



**YILDIZ TECHNICAL UNIVERSITY**  
**FACULTY OF ELECTRICAL AND ELECTRONICS**  
**SECURITY OF COMPUTER SYSTEMS**  
**(BLM4011)**  
**LAB1 ARP-POISONING LAB REPORT**

19011033 – Erkan Vatan  
20011602 – Berke Özgen  
erkan.vatan@std.yildiz.edu.tr  
berke.ozgen1@std.yildiz.edu.tr

**DEPARTMENT OF COMPUTER ENGINEERING**

## 1 INTRODUCTION

ARP is a protocol that is used to bridge the data link layer and the network layer by matching the MAC addresses to their corresponding IP addresses. IP addresses are used in the network layer to communicate with machines inside and outside of the local network and can change over time. MAC addresses, however, are physically assigned addresses that are unique to each network-connected machine universally. MAC addresses are used in the data link layer to communicate between devices in the same network and are assigned by the manufacturer to never change through its lifetime.

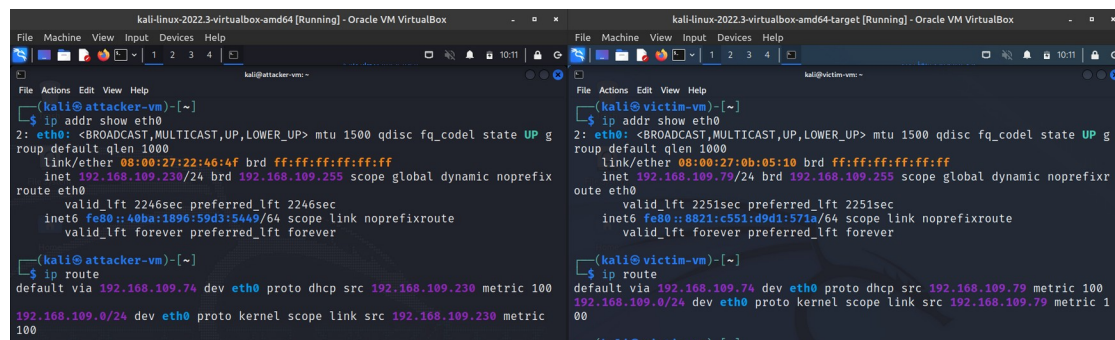
In a local network, when a data packet is being sent from one machine to another by using the receiver machine's known IP address, the sender broadcasts a message to all devices under that subnet to find out the MAC address of the receiver. Once the MAC address is discovered, the sender caches the address in a table called the ARP table. This way the machine can check the table before sending a request, preventing the need to rediscover addresses at every request.

Because of the lack of authentication in ARP, any machine inside the local network can respond to ARP requests. When a machine is trying to discover its recipient machine, it sends an ARP request that can be replied to from anyone in the local network. This vulnerability can be used to send spoofed ARP responses to fill a target's ARP table with fake information. The attackers can gain access to the target machine's packet flow by disguising themselves as the recipient, performing an attack called "man-in-the-middle attack". This kind of attack would need the attacker to poison the receiver's IP address on the sender's ARP table and the sender's IP address on the receiver's ARP table to pair with the attacker's MAC address. A "denial-of-service attack" can also be done by routing every packet to a single MAC address, causing the packets to be dropped.

Due to the nature of how ARP poisoning works, an attack can be identified by checking ARP table of the target before and after the attack. The biggest sign of an attack is when the MAC address for a machine does not match its original address or there are multiple machines with the same MAC address. This can be easily prevented by using a static ARP table where each entry must be entered by hand. This is not very applicable for public networks since IP addresses constantly change inside the network. An ARP poisoning prevention software can also be used inside the network to block any uncertified ARP responses. The most widely implemented method of partly preventing these attacks is encryption. Through the usage of HTTPS and other encrypted protocols, although the attacker can still see the packet flow, its impossible to read the contents of the requests.

## 2 METHOD

- Attacker must be in the same local network as the victim.
- Attacker finds the IP of the router with the command ``ip route`` (Figure 1).
- Attacker uses a spoofing tool like Ettercap to flood the network with forged entries.
- In Ettercap, attacker scans the network for available hosts.
- Attacker selects the router and victim from the host list and sets them as target 1 and target 2. ARP Spoofing attack is started.
- After starting the spoofing attack, the router MAC address in the ARP cache of the victim is updated with the MAC address of the attacker. Due to this, all packets pass through attacker before being sent to the router. That is why this attack type is also called "Man-in-the-Middle attack".
- Attacker can sniff the packets by using a network analyzer tool like Wireshark. Results can be filtered with "`ip.addr == <victim_ip> && http`" filter to reduce the network traffic down to packets coming only from victim.



The image shows two terminal windows from Oracle VM VirtualBox. The left window is titled 'kali-linux-2022.3-virtualbox-amd64 [Running] - Oracle VM VirtualBox' and shows the output of the `ip addr show eth0` and `ip route` commands. The right window is titled 'kali-linux-2022.3-virtualbox-amd64-target [Running] - Oracle VM VirtualBox' and shows the output of the `ip addr show eth0` and `ip route` commands. Both windows show the default gateway as 192.168.109.74.

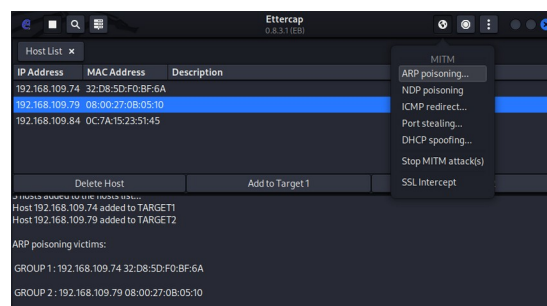
```
(kali@attacker-vm)-[~]
$ ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 08:00:27:22:46:4f brd ff:ff:ff:ff:ff:ff
    inet 192.168.109.230/24 brd 192.168.109.255 scope global dynamic noprefixr
oute eth0
        valid_lft 2246sec preferred_lft 2246sec
        inet6 fe80::40ba:1896:59d3:5449/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

(kali@attacker-vm)-[~]
$ ip route
default via 192.168.109.74 dev eth0 proto dhcp src 192.168.109.230 metric 100
192.168.109.0/24 dev eth0 proto kernel scope link src 192.168.109.230 metric 100

(kali@victim-vm)-[~]
$ ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 08:00:27:0b:05:10 brd ff:ff:ff:ff:ff:ff
    inet 192.168.109.79/24 brd 192.168.109.255 scope global dynamic noprefixr
oute eth0
        valid_lft 2251sec preferred_lft 2251sec
        inet6 fe80::8821:c551:d9d1:571a/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

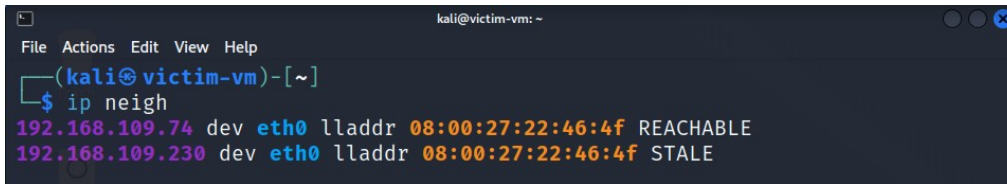
(kali@victim-vm)-[~]
$ ip route
default via 192.168.109.74 dev eth0 proto dhcp src 192.168.109.79 metric 100
192.168.109.0/24 dev eth0 proto kernel scope link src 192.168.109.79 metric 100
```

**Figure 1** IP addresses of attacker and victim virtual machines are discovered (attacker: 192.168.109.230, victim: 192.168.109.79). Using the ``ip route`` command, it is checked that they are connected to the same router (default: 192.168.109.74).



**Figure 2** Attacker starts the Ettercap program. After scanning the hosts in the network, router is selected as Target 1 and victim is selected as Target 2. And the ARP Poisoning attack is started.

### 3 RESULTS



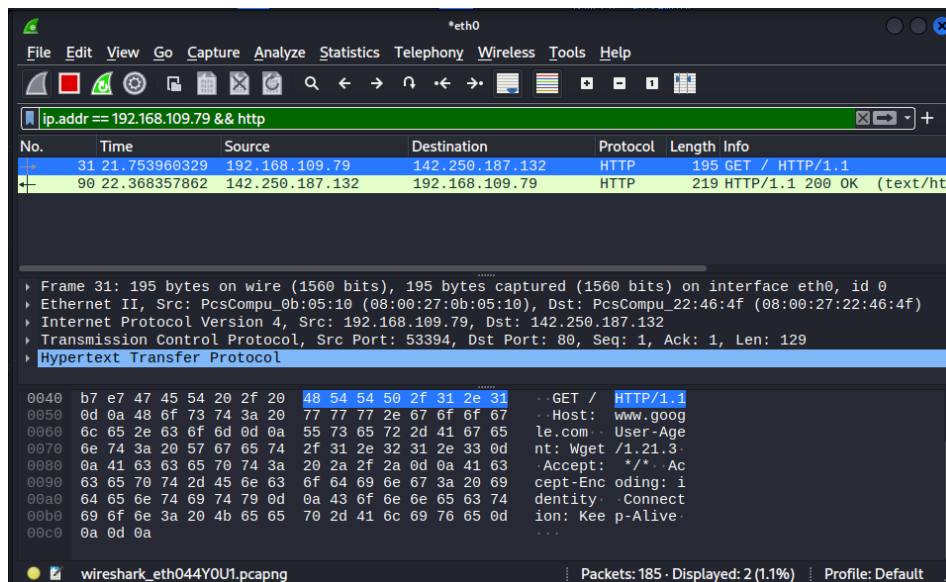
```
kali@victim-vm: ~  
File Actions Edit View Help  
(kali@victim-vm)-[~]  
$ ip neigh  
192.168.109.74 dev eth0 lladdr 08:00:27:22:46:4f REACHABLE  
192.168.109.230 dev eth0 lladdr 08:00:27:22:46:4f STALE
```

**Figure 3** ARP poisoning effect can be seen in the victim's machine by checking the ARP table. Now router MAC address is the same as the attacker.



```
kali@victim-vm: ~  
File Actions Edit View Help  
(kali@victim-vm)-[~]  
$ wget www.google.com  
--2022-11-11 10:53:29-- http://www.google.com/  
Resolving www.google.com (www.google.com)... 142.250.187.132, 2a00:1450:4017:  
80e::2004  
Connecting to www.google.com (www.google.com)|142.250.187.132|:80... connecte  
d.  
HTTP request sent, awaiting response... 200 OK  
Length: unspecified [text/html]  
Saving to: 'index.html'  
  
index.html [====>] 14.28K --.-KB/s in 0.09s  
  
2022-11-11 10:53:30 (157 KB/s) - 'index.html' saved [14625]
```

**Figure 4** Victim machine visits a website. The connection is established and packets are transferred between victim and the server. But this traffic first goes through the attacker and then packets are transmitted to the router.



```
*eth0  
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help  
ip.addr == 192.168.109.79 && http  
No. Time Source Destination Protocol Length Info  
31 21.753966329 192.168.109.79 142.250.187.132 HTTP 195 GET / HTTP/1.1  
90 22.368357862 142.250.187.132 192.168.109.79 HTTP 219 HTTP/1.1 200 OK (text/html)  
  
Frame 31: 195 bytes on wire (1560 bits), 195 bytes captured (1560 bits) on interface eth0, id 0  
Ethernet II, Src: PcsCompu_0b:05:10 (08:00:27:0b:05:10), Dst: PcsCompu_22:46:4f (08:00:27:22:46:4f)  
Internet Protocol Version 4, Src: 192.168.109.79, Dst: 142.250.187.132  
Transmission Control Protocol, Src Port: 53394, Dst Port: 80, Seq: 1, Ack: 1, Len: 129  
Hypertext Transfer Protocol  
GET / HTTP/1.1  
Host: www.goog  
le.com User-Age  
nt: Wget /1.21.3  
Accept: /* Ac  
cept-Enc oding: i  
dentity Connect  
ion: Kee p-Alive  
...  
wireshark_eth044YOU1.pcapng Packets: 185 - Displayed: 2 (1.1%) Profile: Default
```

**Figure 5** Attacker can inspect the HTTP request and the HTTP response by using Wireshark network analyzer tool.