Department of Information Technology

Academic Year: 2022-23 **Semester: V**
Class / Branch: TE IT Subject: Security Lab (SL)
Subject Lab Incharge: Prof. Apeksha Mohite

Experiment No. 1

1. Aim: To study IP spoofing and ARP spoofing over a local area network.

2. Theory:

2.1 IP Spoofing

2.1.1 What is Spoofing

A spoofing attack is when a malicious party impersonates another device or user on a network in order to launch attacks against network hosts, steal data, spread malware or bypass access controls. There are several different types of spoofing attacks that malicious parties can use to accomplish this. Some of the most common methods include IP address spoofing attacks, ARP spoofing attacks and DNS server spoofing attacks.

2.1.2 What is IP Spoofing

IP address spoofing is one of the most frequently used spoofing attack methods. In an IP address spoofing attack, an attacker sends IP packets from a false or "spoofed" source address in order to disguise itself. Denial-of- service attacks often use IP spoofing to overload networks and devices with packets that appear to be from legitimate source IP addresses.

There are two ways that IP spoofing attacks can be used to overload targets with traffic. One method is to simply flood a selected target with packets from multiple spoofed addresses. This method works by directly sending a victim more data than it can handle. The other method is to spoof the source IP address of known host which exist over same LAN and send request to server using this IP address. And server

will response back to the spoofed IP address. This is the scenario we are going to study and implement in this experiment.

2.1.3 Simulation of IP Spoofing attack using netkit

Every IP datagram sent in the Internet contains a source and destination IP address in its header. The source is the original sender of the datagram and the destination is the intended recipient. So, when your computer contacts a server on the Internet, that server knows your IP address as it is included in the source field of the IP datagram.

In this experiment, we are going to spoof the IP address of a host present on same LAN and send the request to the server using this spoofed IP address. The server will response back to host whose IP address we have spoofed for sending request. Setting the IP source address of datagrams to be a fake address is called address spoofing. In Linux it is very easy to do using iptables.

In the scenario shown in figure 1, there are three workstations with IP addresses 192.168.1.11, 192.168.1.12 and 192.168.1.13 respectively. Here 192.168.1.11 is an attacker and it sends request to 192.168.1.13 with spoofed IP address 192.168.1.12. As source IP address of the IP datagram received at destination is 192.168.1.12 receiver will response back same IP address.

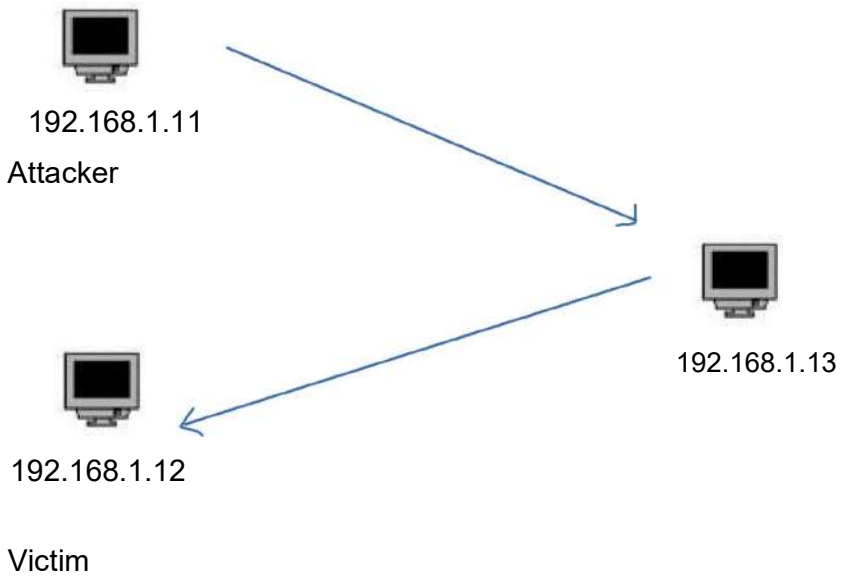


Fig. 1 IP Spoofing using IP address of known Host over same LAN.

Steps given below are used for simulation of the given scenario of IP spoofing attack using netkit.

Step1 : Using netkit create a LAN network having three workstations PC1,PC2 and PC3.All these three workstations should have same interface eth0 over same LAN A by using command

vstart pc1 –eth0=A

```
root@apsit-HP-Notebook:/home/apsit/Downloads/netkit# vstart pc1 --eth0=A


============= Starting virtual machine "pc1" =============
   Kernel:     /home/apsit/Downloads/netkit/kernel/netkit-kernel
   Modules:    /home/apsit/Downloads/netkit/kernel/modules
   Memory:     32 MB
   Model fs:   /home/apsit/Downloads/netkit/fs/netkit-fs
   Filesystem: /home/apsit/Downloads/netkit/pc1.disk
   Interfaces: eth0 @ A (/root/.netkit/hubs/vhub_root_A.cnct)
   Hostfs at:  /root

Running ==> /home/apsit/Downloads/netkit/bin/uml_switch -hub -unix /root/.netkit
/hubs/vhub_root_A.cnct </dev/null 2>&1
Running ==> xterm -e /home/apsit/Downloads/netkit/kernel/netkit-kernel modules=/
home/apsit/Downloads/netkit/kernel/modules name=pc1 title=pc1 umid=pc1 mem=36M u
bd0=/home/apsit/Downloads/netkit/pc1.disk,/home/apsit/Downloads/netkit/fs/netkit
-fs root=98:1 uml_dir=/root/.netkit/mconsole eth0=daemon,,,/root/.netkit/hubs/vh
ub_root_A.cnct hosthome=/root quiet con0=fd:0,fd:1 con1=null SELINUX_INIT=0
root@apsit-HP-Notebook:/home/apsit/Downloads/netkit# vstart pc2 --eth0=A
```

Similarly PC2 and PC3 are created using same command over LAN A.

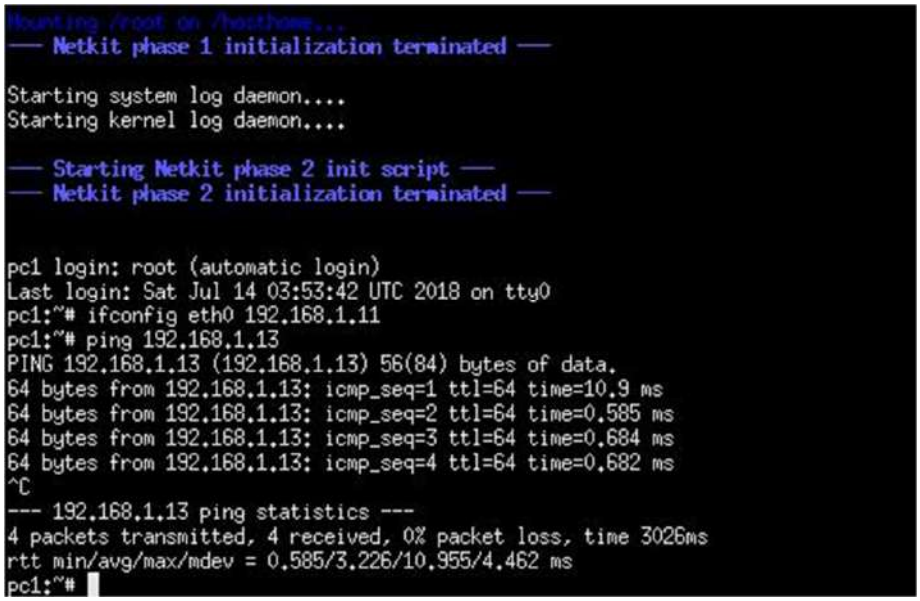
Step 2: Assign IP addresses to PC1,PC2 and PC3 as 192.168.1.11, 192.168.1.12 and 192.168.1.13 respectivley.

ifconfig eth0 192.168.1.11

this command will provide the ip 192.168.1.11 to pc1 and similarly we have provided the ip's to two others pc's pc2:-192.168.1.12 pc3:-192.168.1.13

Step 3: Ping without address spoofing

ping form PC1 to PC3 and check that response is given back PC1.

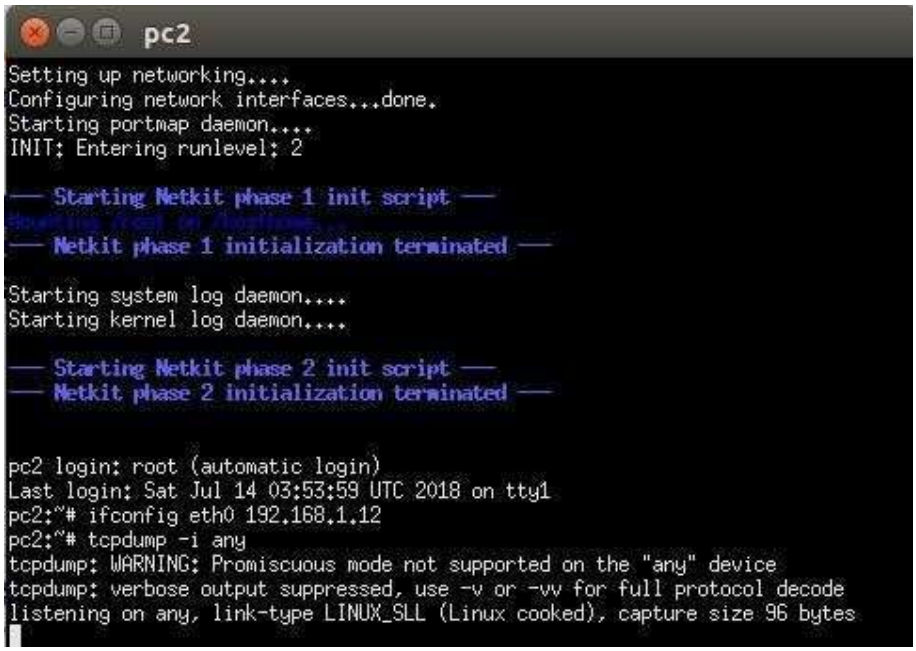


Step 4: PC1 will spoof IP address of PC2 by making changes in iptables of PC1.

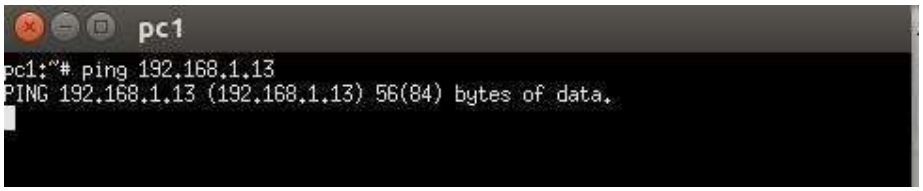
Following command is used to spoof the IP address of PC2 to create false identity by PC1.

iptables -t nat -A POSTROUTING -p icmp -j SNAT --to-source 192.168.1.12

Step 5: Packets are caputred using tcpdump. As response will be sent from PC3 to PC2, tcpdump command is executed on PC2 so that it can capture the reply.



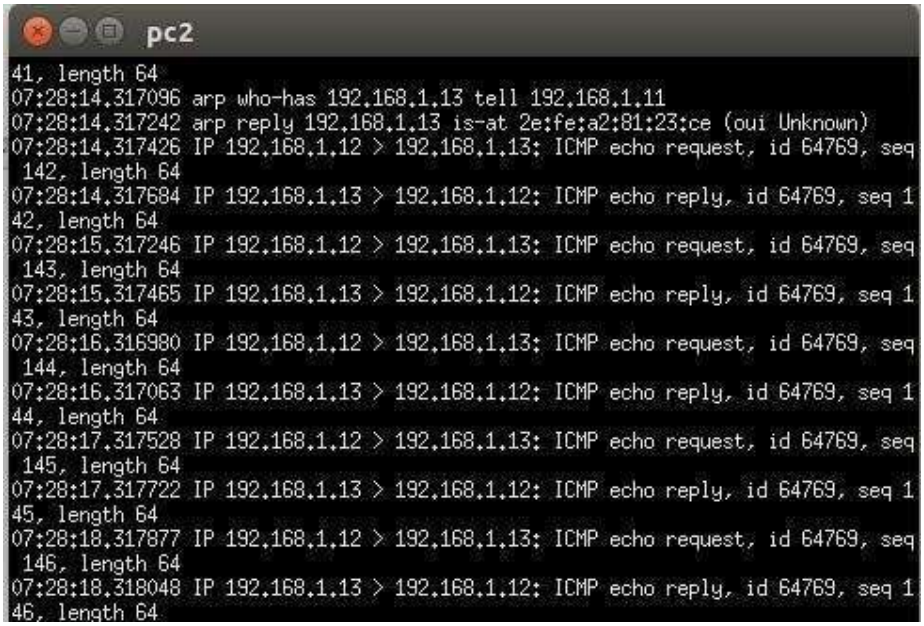
Step 6: As tcpdump is in listening state on PC2 , we can ping PC3 from PC1 to check to outocmes of IP spoofing attack.The ping command triggers ICMP Echo Request packets to be sent to the destination IP address every one second. When a computer receives an ICMP Echo Request it will reply with a ICMP Echo Reply.



Before the first ICMP Echo Request packet is sent by PC1, it must first discover the hardware address for the node with IP address 192.168.1.13. In a LAN communications are performed using the data link layer protocol, in this case Ethernet. Although PC1 knows the destination IP address, it must know the destination hardware (Ethernet or MAC) address to send the Ethernet frame to PC3. The Address Resolution Protocol (ARP) is used to perform this mapping of IP address to hardware address.PC1 broadcasts an ARP Request message to all nodes in the LAN, asking other nodes who has (knows) the hardware address for 192.168.1.13. PC3 has this IP

address, and therefore responds with an ARP Reply telling PC1 the corresponding hardware address: 08:00:27:c5:9f:e9. Now PC1 can send the ICMP Echo Request to PC2.

Step 7: Due to IP spoofing attack done by PC1, for PC3 the ping request is from PC2 so reply which is given back is for PC2.



So we have simulated IP spoofing to see the outcomes of IP spoofing attack. Now in next section we will discuss ARP Spoofing or MAC spoofing.

2.2 ARP Spoofing

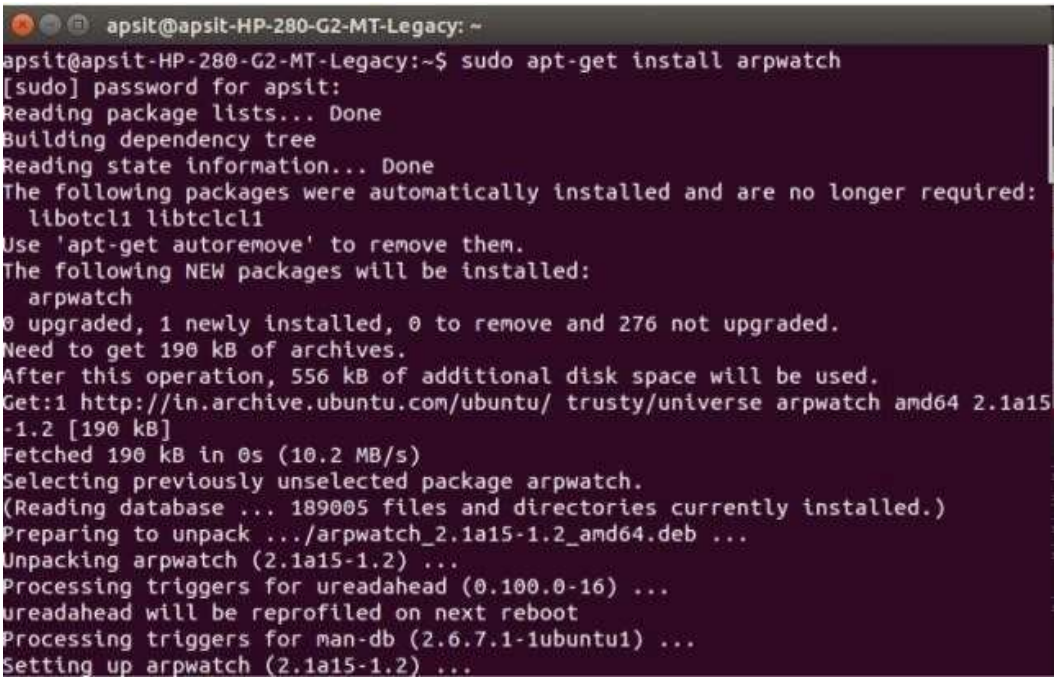
2.2.1 What is ARP Spoofing

ARP is short form of Address Resolution Protocol.This is a protocol that is used to resolve IP addresses to MAC (Media Access Control) addresses for transmitting data. In an ARP spoofing attack, a malicious party sends spoofed ARP messages across a local area network in order to link the attacker's MAC address with the IP address of a legitimate member of the network. This type of spoofing attack results in data that is intended for the host's IP address getting sent to the attacker instead. Malicious parties commonly use ARP spoofing to steal information, modify data in-transit or stop traffic on a LAN. ARP spoofing attacks can also be used to facilitate other

types of attacks, including denial-of-service, session hijacking and man-in-the-middle attacks. ARP spoofing only works on local area networks that use the Address Resolution Protocol.

2.2.2 Simulation ARP spoofing attack

Step 1: Install arpwatch. Arpwatch is an open source computer software program that helps us to monitor Ethernet traffic activity (like Changing IP and MAC Addresses) on the network and maintains a database of ethernet/ip address pairings. It produces a log of noticed pairing of IP and MAC addresses information along with a timestamps, so we can carefully watch when the pairing activity appeared on the network.



Step 2: Check the status of arpwatch to confirm that arpwatch is in running state.

Step 3: arpwatch maintains a log file to store information about IP addresses and MAC addresses.So any change in IP or MAC address can be noticed by the log entries of /var/log/syslog. Check the contents of /var/log/syslog using command tail -f /var/log/syslog.

```
Jul 19 10:15:55 apsit-HP-Notebook systemd[1]: Starting Hostname Service...
Jul 19 10:15:55 apsit-HP-Notebook dbus[827]: [system] Successfully activated ser
vice 'org.freedesktop.hostname1'
Jul 19 10:15:55 apsit-HP-Notebook systemd[1]: Started Hostname Service.
Jul 19 10:16:04 apsit-HP-Notebook arpwatch: new station 10.1.1.46 d8:32:e3:26:5e
:de enp1s0
Jul 19 10:16:04 apsit-HP-Notebook arpwatch: execl: /usr/lib/sendmail: No such fi
le or directory
Jul 19 10:16:04 apsit-HP-Notebook arpwatch: reaper: pid 4287, exit status 1
Jul 19 10:16:26 apsit-HP-Notebook org.gnome.Screenshot[1779]: ** Message: Unable
 to select area using GNOME Shell's builtin screenshot interface, resorting to f
allback X11.
Jul 19 10:16:36 apsit-HP-Notebook arpwatch: new station 192.168.1.40 34:de:1a:74
:71:74 enp1s0
Jul 19 10:16:36 apsit-HP-Notebook arpwatch: execl: /usr/lib/sendmail: No such fi
le or directory
Jul 19 10:16:36 apsit-HP-Notebook arpwatch: reaper: pid 4322, exit status 1
Jul 19 10:16:38 apsit-HP-Notebook arpwatch: new station 192.168.33.21 78:e3:b5:9
b:c3:89 enp1s0
Jul 19 10:16:38 apsit-HP-Notebook arpwatch: execl: /usr/lib/sendmail: No such fi
le or directory
Jul 19 10:16:39 apsit-HP-Notebook arpwatch: reaper: pid 4323, exit status 1
```

Step 4: Ping to any node on the same LAN. Here we ping to machine having IP address 192.168.36.101 Now 192.168.36.101 node has the IP adress and MAC address of your machine.

```
apsit@apsit-HP-Notebook:~$ ping 192.168.1.130
PING 192.168.1.130 (192.168.1.130) 56(84) bytes of data.
64 bytes from 192.168.1.130: icmp_seq=1 ttl=64 time=0.272 ms
64 bytes from 192.168.1.130: icmp_seq=2 ttl=64 time=0.324 ms
64 bytes from 192.168.1.130: icmp_seq=3 ttl=64 time=0.312 ms
64 bytes from 192.168.1.130: icmp_seq=4 ttl=64 time=0.302 ms
64 bytes from 192.168.1.130: icmp_seq=5 ttl=64 time=0.488 ms
64 bytes from 192.168.1.130: icmp_seq=6 ttl=64 time=0.340 ms
^C
--- 192.168.1.130 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5074ms
rtt min/avg/max/mdev = 0.272/0.339/0.488/0.072 ms
```

Step 5: Now change the MAC address of your system using ifconfig command. Again ping again to 192.168.36.101 with this changed MAC address.

```
apsit@apsit-HP-Notebook:~$ sudo ifconfig enp1s0 hw ether 00:1a:ff:0a:e7:1b
apsit@apsit-HP-Notebook:~$ ping 192.168.1.130
PING 192.168.1.130 (192.168.1.130) 56(84) bytes of data.
64 bytes from 192.168.1.130: icmp_seq=1 ttl=64 time=0.443 ms
64 bytes from 192.168.1.130: icmp_seq=2 ttl=64 time=0.242 ms
64 bytes from 192.168.1.130: icmp_seq=3 ttl=64 time=0.318 ms
64 bytes from 192.168.1.130: icmp_seq=4 ttl=64 time=0.402 ms
^C
--- 192.168.1.130 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3027ms
rtt min/avg/max/mdev = 0.242/0.351/0.443/0.078 ms
apsit@apsit-HP-Notebook:~$ ifconfig
enp1s0    Link encap:Ethernet  HWaddr 00:1a:ff:0a:e7:1b
          inet addr:192.168.1.82  Bcast:192.168.255.255  Mask:255.255.0.0
          inet6 addr: fe80::1a64:8027:42d1:e7c9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:66043 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2236 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13703357 (13.7 MB)  TX bytes:196237 (196.2 KB)
```

Step 6: Changes done in MAC address are notified in log entries of /var/log/syslog .

```
le or directory
Jul 19 10:20:35 apsit-HP-Notebook arpwatch: reaper: pid 4489, exit status 1
Jul 19 10:20:35 apsit-HP-Notebook arpwatch: new station 169.254.88.216 dc:4a:3e:
8d:8b:da enp1s0
Jul 19 10:20:39 apsit-HP-Notebook arpwatch: new station 192.168.1.215 dc:4a:3e:8
d:8b:da enp1s0
Jul 19 10:20:39 apsit-HP-Notebook arpwatch: execl: /usr/lib/sendmail: No such fi
le or directory
Jul 19 10:20:39 apsit-HP-Notebook arpwatch: reaper: pid 4492, exit status 1
Jul 19 10:20:40 apsit-HP-Notebook arpwatch: execl: /usr/lib/sendmail: No such fi
le or directory
Jul 19 10:20:40 apsit-HP-Notebook arpwatch: reaper: pid 4490, exit status 1
Jul 19 10:20:41 apsit-HP-Notebook arpwatch: changed ethernet address 192.168.1.8
2 00:1a:ff:0a:e7:1b (00:1a:ff:0a:e7:1c) enp1s0
Jul 19 10:20:41 apsit-HP-Notebook arpwatch: execl: /usr/lib/sendmail: No such fi
le or directory
Jul 19 10:20:41 apsit-HP-Notebook arpwatch: reaper: pid 4494, exit status 1
Jul 19 10:20:47 apsit-HP-Notebook arpwatch: new station 10.1.1.69 78:e3:b5:ab:56
:e0 enp1s0
Jul 19 10:20:47 apsit-HP-Notebook arpwatch: execl: /usr/lib/sendmail: No such fi
le or directory
Jul 19 10:20:47 apsit-HP-Notebook arpwatch: reaper: pid 4497, exit status 1
^C
```

Thus we have simulated ARP spoofing attack and noticed the log entries contaning the changed MAC address alert.

3. Conclusion:

Thus we have studied IP spoofing and ARP spoofing over a local area network. Arpwatch is a great is an open source computer software tool for monitoring Ethernet traffic activity (like Changing IP and MAC Addresses) on your network and maintains a database of ethernet/ip address pairings. It produces a log of noticed pairing of IP and MAC addresses information along with a timestamps, so you can carefully watch when the pairing activity appeared on the network.