Network Spoofing and Security Testing

1. Overview

I will use the Cyber Range: Kali Linux with Metasploitable (2018) environment to complete IP spoofing, MAC spoofing. Students will also use the Kali and Vulnerable Windows (64-bit) VMs environment to complete and examine an MITM attack called ARP poisoning.

2. Resources required

Required a Kali Linux VM running in the Cyber Range.

3. Initial Setup

I will log in to your Cyber Range account and select the Kali Linux with Metasploitable (2018) environment, then click "start" to start your environment and "join" to get to your Linux desktop login. Log in using these credentials:

Username: **`student`**
Password: **`student`**

4. Tasks

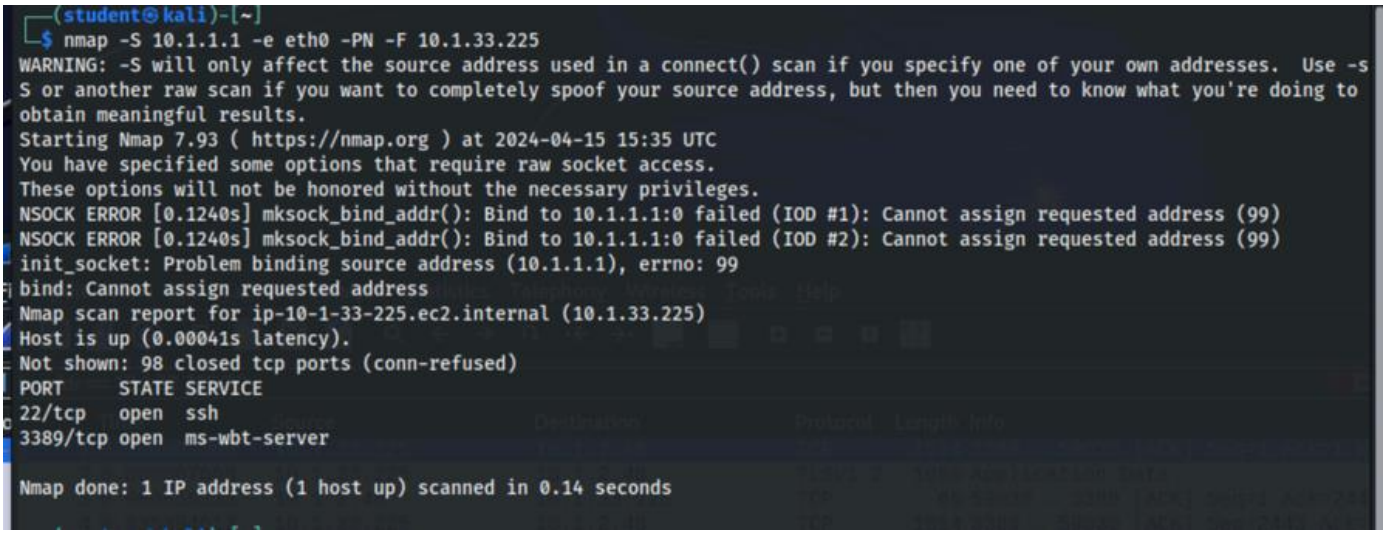
Task 1: IP Spoofing using nmap

Nmap has an option to let me scan a host using a spoofed IP. To spoof the IP during a nmap scan, I completed the following:

1. Open Wireshark. When it loads, choose eth0 by double clicking eth0 and then filter using `ip.addr == <spoofed IP> && ip.addr == <target IP>`, where <spoofed IP> is any valid but fake IP you want to use and <target IP> is the target Metasploitable IP address. NOTE: You will need to open Wireshark as root.

2. Open a terminal and type the following: `nmap -S <spoofed IP> -e eth0 -PN -F <Target IP >`

   Quick reminder of options:
   -S the spoof (i.e., fake) IP address
   -e specifies the interface
   -PN asks nmap do not ping before scanning, such that your real IP is not revealed.
   -F is sets the fast scan option (top 100 ports)



The spoofed IP I inserted was not able to be assigned, there was also a problem binding the source address. Whereas the target IP was scanned and was reported that 98 closed TCP ports weren't shown. The spoofed IP was 10.1.1.1 and the Target IP was 10.1.33.225.

The reason why the packet can't be sent in the reverse direction is that the packet will be received by a fake IP address and not by the attacker's IP. If the attacker has the power to handle the whole network, then the packet can be received in reverse direction otherwise it is very difficult to receive the packet in reverse direction.

Task 2: Mac Spoofing with Macchanger

Many organizations will prevent individuals from accessing their systems by using MAC filtering. This may work in some cases; however, many hackers will sniff the network and find a valid MAC address, then spoof that address using a program like Macchanger. The first thing I want to do is look at the options under the help menu.

- Typed `macchanger -h` and press enter
- Open a new terminal tab and become root.

Return to the macchanger tab and complete the follow command

- `sudo macchanger -m <Metasploitable MAC Address> eth0`

```
22/tcp open  ssh
msf6 > nmap -sS -Pn -v -p 22 10.1.33.225/20 | grep -B4 'open'
[*] exec: nmap -sS -Pn -v -p 22 10.1.33.225/20 | grep -B4 'open'

Initiating Parallel DNS resolution of 1 host. at 19:13
Completed Parallel DNS resolution of 1 host. at 19:13, 0.00s elapsed
Initiating SYN Stealth Scan at 19:13
Scanning 14 hosts [1 port/host]
Discovered open port 22/tcp on 10.1.38.209
--
Nmap scan report for ip-10-1-38-209.ec2.internal (10.1.38.209)
Host is up (0.000093s latency).

PORT   STATE SERVICE
22/tcp open  ssh
--
MAC Address: 0A:FF:E0:ED:8A:51 (Unknown)

Initiating SYN Stealth Scan at 19:13
Scanning ip-10-1-33-225.ec2.internal (10.1.33.225) [1 port]
Discovered open port 22/tcp on 10.1.33.225
--
Nmap scan report for ip-10-1-33-225.ec2.internal (10.1.33.225)
Host is up (0.00014s latency).
```

```
      -4 use IPv4 query transport only
      -6 use IPv6 query transport only

┌──(student㉿kali)-[~]
└─$ macchanger -s eth0
Current MAC:   0a:ff:f4:a8:e1:d1 (unknown)
Permanent MAC: 0a:ff:f4:a8:e1:d1 (unknown)

┌──(student㉿kali)-[~]
└─$ macchanger -r eth0
Current MAC:   0a:ff:f4:a8:e1:d1 (unknown)
Permanent MAC: 0a:ff:f4:a8:e1:d1 (unknown)
[ERROR] Could not change MAC: interface up or insufficient permissions: Operation not permitted

┌──(student㉿kali)-[~]
└─$ sudo macchanger -m 0A:FF:E):ED:8A:51  eth0
bash: syntax error near unexpected token `)'

┌──(student㉿kali)-[~]
└─$ sudo macchanger -m 0A:FF:E0:ED:8A:51  eth0
Current MAC:   0a:ff:f4:a8:e1:d1 (unknown)
Permanent MAC: 0a:ff:f4:a8:e1:d1 (unknown)
[ERROR] Could not change MAC: interface up or insufficient permissions: Operation not supported
```

I first used a command to display the appropriate Metasploitable Mac address, I did so by using the commands for lab 4. I used the Mac address in a regular student terminal to run the sudo macchanger

command and received an error. The error was caused because I was limited to the current Cyber Ranger Environment.

After taking a break for the Lab I think I accidentally reset the VM, so the IP and Mac address changed.

I was still able to go through each task with the different IP and Mac address.

Task 3: ARP spoofing for internal MIM attack

On the Kali (kali.example.com) VM, I opened a terminal and completed the following:

- `sudo apt-get update` and wait for it to finish. This will update the package lists in Linux such that new packages can be found.
- `sudo apt-get install dsniff` and press enter.
- `y` if I get a prompt and hit enter.

`Dsniff` is the package that contains `ARPspoof`.

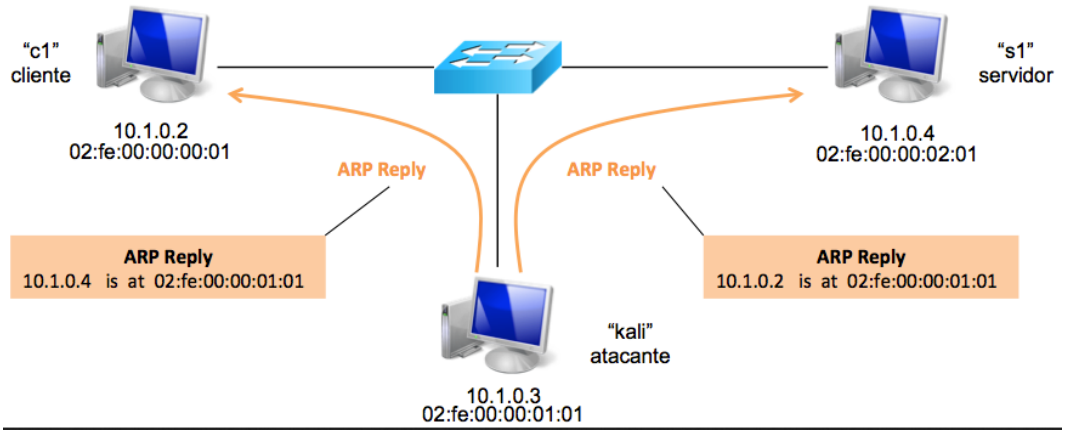
- Found the default gateway by typing `ip route` and pressing enter.
- Found the interface by typing `ifconfig` and pressing enter.

Next, I need to get our target IP address. Switched to the Windows VM tab, opened a command prompt, typed `ipconfig`, and pressed enter.

Then, I switched back to the Kali VM and returned to the terminal. Looking at the command below, I replaced the first IP address (target) with the default Windows VM IP and the second IP with the default gateway IP.

- `sudo arpspoof -i eth0 -t <target> <default gateway>`

To break this down, I am sending an ARP reply to the network stating that the attacker's MAC address is associated with the IP address of the default gateway. Such that if the victim is to send a packet to the IP of the default gateway, the victim's Link layer will send the packet to the attacker because it thinks that the attacker's MAC address is associated with the IP address. When the attacker receives the packet, the attacker can forward it to the real gateway, and in the meantime, inspect and even modify the packet. The above steps can't be implemented in the Cyber Range environment, and I will stop the lab here.



Source: https://raw.githubusercontent.com/cletomcj/vnx-sdn/master/noarpspoof/ARP_spoofing_2_esquema.png

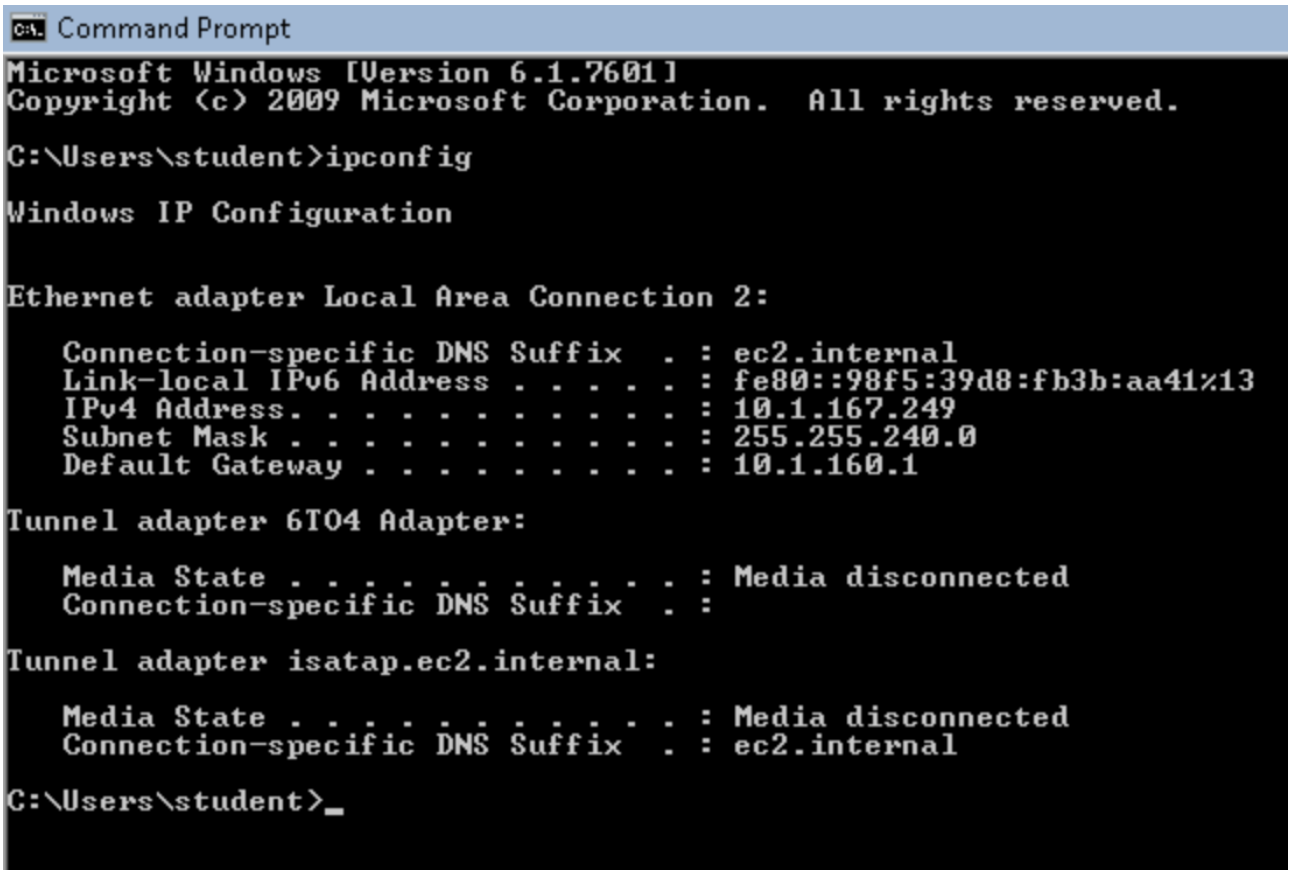
To sniff the network requests that are passing through the victim's machine, Ican use Wireshark on the Kali VM.

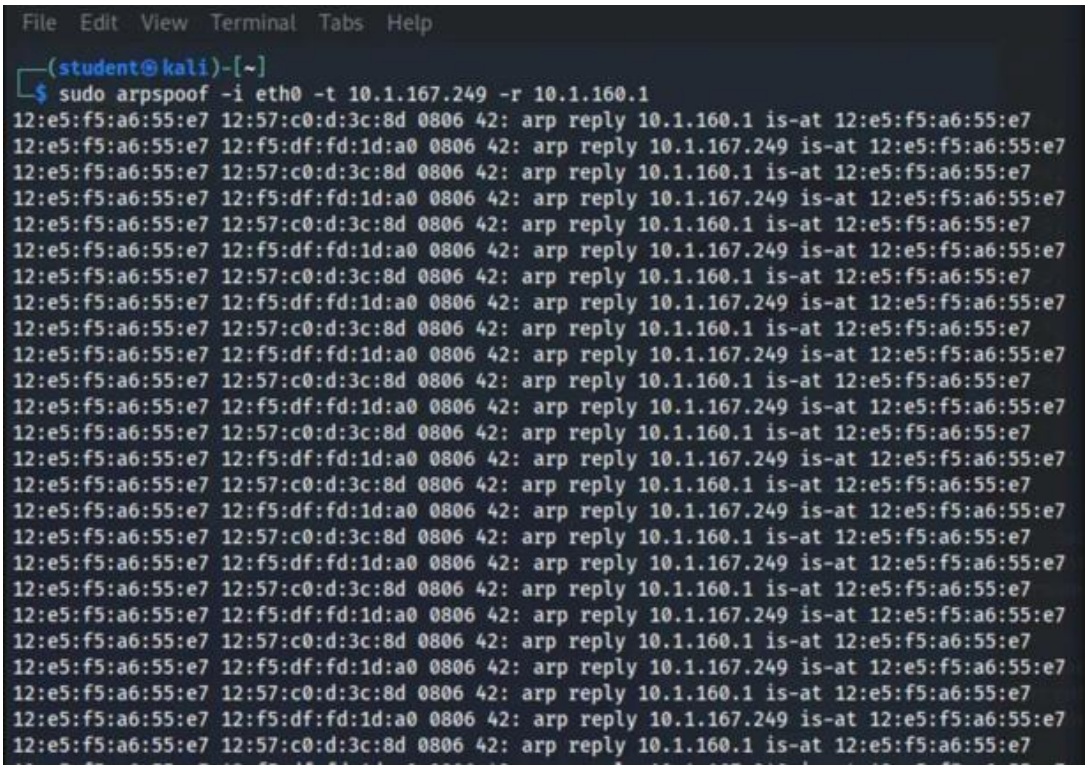
- Open Wireshark
- When Wireshark loads, choose eth0 by double clicking eth0.
- In the filter box, type `arp` and press enter.
- You should be able to see multiple ARP packets that advertising the default Gateway's IP is associated with the MAC address of your own machine.

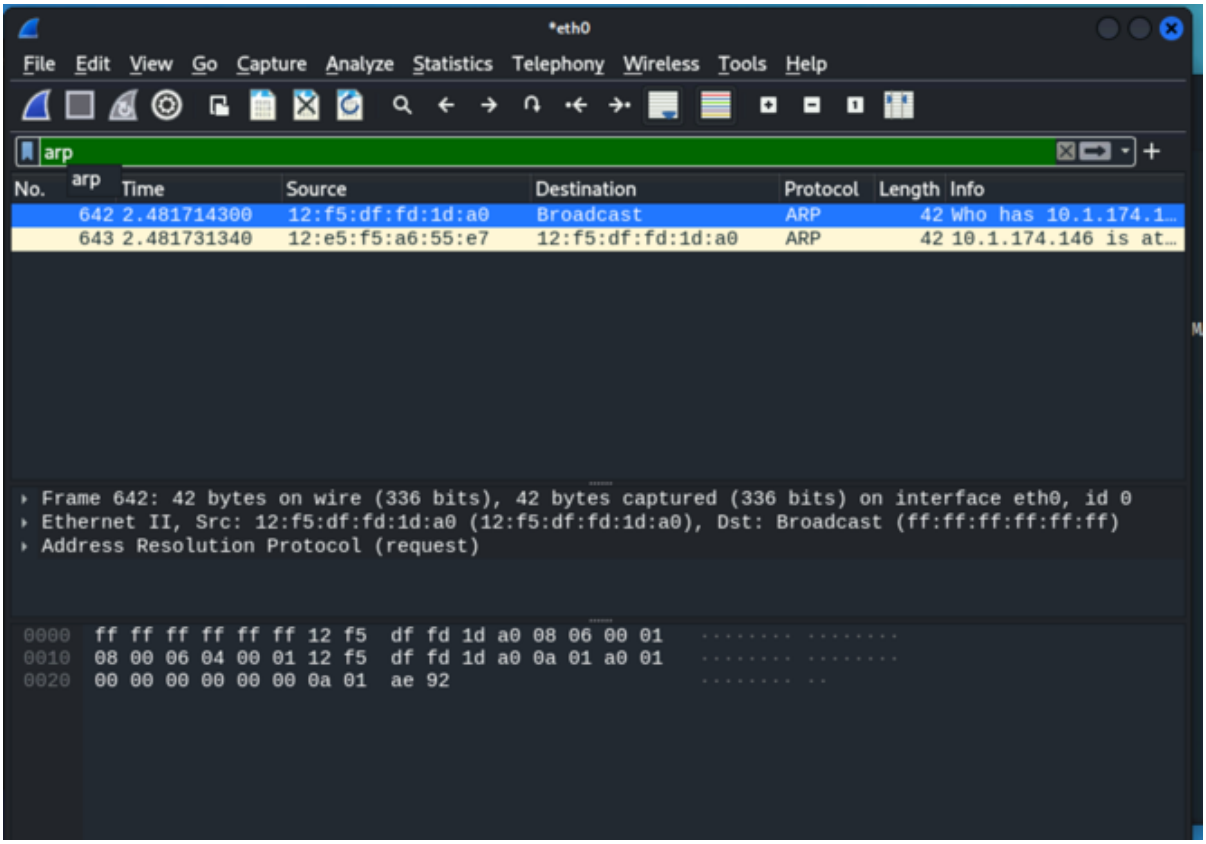
```
┌──(student㉿kali)-[~]
└─$ ip route
default via 10.1.160.1 dev eth0
10.1.160.0/20 dev eth0 proto kernel scope link src 10.1.174.146

┌──(student㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 10.1.174.146  netmask 255.255.240.0  broadcast 10.1.175.255
        inet6 fe80::10e5:f5ff:fea6:55e7  prefixlen 64  scopeid 0x20<link>
        ether 12:e5:f5:a6:55:e7  txqueuelen 1000  (Ethernet)
        RX packets 1771  bytes 112828 (110.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3665  bytes 3753695 (3.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 39  bytes 3194 (3.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 39  bytes 3194 (3.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```







I first found the default gateway and interface of my VM, and I then found the target IP address within the Windows VM. Afterward, I went back to the Kali Linux Vim, keeping in mind the target IP and default gateway, I then was able to send an ARP reply beginning with the Mac address. Within the Kali window, I opened Wireshark and used ARP as a filter to display ARP packets that are advertised by the default gateway's IP associated with the Mac Address of my machine.