

ARP Spoofing

Rebecca Fox, Yemi Shin

- A. What is Kali's main interface's MAC address? (The main interface is probably called eth0, but check ifconfig to be sure.)**
- Kali's main interface's MAC address is 08:00:27:11:cf:53
- B. What is Kali's main interface's IP address?**
- Kali's main interface's IP address is 10.0.2.15
- C. What is Metasploitable's main interface's MAC address?**
- Metasploitable's main interface's MAC address is 08:00:27:1a:45:12
- D. What is Metasploitable's main interface's IP address?**
- Metasploitable's main interface's IP address is 10.0.2.4
- E. Show Kali's routing table. (Use "netstat -r" to see it with symbolic names, or "netstat -rn" to see it with numerical addresses.)**

```
(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
```

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

```
(kali㉿kali)-[~]
$ arp
Address             HWtype  HWaddress          Flags Mask
Iface
10.0.2.3            ether   08:00:27:2b:5d:6f   C
eth0
10.0.2.1            ether   52:54:00:12:35:00   C
eth0
```

- G. Show Metasploitable's routing table.**

```
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
```

H. Show Metasploitable's ARP cache.

```
msfadmin@metasploitable:~$ arp
Address      HWtype  HWaddress      Flags Mask    Iface
10.0.2.3     ether   08:00:27:2B:5D:6F  C           eth0
10.0.2.1     ether   52:54:00:12:35:00  C           eth0
msfadmin@metasploitable:~$ _
```

I. 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.

```
msfadmin@metasploitable:~$ nslookup
> http://cs231.jeffondich.com/
Server:      137.22.198.40
Address:     137.22.198.40#53

** server can't find http://cs231.jeffondich.com/: NXDOMAIN
>
```

- It would want to send the request to the MAC address of cs231.jeffondich.com because this is the server that is hosting the website and therefore would be the desired server to initiate TCP handshake with.
- J. 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?**
 - We saw an HTTP response on metasploitable but we did not see any captured packets in Wireshark on Kali.
- K. 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

I'll post some screenshots on Slack of how I got Ettercap to do these things. Honestly, I don't know who redesigned this user interface to make it so much harder to do things, but they did. (Common enough in the Linux UI world.)

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

```
Address      HWtype  HWaddress  Flags Mask  Iface
10.0.2.3     ether   08:00:27:11:CF:53  C           eth0
10.0.2.1     ether   08:00:27:11:CF:53  C           eth0
10.0.2.2     ether   08:00:27:11:CF:53  C           eth0
msfadmin@metasploitable:~$
```

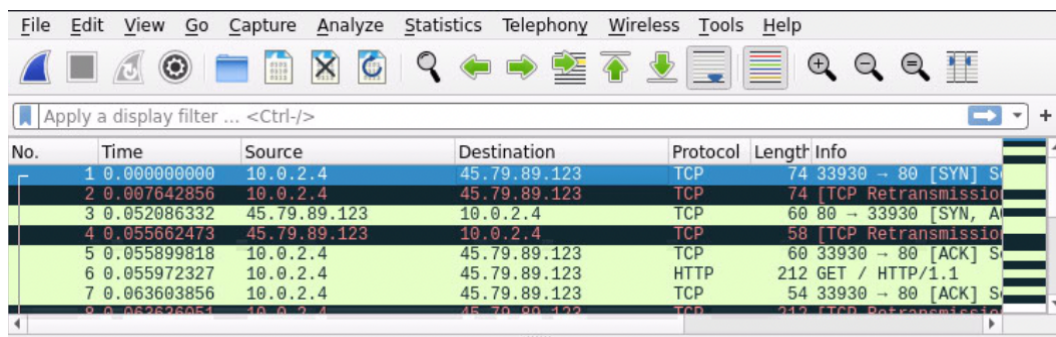
- All the MAC addresses are the same, and they are all directed towards Ettercap's (or Kali's) MAC address (08:00:27:11:cf:53)!

M. If you execute "curl http://cs231.jeffondich.com/" on Metasploitable now, to what MAC address will Metasploitable send the TCP SYN packet? Explain why.

- 08:00:27:11:CF:53. Our ARP spoofing changed the HWaddress of all of the IP address to appear to be the same MAC address so that we can view all of the information sent out by metasploitable

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

O. 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	33930 → 80 [SYN] S
2	0.007642856	10.0.2.4	45.79.89.123	TCP	74	[TCP Retransmission]
3	0.052086332	45.79.89.123	10.0.2.4	TCP	60	80 → 33930 [SYN, A
4	0.055662473	45.79.89.123	10.0.2.4	TCP	58	[TCP Retransmission]
5	0.055899818	10.0.2.4	45.79.89.123	TCP	60	33930 → 80 [ACK] S
6	0.055972327	10.0.2.4	45.79.89.123	HTTP	212	GET / HTTP/1.1
7	0.063603856	10.0.2.4	45.79.89.123	TCP	54	33930 → 80 [ACK] S
8	0.062626051	10.0.2.4	45.79.89.123	TCP	74	[TCP Retransmission]

- We do still see an HTTP response on Metasploitable, but unlike before ARP spoofing we do see the capture packets in wireshark (pictured above). We can also see the requests from metasploitable and cs231.jeffondich.com.

```

DHCP: [08:00:27:1a:45:12] REQUEST 10.0.2.4
DHCP: [10.0.2.3] ACK : 10.0.2.4 255.255.255.0 GW 10.0.2.1 DNS 137.22.198.41 "carleton.edu"
DHCP: [08:00:27:1a:45:12] REQUEST 10.0.2.4
DHCP: [10.0.2.3] ACK : 10.0.2.4 255.255.255.0 GW 10.0.2.1 DNS 137.22.198.41 "carleton.edu"
DHCP: [08:00:27:1a:45:12] REQUEST 10.0.2.4
DHCP: [10.0.2.3] ACK : 10.0.2.4 255.255.255.0 GW 10.0.2.1 DNS 137.22.198.41 "carleton.edu"

```

P. 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.)

- Kali changed Metasploitable's ARP cache so that all of the MAC addresses were the same address that corresponds to ettercap. It does this by altering the MAC address given to metasploitable when they send out ARP requests for their desired IP addresses. This way, when Metasploitable tries to send requests, it gets sent to ettercap who can view what is happening and, if they want to, change it before sending the information back to metasploitable.

Source	Destination	Protocol	Length	Info
csCompu_11:cf:53	PcsCompu_8e:4e:0d	ARP	42	10.0.2.4 is at 08:00:27:11:cf:53 (dupli
csCompu_11:cf:53	PcsCompu_1a:45:12	ARP	42	10.0.2.2 is at 08:00:27:11:cf:53
csCompu_11:cf:53	RealtekU_12:35:00	ARP	42	10.0.2.4 is at 08:00:27:11:cf:53 (dupli
csCompu_11:cf:53	PcsCompu_1a:45:12	ARP	42	10.0.2.1 is at 08:00:27:11:cf:53
csCompu_11:cf:53	RealtekU_12:35:00	ARP	42	10.0.2.4 is at 08:00:27:11:cf:53 (dupli
csCompu_11:cf:53	PcsCompu_1a:45:12	ARP	42	who has 10.0.2.4? Tell 10.0.2.15
csCompu_1a:45:12	PcsCompu_11:cf:53	ARP	60	10.0.2.4 is at 08:00:27:1a:45:12

Q. 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.)

- The ARP spoofing detector would need to find a way to confirm that the sender of the desired MAC address is the desired server, and confirm that the MAC address matches the actual address of the given IP address. It could also check that the MAC addresses for all IP addresses in the arp cache are not the same address. It could possibly use a certificate system to implement this. This may generate a false positive when there are various dependencies to the server system. In one instance, multiple servers might exist in a single machine, in which case they would all have the same MAC address. This would prevent the third party authority to verify which "sub" server is the actual requester among the possibly many servers that share the same MAC address. In another case, not all servers might have the authority to send their own MAC address due to some dependency, and if they rely on a third party to send a MAC address for them, our detector would say this is invalid due to the IP address not matching the sender of the MAC address.