARP	42	10.0.2.4 is at 08:00:27:99:12:62
2	0.000368804	PcsCompu_99:12:62	PcsCompu_4f:a1:65	ARP	42	10.0.2.1 is at 08:00:27:99:12:62
3	0.012126835	PcsCompu_99:12:62	RealtekU_12:35:00	ARP	42	10.0.2.3 is at 08:00:27:99:12:62
4	0.012556280	PcsCompu_99:12:62	PcsCompu_e8:d6:ce	ARP	42	10.0.2.1 is at 08:00:27:99:12:62
5	0.023981791	PcsCompu_99:12:62	RealtekU_12:35:00	ARP	42	10.0.2.2 is at 08:00:27:99:12:62
6	0.024373473	PcsCompu_99:12:62	RealtekU_12:35:00	ARP	42	10.0.2.1 is at 08:00:27:99:12:62
7	1.035392523	PcsCompu_99:12:62	RealtekU_12:35:00	ARP	42	10.0.2.4 is at 08:00:27:99:12:62
8	1.035973667	PcsCompu_99:12:62	PcsCompu_4f:a1:65	ARP	42	10.0.2.1 is at 08:00:27:99:12:62
9	1.046647422	PcsCompu_99:12:62	RealtekU_12:35:00	ARP	42	10.0.2.3 is at 08:00:27:99:12:62
10	1.047164197	PcsCompu_99:12:62	PcsCompu_e8:d6:ce	ARP	42	10.0.2.1 is at 08:00:27:99:12:62
11	1.057635514	PcsCompu_99:12:62	RealtekU_12:35:00	ARP	42	10.0.2.2 is at 08:00:27:99:12:62
12	1.058042728	PcsCompu_99:12:62	RealtekU_12:35:00	ARP	42	10.0.2.1 is at 08:00:27:99:12:62
13	2.068900268	PcsCompu_99:12:62	RealtekU_12:35:00	ARP	42	10.0.2.4 is at 08:00:27:99:12:62

q. A detector could search for many repeated ARP replies from the same MAC address that do not correspond to any ARP request. This would lead to false positives only if some device is broadcasting that it is now the owner of an existing IP--perhaps this would during the setup process of a new router.