CN PACKAGE ABSTRACT

Team Members:

1.Tarun Gunuguntla(19PD13)

2. M.S.Jegadeesh Manickam(19PD15)

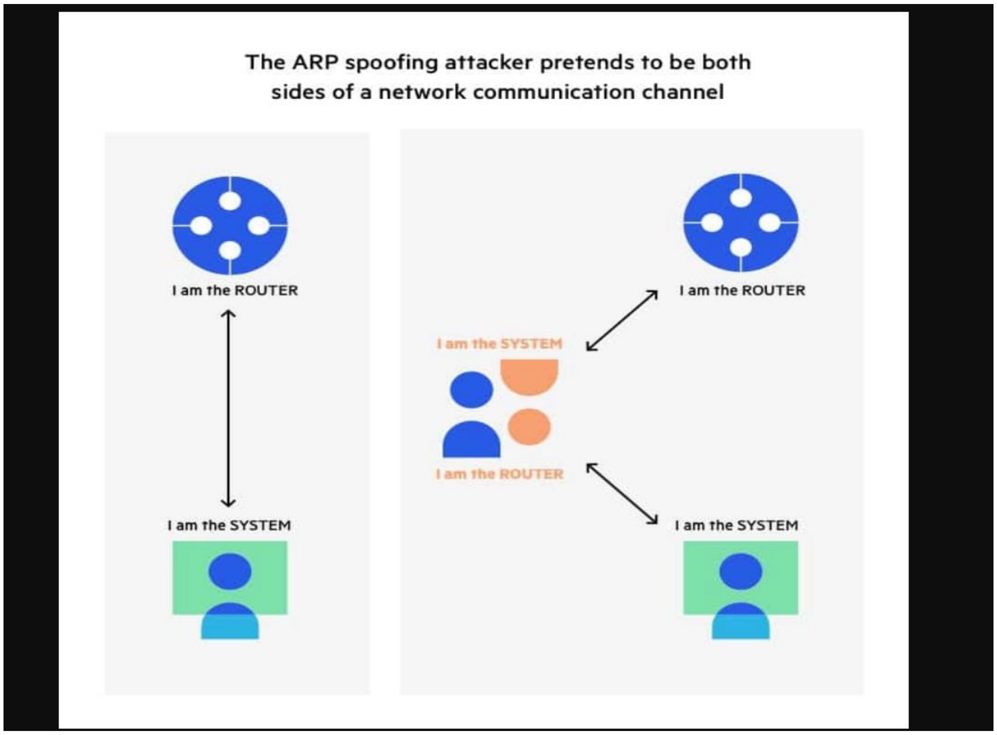
ABSTRACT:

We will be performing ARP Spoofing ,Arp Spoofing Attack Detection and Prevention.

Here we have used our Kali linux has the attacking machine and the Ubuntu as the victim.

We will be sending spoofed Arp responses from Kali linux to Ubuntu.

The concept of ARP Spoofing is illustrated below :



Algorithm for ARP Spoof Detection:

. After capturing the ARP packet in the victim System, it is analyzed to obtain the following two components:

- The source MAC address of the sniffed ARP packet (could be spoofed) .

- The real physical address of the sender (can be found by starting an ARP request for the source IP address).

These two components are then compared. If found to differ, then the system is definitely facing an ARP attack

Attack Prevention:

We will be making a static entry in the ARP cache using the ‘arp –s’ Command. Setting up a static entry for an address prevents the devices from listening to ARP responses for that address and thus ARP spoofing is prevented since the ARP cache of the target cannot be altered for the said MAC address

CONTRIBUTION:

19PD15: Did Research on the above mentioned concepts and implemented ARP Spoofing and Prevention.

19PD13: Implemented ARP Spoofing Detection and sniffing http payload.

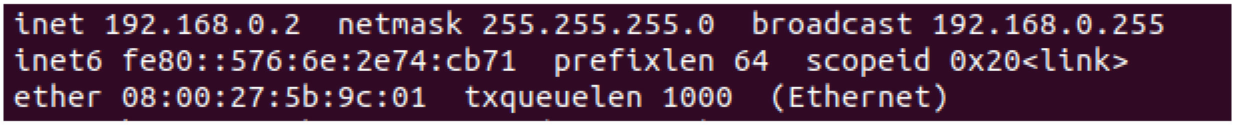
We learned the concepts together and implemented the concepts using Scapy together on our own.

OUTPUT

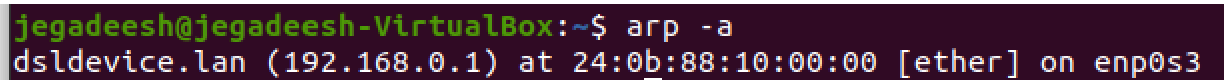
ARP SPOOFING ATTACK:-

Victim(Before Attack):

IP Address and Arp-Cache before Attack:



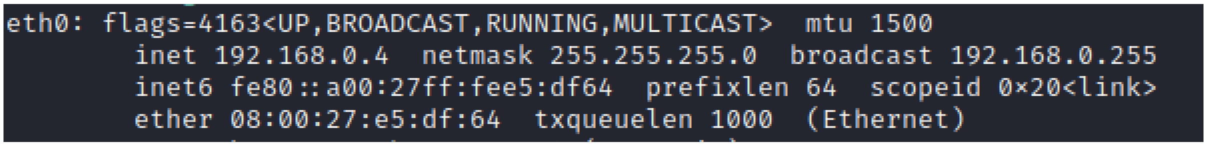
```
inet 192.168.0.2  netmask 255.255.255.0  broadcast 192.168.0.255
inet6 fe80::576:6e:2e74:cb71  prefixlen 64  scopeid 0x20<link>
ether 08:00:27:5b:9c:01  txqueuelen 1000  (Ethernet)
```



```
jegadeesh@jegadeesh-VirtualBox:~$ arp -a
dsldevice.lan (192.168.0.1) at 24:0b:88:10:00:00 [ether] on enp0s3
```

Attacker(kali Linux):

IP and MAC address:

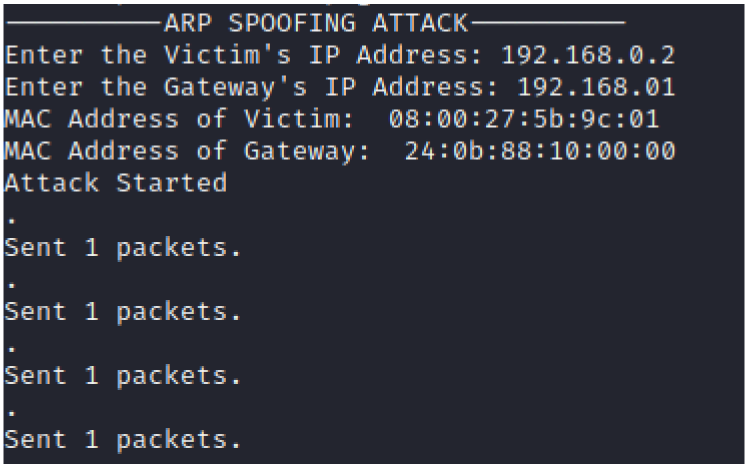


```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.4  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a00:27ff:fee5:df64  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:e5:df:64  txqueuelen 1000  (Ethernet)
```

Initiating Attack on kali:

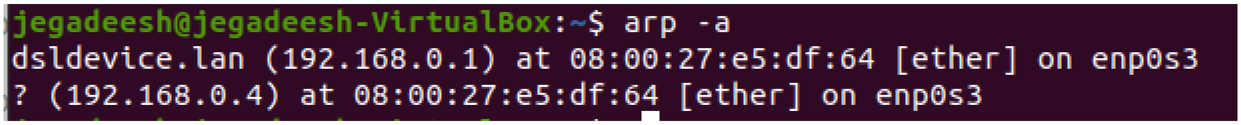
So in our Kali Linux, we will be sending spoofed Arp response to the victim with source IP as the Router’s IP and source MAC address as our machine’s MAC, so in this way victim thinks our machine as Router.

Similarly we send spoofed response to router with source IP as victim’s IP and source MAC as Attacker’s MAC ,so in this way router will be sending any response packet to the Attacker’s Machine.



```
──────────ARP SPOOFING ATTACK──────────
Enter the Victim's IP Address: 192.168.0.2
Enter the Gateway's IP Address: 192.168.01
MAC Address of Victim:  08:00:27:5b:9c:01
MAC Address of Gateway:  24:0b:88:10:00:00
Attack Started
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
```

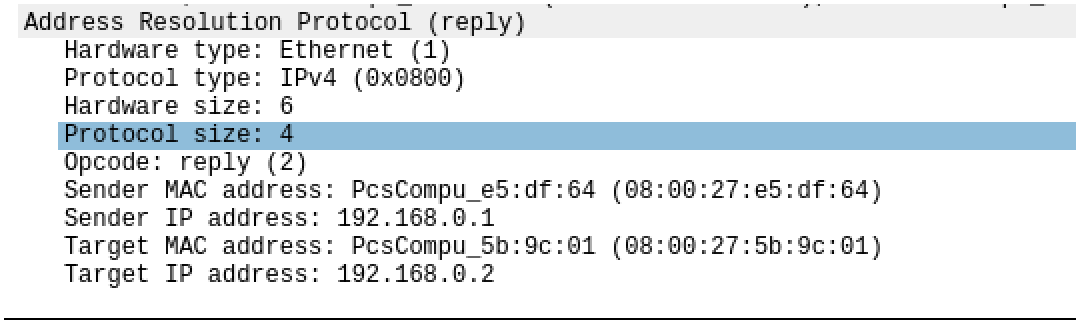
Victim(after Attack):



```
jegadeesh@jegadeesh-VirtualBox:~$ arp -a
dsldevice.lan (192.168.0.1) at 08:00:27:e5:df:64 [ether] on enp0s3
? (192.168.0.4) at 08:00:27:e5:df:64 [ether] on enp0s3
```

As you can see the MAC Address of router is replaced with MAC Address of Attacker Machine

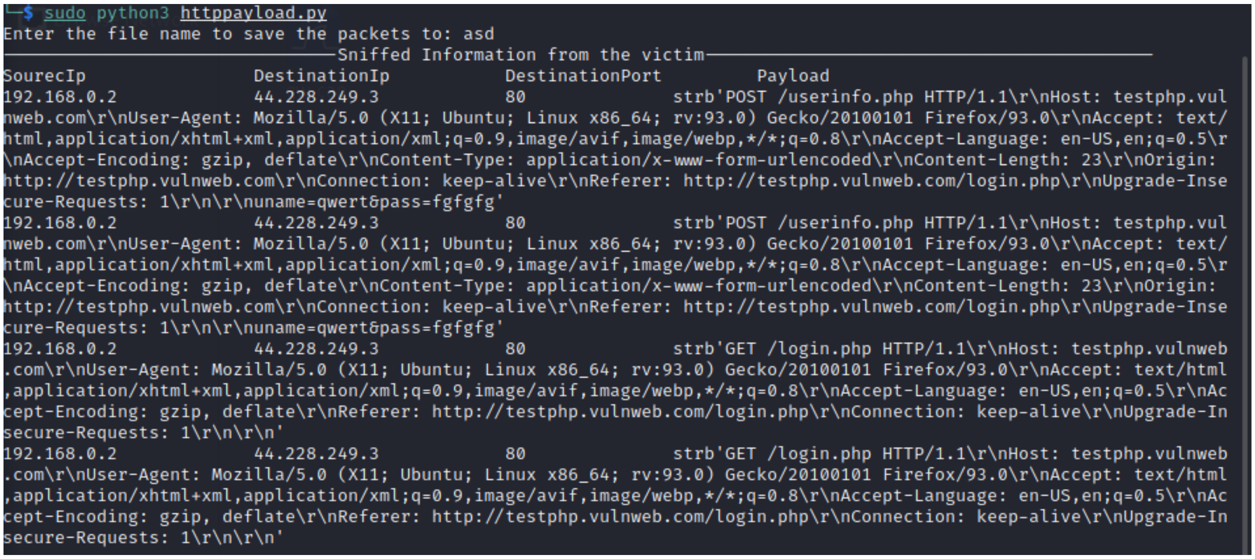
Wireshark Capture of Arp Messages at Victim:



```
Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: PcsCompu_e5:df:64 (08:00:27:e5:df:64)
    Sender IP address: 192.168.0.1
    Target MAC address: PcsCompu_5b:9c:01 (08:00:27:5b:9c:01)
    Target IP address: 192.168.0.2
```

At Attacker:

Sniffing user sensitive information on http requests from victim:

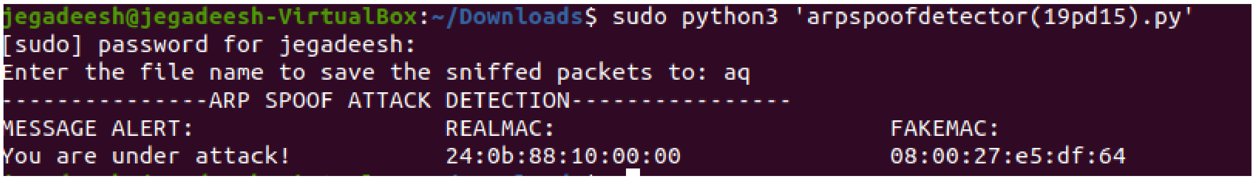


```
└─$ sudo python3 httppayload.py
Enter the file name to save the packets to: asd
────────────────────────────────Sniffed Information from the victim───────────────────────────────────────────
SourecIp                DestinationIp           DestinationPort         Payload
192.168.0.2             44.228.249.3            80                strb'POST /userinfo.php HTTP/1.1\r\nHost: testphp.vul
nweb.com\r\nUser-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:93.0) Gecko/20100101 Firefox/93.0\r\nAccept: text/
html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\nAccept-Language: en-US,en;q=0.5\r
\nAccept-Encoding: gzip, deflate\r\nContent-Type: application/x-www-form-urlencoded\r\nContent-Length: 23\r\nOrigin:
http://testphp.vulnweb.com\r\nConnection: keep-alive\r\nReferer: http://testphp.vulnweb.com/login.php\r\nUpgrade-Inse
cure-Requests: 1\r\n\r\nuname=qwert&pass=fgfgfg'
192.168.0.2             44.228.249.3            80                strb'POST /userinfo.php HTTP/1.1\r\nHost: testphp.vul
nweb.com\r\nUser-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:93.0) Gecko/20100101 Firefox/93.0\r\nAccept: text/
html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\nAccept-Language: en-US,en;q=0.5\r
\nAccept-Encoding: gzip, deflate\r\nContent-Type: application/x-www-form-urlencoded\r\nContent-Length: 23\r\nOrigin:
http://testphp.vulnweb.com\r\nConnection: keep-alive\r\nReferer: http://testphp.vulnweb.com/login.php\r\nUpgrade-Inse
cure-Requests: 1\r\n\r\nuname=qwert&pass=fgfgfg'
192.168.0.2             44.228.249.3            80                strb'GET /login.php HTTP/1.1\r\nHost: testphp.vulnweb
.com\r\nUser-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:93.0) Gecko/20100101 Firefox/93.0\r\nAccept: text/html
,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\nAccept-Language: en-US,en;q=0.5\r\nAc
cept-Encoding: gzip, deflate\r\nReferer: http://testphp.vulnweb.com/login.php\r\nConnection: keep-alive\r\nUpgrade-In
secure-Requests: 1\r\n\r\n'
192.168.0.2             44.228.249.3            80                strb'GET /login.php HTTP/1.1\r\nHost: testphp.vulnweb
.com\r\nUser-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:93.0) Gecko/20100101 Firefox/93.0\r\nAccept: text/html
,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\nAccept-Language: en-US,en;q=0.5\r\nAc
cept-Encoding: gzip, deflate\r\nReferer: http://testphp.vulnweb.com/login.php\r\nConnection: keep-alive\r\nUpgrade-In
secure-Requests: 1\r\n\r\n'
```

ARP SPOOF DETECTION:

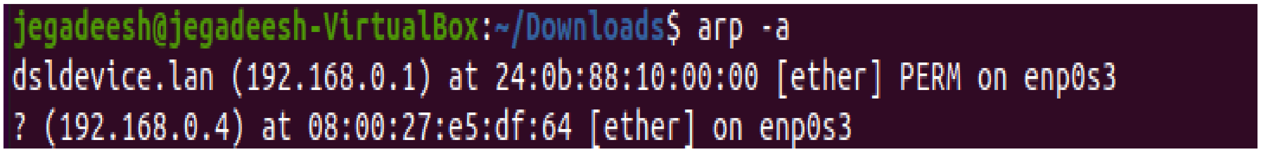
From the Below output we can see that MAC Address of router is replaced with the MAC Address of the Attacker's Machine, so we are under ARP Spoofing attack

At Victim:



```
jegadeesh@jegadeesh-VirtualBox:~/Downloads$ sudo python3 'arpspoofdetector(19pd15).py'
[sudo] password for jegadeesh:
Enter the file name to save the sniffed packets to: aq
---------------ARP SPOOF ATTACK DETECTION----------------
MESSAGE ALERT:                  REALMAC:                        FAKEMAC:
You are under attack!           24:0b:88:10:00:00               08:00:27:e5:df:64
```

Prevention



```
jegadeesh@jegadeesh-VirtualBox:~/Downloads$ arp -a
dsldevice.lan (192.168.0.1) at 24:0b:88:10:00:00 [ether] PERM on enp0s3
? (192.168.0.4) at 08:00:27:e5:df:64 [ether] on enp0s3
```

We have made static entry for Gateway to prevent ARP Spoofing.