Jack Huffman
CS338 Computer Security
Jeff Ondich

Execution

a. 00:0c:29:7e:8d:ac
b. 192.168.225.128
c. 00:0c:29:8a:f0:f9
d. 192.168.225.129
e.

```
┌──(kali㉿kali)-[~]
└─$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Ifac
e
default         192.168.225.2   0.0.0.0         UG        0 0          0 eth0
192.168.225.0   0.0.0.0         255.255.255.0   U         0 0          0 eth0
```

f.

```
┌──(kali㉿kali)-[~]
└─$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.225.254          ether   00:50:56:e6:64:a9   C                     eth0
192.168.225.2            ether   00:50:56:f1:36:31   C                     eth0
```

g.

```
msfadmin@metasploitable:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.168.225.0   *               255.255.255.0   U         0 0          0 eth0
default         192.168.225.2   0.0.0.0         UG        0 0          0 eth0
msfadmin@metasploitable:~$ _
```

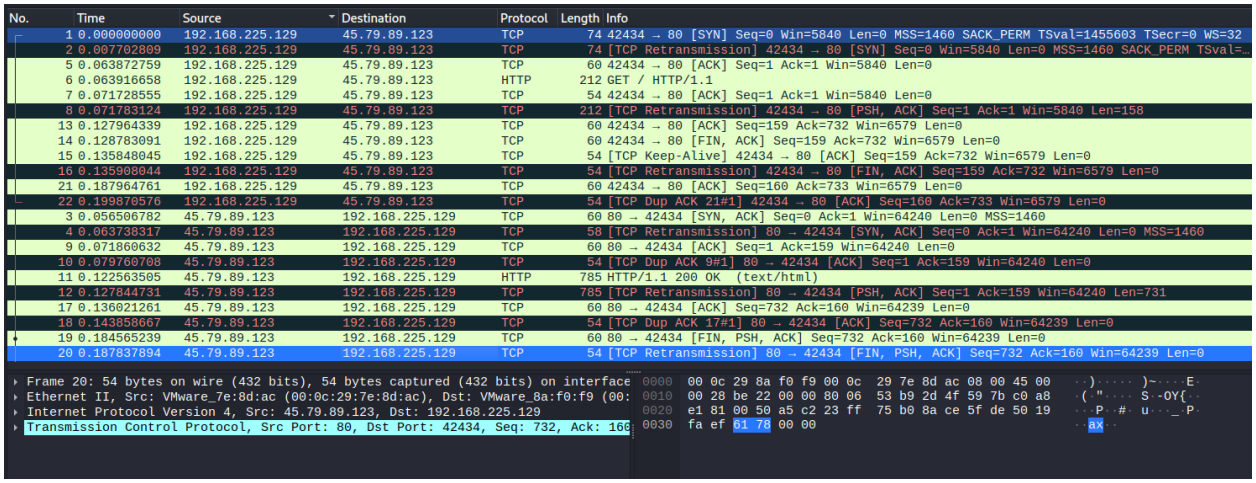
h.

```
msfadmin@metasploitable:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.225.2            ether   00:50:56:F1:36:31   C                     eth0
msfadmin@metasploitable:~$
```

i. Metasploitable would send the packet to 00:50:56:F1:36:31, this is because it is the only MAC address in Metasploitable's arp cache so it would by default send it to this MAC address.
j. On metasploitable I do see the HTML file that is from the CS338 website and on Kali there are 11 captured packets of the communication between metasploitable and cs338
k. Done, more details below (I may have jumped the gun and did the curl command execution early)
l.

```
msfadmin@metasploitable:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.225.2            ether   00:50:56:F1:36:31   C                     eth0
192.168.225.128          ether   00:0C:29:7E:8D:AC   C                     eth0
192.168.225.1            ether   00:50:56:C0:00:08   C                     eth0
192.168.225.254          ether   00:50:56:E6:64:A9   C                     eth0
msfadmin@metasploitable:~$ _
```

m. With ettercap spoofing the ARP cache, I imagine when we execute "curl http://cs338.jeffondich.com/" on metasploitable it will send it to the MAC address of Kali. that is 00:0c:29:7e:8d:ac.

n. 

o. We do see the HTTP response on metasploitable which received the HTML file of the webpage. There are several captured packets in wireshark. We can see in wire shark the [TCP retransmission] packets which is essentially where a packet, intended for cs338.jeffondich was intercepted by Kali and then resent back on its way to cs33.jeffondich.

p. It appears that ettercap was able to poison metasploitable's ARP cache by adding additional MAC addresses so that instead of everything getting sent by default to 00:50:56:F1:36:31 (which is likely the router) it instead adds some additional checking where in our case, the IP address lined up with instructions (in the ARP cache) to send the packet of data to the MAC address of Kali 00:0c:29:7e:8d:ac. Essentially what happened was additional IP to MAC address mappings were added to the ARP cache which derailed the communication to go through Kali.

q. If I was going to make an ARP spoofing detector I would probably have my tool keep track of the ARP table over time and whenever there is a change made to the table, check if the MAC address lines up with the proper IP address. Additionally, it would probably want to check if something like one IP address was mapped to two different MAC addresses (alert for possible spoofing) or but also if the number of ARP requests significantly outweighs the number of ARP replies could be an indicator of ARP spoofing. One scenario where my detector might generate false positives would be in something like real life network changes, such as replacing/reconfiguring specific devices.To help combat things like this, we should allow whitelisting by administrators so that these changes don't generate false positives.

Synthesis

a. Mal's first goal would be to identify who Alice and Bob are. In this case it would be that Alice is metasploitable, Bob is cs33.jeffondich.com and Mal is Kali. Mal would then do

some ARP cache manipulation to Alice's machine, essentially sending forged ARP messages that would associate Mal's MAC address with the IP address of Bob, meaning that data which is intended to be sent to Bob will unknowingly be routed to Mal. While it might make sense that Alice should send all of its data intended for recipients outside it's local network to the address of the router, if Alice has a particularly vulnerable ARP cache it is possible for Mal to simply send these ARP messages telling Alice to update her cache with the forged information. Once this is done, all messages intended for Bob will instead be received by Mal who can choose to edit them and send them to Bob or simply eavesdrop on the contents and send it to Bob.

b. From Alice's perspective she is simply looking at the ARP cache and its associated MAC addresses. If she wasn't able to detect that the ARP cache has been manipulated it would be difficult to detect that there is an issue (so long as Mal sends the proper packets back to Alice so Alice's access to the intended web service isn't denied). Also, ARP spoofing is a relatively silent attack which wouldn't trigger any alerts or change the MAC address of Alice's devices, rather just rerouting their signals. One way that Alice could detect the altering of her ARP cache would be to continuously monitor the ARP cache for changes and inconsistencies.
c. From Bob's perspective the attack is likely not visible. If Bob is simply acting as a web server which responds to requests then it would not have a way of telling that the person contacting them is impersonating someone else unless it had some anti ARP spoofing technology. It's likely that even if Bob receives a request from an incorrect MAC address they will still send a normal response to the request.
d. With ARP spoofing Mal would be able to intercept and eavesdrop on the packets being sent between Alice and Bob, but with a HTTPS protocol in place the data itself would be encrypted with a method such that Mal would not have access to the necessary information to decrypt the message. Mal would likely be able to intercept the initial TCP handshake and determine the IP addresses of Alice and Bob but because of the verification procedures used by HTTPS, the meaningful data will be encrypted when it reaches Mal and Mal won't be able to decrypt it. The use of HTTPS doesn't directly assist in the detection of ARP spoofing however, it just increases the security of the data when it is in transmission.