

# Homework 2 - Information Security (ICS344)

Alfaifi, Ammar

## 1 My environment

In my case I have two real machines. The victim machine is an MacOS and the attacker is an Arch OS machine. All machines are connected to my home wireless LAN. Where the victim IP address is 192.168.0.101, attacker's IP address 192.168.0.104, the gateway's IP address 192.168.0.1.

## 2 ARP Poisoning

### 2.1 Normal Operation

For the victim normal operation, I see that any packets going from the its IP address it goes directly to the network gateway, as it supposed to be. See Figure 1

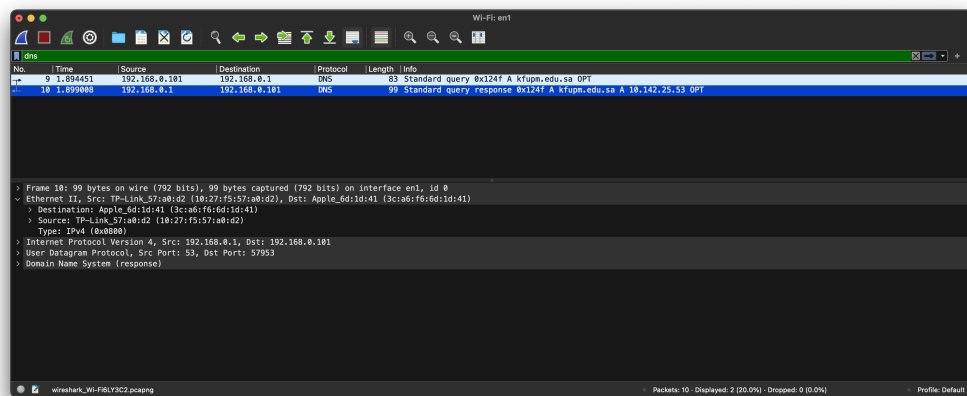


Figure 1: I used `dig kfupm.edu.sa` as test to see where the packet will travel thru. It passes thru the wireless gateway, no MITM exists.

Also, see Figure 2 for the victim's ARP table in normal operation.

```
➔ ~ arp -a
? (192.168.0.1) at 10:27:f5:57:a0:d2 on en1 ifscope [ethernet]
? (192.168.0.100) at 26:86:f6:3b:f9:e7 on en1 ifscope [ethernet]
? (192.168.0.101) at 3c:a6:f6:6d:1d:41 on en1 ifscope permanent [ethernet]
? (192.168.0.102) at 3c:6a:9d:11:51:25 on en1 ifscope [ethernet]
? (192.168.0.103) at f2:3e:7e:84:44:8b on en1 ifscope [ethernet]
? (192.168.0.104) at f4:f:24:1a:13:f1 on en1 ifscope [ethernet]
? (192.168.0.105) at 5e:f:9b:cc:c5:81 on en1 ifscope [ethernet]
```

Figure 2: This is the ARP table for victim's machine in normal operation, notice the gateway IP address as well as that for attacker.

## 3 Launching Attack

### 3.1 Get MAC addresses

To start the attack we need the the corresponding MAC addresses of both the network gateway and victim. I'll use `scapy` in interactive mode to send an ARO request to get their MAC addresses. See Figure 7 for the

commands that I used.

```
>>> arp = Ether(dst=':'.join(['ff']*6)) / ARP(op=1, pdst='192.168.0.101')
>>> res, un = srp(arp)
Begin emission:
Finished sending 1 packets.
..^C
Received 2 packets, got 0 answers, remaining 1 packets
>>> res, un = srp(arp, timeout=2)
Begin emission:
Finished sending 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
>>> res[0][1].hwsrc
'3c:a6:f6:6d:1d:41'
>>> arp = Ether(dst=':'.join(['ff']*6)) / ARP(op=1, pdst='192.168.0.1')
>>> res, un = srp(arp, timeout=2)
Begin emission:
Finished sending 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
>>> res[0][1].hwsrc
'10:27:f5:57:a0:d2'
>>> 
```

Figure 3: This is the ARP table for victim's machine in normal operation, notice the gateway IP address as well as that for attacker.

Now for the victim machine: IP address is 192.168.0.101 and MAC is 3c:a6:f6:6d:1d:41  
for the gateway router: IP address is 192.168.0.1 and MAC is 10:27:f5:57:a0:d2

### 3.2 Spoof ARP Table

Note: In ARP layer, hwsrc and hwdst represent MAC address of source and destination respectively, while psrc and pdst represent the IP address of source and destination respectively. Also I run this command to allow port forwarding from the attacker's machine to the corresponding dst machine: `echo 1 > /proc/sys/net/ipv4/ip_forward`  
See the commands to spoof the victim ARP table Figure 4.

## 4 ARP Table

Let's see the victim's ARP table after attack, in Figure 4.

```
➔ ~ arp -a
? (192.168.0.1) at f4:f:24:1a:13:f1 on en1 ifscope [ethernet]
? (192.168.0.100) at 26:86:f6:3b:f9:e7 on en1 ifscope [ethernet]
? (192.168.0.101) at 3c:a6:f6:6d:1d:41 on en1 ifscope permanent [ethernet]
? (192.168.0.102) at 3c:6a:9d:11:51:25 on en1 ifscope [ethernet]
? (192.168.0.103) at f2:3e:7e:84:44:8b on en1 ifscope [ethernet]
? (192.168.0.104) at f4:f:24:1a:13:f1 on en1 ifscope [ethernet]
```

Figure 4: We can see now in the victim's ARP table the MAC address is the same as that for the attacker machine with an IP 192.168.0.104.

No.	Time	Source	Destination	Protocol	Length	Info
13	5.808589	192.168.0.101	192.168.0.1	DNS	83	Standard query 0xa8b8 A kfupm.edu.sa OPT
15	5.904422	192.168.0.1	192.168.0.101	DNS	99	Standard query response 0xa8b8 A kfupm.edu.sa A
16	6.042986	192.168.0.101	192.168.0.1	DNS	85	Standard query 0x888d HTTPS login.microsoftonline
17	6.043173	192.168.0.101	192.168.0.1	DNS	85	Standard query 0xd6e3 A login.microsoftonline
19	6.077995	192.168.0.1	192.168.0.101	DNS	328	Standard query response 0xd6e3 A login.microsoftonline
33	6.665455	192.168.0.101	192.168.0.1	DNS	82	Standard query 0xddce HTTPS ocsps.edge.digicert.com
34	6.665572	192.168.0.101	192.168.0.1	DNS	82	Standard query 0x3941 A ocsps.edge.digicert.com
37	6.764454	192.168.0.1	192.168.0.101	DNS	163	Standard query response 0x3941 A ocsps.edge.digicert.com
52	7.090053	192.168.0.101	192.168.0.1	DNS	85	Standard query 0x888d HTTPS login.microsoftonline
76	7.665499	192.168.0.101	192.168.0.1	DNS	82	Standard query 0xddce HTTPS ocsps.edge.digicert.com
78	7.740981	192.168.0.101	192.168.0.1	DNS	85	Standard query 0x88ea HTTPS clients.config.office.ne
79	7.741158	192.168.0.101	192.168.0.1	DNS	85	Standard query 0xe71a A clients.config.office.ne
82	7.799274	192.168.0.1	192.168.0.101	DNS	171	Standard query response 0xe71a A clients.config.office
176	8.787684	192.168.0.101	192.168.0.1	DNS	85	Standard query 0x88ea HTTPS clients.config.office
194	8.949410	192.168.0.101	192.168.0.1	DNS	85	Standard query 0x888d HTTPS login.microsoftonline
195	8.949570	192.168.0.101	192.168.0.1	DNS	82	Standard query 0xddce HTTPS ocsps.edge.digicert.com

Frame 13: 83 bytes on wire (664 bits), 83 bytes captured (664 bits) on interface en1, id 0  
 Ethernet II, Src: Apple\_6d:1d:41 (3c:a6:f6:6d:1d:41), Dst: Apple\_1a:13:f1 (f4:0f:24:1a:13:f1)  
 Destination: Apple\_1a:13:f1 (f4:0f:24:1a:13:f1)  
 Source: Apple\_6d:1d:41 (3c:a6:f6:6d:1d:41)  
 Type: IPv4 (0x0800)

Figure 5: We send a dig request from the victim and in wireshark we see indeed the packet goes to the attacker machine first, MITM.

No.	Time	Source	Destination	Protocol	Length	Info
31	6.760064	Broadcast	Broadcast	ARP	42	Who has 192.168.0.101? Tell 192.168.0.104
32	6.760131	Apple_6d:1d:41	Apple_1a:13:f1	ARP	42	192.168.0.101 is at 3c:a6:f6:6d:1d:41
33	6.786250	Apple_1a:13:f1	Apple_6d:1d:41	ARP	42	192.168.0.1 is at f4:0f:24:1a:13:f1

Frame 33: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface en1, id 0  
 Ethernet II, Src: Apple\_1a:13:f1 (f4:0f:24:1a:13:f1), Dst: Apple\_6d:1d:41 (3c:a6:f6:6d:1d:41)  
 Destination: Apple\_6d:1d:41 (3c:a6:f6:6d:1d:41)  
 Source: Apple\_1a:13:f1 (f4:0f:24:1a:13:f1)  
 Type: ARP (0x0806)  
 Address Resolution Protocol (reply)  
 Hardware type: Ethernet (1)  
 Protocol type: IPv4 (0x0800)  
 Hardware size: 6  
 Protocol size: 4  
 Opcode: reply (2)  
 Sender MAC address: Apple\_1a:13:f1 (f4:0f:24:1a:13:f1)  
 Sender IP address: 192.168.0.1  
 Target MAC address: Apple\_6d:1d:41 (3c:a6:f6:6d:1d:41)  
 Target IP address: 192.168.0.101

Figure 6: This is the ARP table for victim's machine in normal operation, notice the gateway IP address as well as that for attacker.

```

~ arp -a
? (192.168.0.1) at 10:27:f5:57:a0:d2 on en1 ifscope [ethernet]
? (192.168.0.100) at 26:86:f6:3b:f9:e7 on en1 ifscope [ethernet]
? (192.168.0.101) at 3c:a6:f6:6d:1d:41 on en1 ifscope permanent [ethernet]
? (192.168.0.102) at 3c:6a:9d:11:51:25 on en1 ifscope [ethernet]
? (192.168.0.103) at f2:3e:7e:84:44:8b on en1 ifscope [ethernet]
? (192.168.0.104) at f4:f:24:1a:13:f1 on en1 ifscope [ethernet]
? (192.168.0.105) at 5e:f:9b:cc:c5:81 on en1 ifscope [ethernet]

```

Figure 7: This is the ARP table for victim's machine in normal operation, notice the gateway IP address as well as that for attacker.