

Homework #2

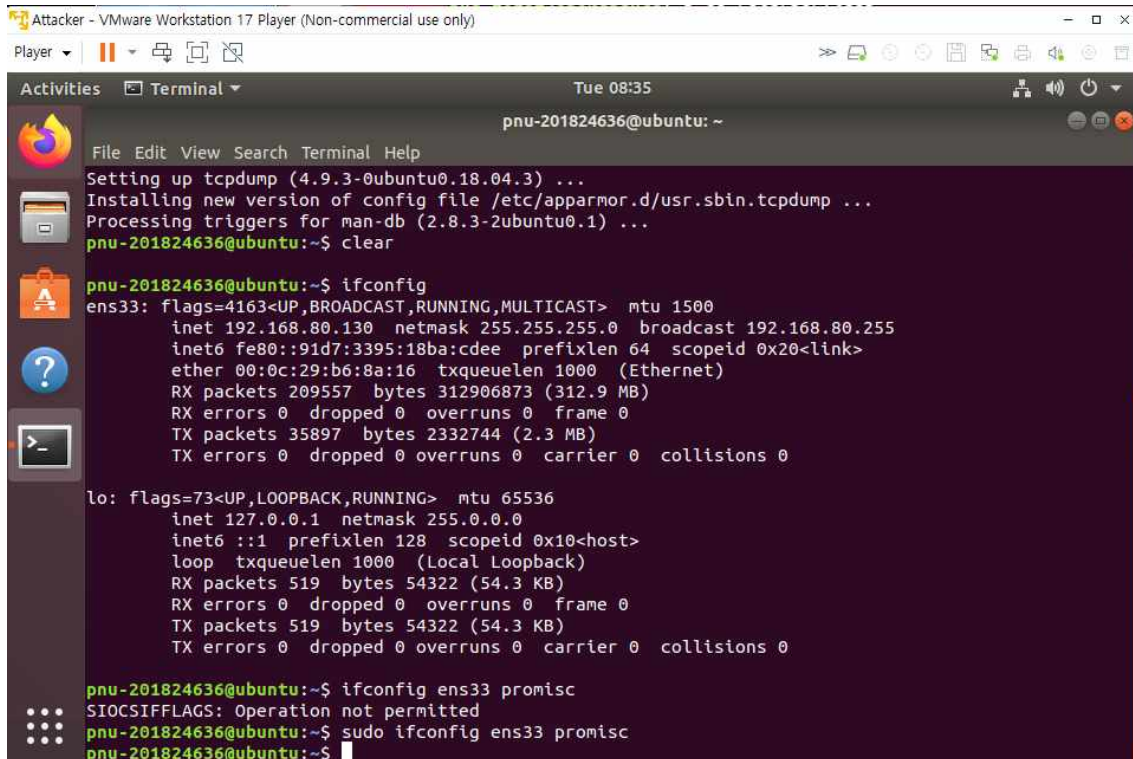


Subject	네트워크 보안
Professor	최 윤 호
Major	정보컴퓨터공학과
Student number	201824636
Name	이 강 우
Date	2023-10-24



1. Sniffing 공격 실습

[실습 1] TCP Dump를 사용한 Sniffing 공격 수행



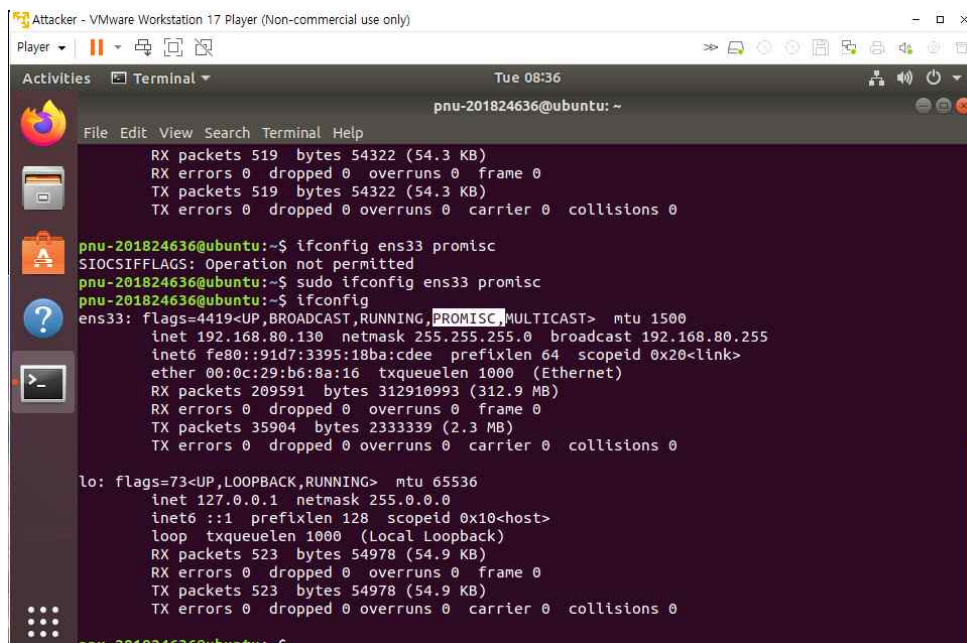
```
Attacker - VMware Workstation 17 Player (Non-commercial use only)
Player
Tue 08:35
pnu-201824636@ubuntu: ~
File Edit View Search Terminal Help
Setting up tcpdump (4.9.3-0ubuntu0.18.04.3) ...
Installing new version of config file /etc/apparmor.d/usr.sbin.tcpdump ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
pnu-201824636@ubuntu:~$ clear

pnu-201824636@ubuntu:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.80.130 netmask 255.255.255.0 broadcast 192.168.80.255
    inet6 fe80::91d7:3395:18ba:cdee prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:b6:8a:16 txqueuelen 1000 (Ethernet)
    RX packets 209557 bytes 312906873 (312.9 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 35897 bytes 2332744 (2.3 MB)
    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 519 bytes 54322 (54.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 519 bytes 54322 (54.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pnu-201824636@ubuntu:~$ ifconfig ens33 promisc
SIOCSIFFLAGS: Operation not permitted
pnu-201824636@ubuntu:~$ sudo ifconfig ens33 promisc
pnu-201824636@ubuntu:~$
```

ifconfig를 통해 어떤 포트를 사용하는지 확인한다. ens33 포트를 사용하는 것을 알 수 있으며, Sniffing 공격을 위해 모든 패킷을 수신하는 promisc 모드로 변경한다.



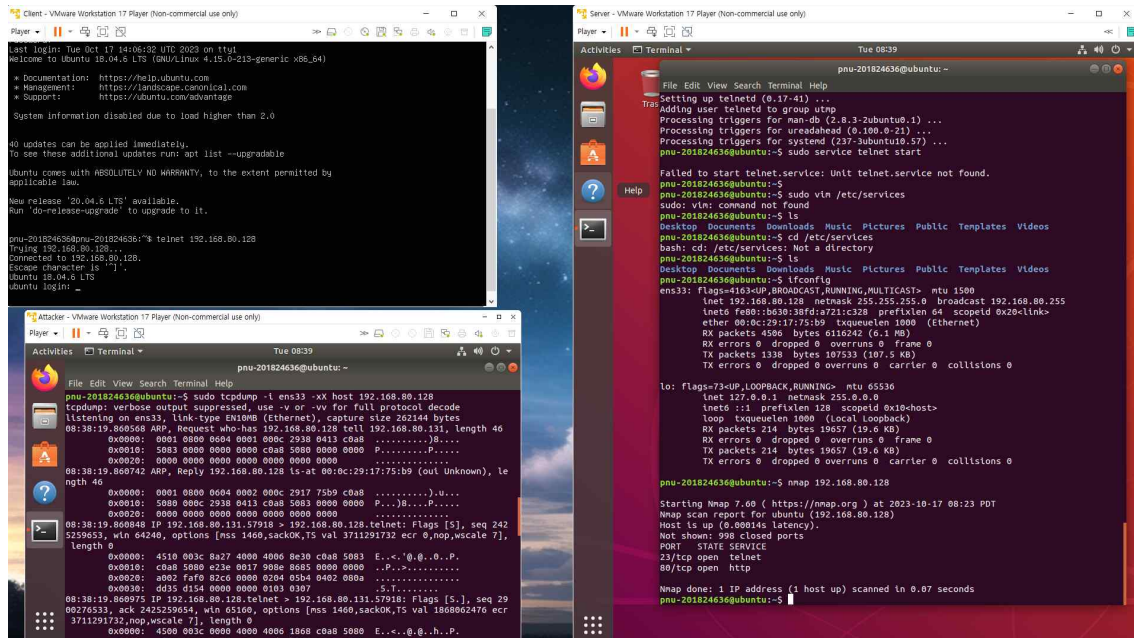
```
pnu-201824636@ubuntu:~$ ifconfig
RX packets 519 bytes 54322 (54.3 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 519 bytes 54322 (54.3 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pnu-201824636@ubuntu:~$ ifconfig ens33 promisc
SIOCSIFFLAGS: Operation not permitted
pnu-201824636@ubuntu:~$ sudo ifconfig ens33 promisc
pnu-201824636@ubuntu:~$ ifconfig
ens33: flags=4419<UP,BROADCAST,RUNNING,PROMISC,MULTICAST> mtu 1500
    inet 192.168.80.130 netmask 255.255.255.0 broadcast 192.168.80.255
    inet6 fe80::91d7:3395:18ba:cdee prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:b6:8a:16 txqueuelen 1000 (Ethernet)
    RX packets 209591 bytes 312910993 (312.9 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 35904 bytes 2333339 (2.3 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

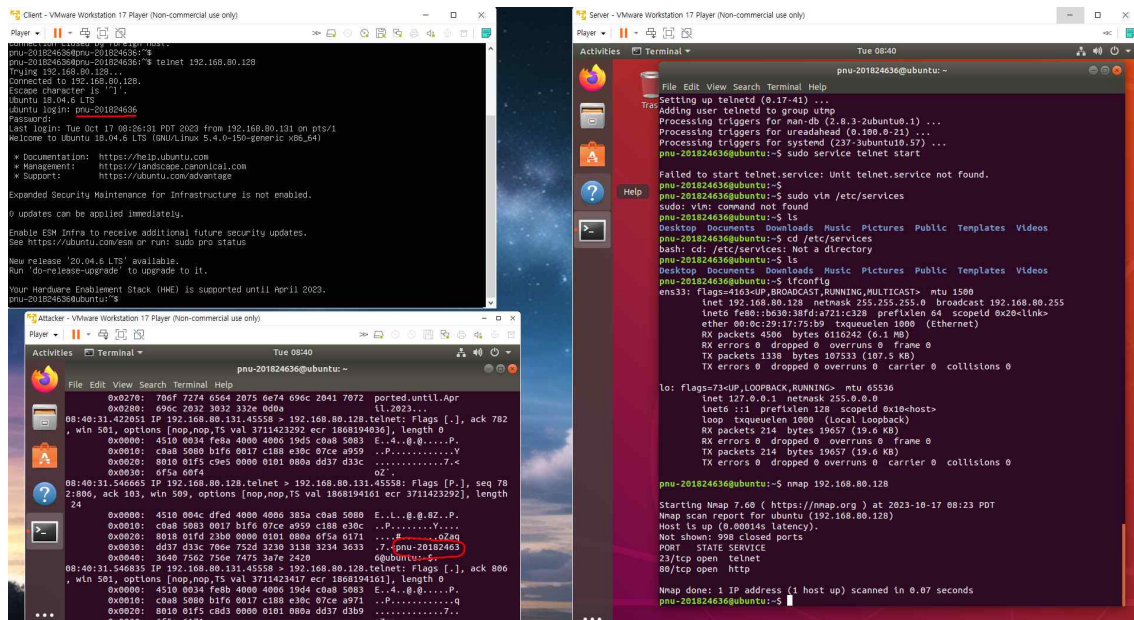
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 523 bytes 54978 (54.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 523 bytes 54978 (54.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pnu-201824636@ubuntu:~$
```

다시 ifconfig하여 확인해보면, promisc모드로 들어간 것을 확인해볼 수 있다.



Attacker가 네트워크를 점검하기 위해 만들어졌지만, sniffing의 수단으로 사용하는 tcpdump를 이용해 Server(192.168.80.128)로 telnet 접속한 클라이언트의 개인정보를 -xX(16진수로 파싱하여) 옵션으로 sniffing 한다.



Client가 Server로 telnet 접속을 마치자, Attacker의 터미널에 plainText로 한 글자씩 Client의 개인정보가 Attacker의 콘솔에 출력된다.

[실습 2] Dsniff를 사용한 Sniffing 공격 수행

```

Attacker - VMware Workstation 17 Player (Non-commercial use only)
Player
Tue 09:11
pnu-201824636@ubuntu: ~
File Edit View Search Terminal Help
root?
pnu-201824636@ubuntu:~$ sudo apt-get install dsniff
[sudo] password for pnu-201824636:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
 fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0
 gir1.2-gstreamer-1.0 gir1.2-gudev-1.0 gir1.2-udisks-2.0 grilo-plugins-0.3-base
 gstreamer1.0-gtk3 libboost-date-time1.65.1 libboost-filesystem1.65.1
 libboost-iostreams1.65.1 libboost-locale1.65.1 libcdr-0.1-1
 libclucene-contribs1v5 libclucene-core1v5 libcms-0.5-5v5 libcolamd2
 libdazzle-1.0-0 libe-book-0.1-1 libdataserverui-1.2-2 libeat0 libepubgen-0.1-1
 libetonyek-0.1-1 libevent-2.1-6 libexiv2-14 libfreerdp-client2-2 libfreerdp2-2
 libgc1c2 libgee-0.8-2 libgexiv2-2 libgom-1.0-0 libgpgmepp6 libgpod-common
 libgpod4 liblangtag-common liblangtag1 liblirc-client0 liblua5.3-0
 libmediaart-2.0-0 libmsh-0.1-1 libodfgen-0.1-1 libqwing2v5 librav1b
 librevenge-0.0-0 libsgutils2-2 libssh-4 libsuitesparseconfig5 libvncclient1
 libwinpr2-2 libxapian3 libxmlsec1-nss lp-solve media-player-info python3-mako
 python3-markupsafe syslinux syslinux-common syslinux-legacy usb-creator-common
 Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
 libnids1.21
  
```

Attacker VMWare에 dsniff 툴을 설치한다. dsniff는 거의 모든 형태의 패킷을 읽을 수 있으며, telnet도 그중 하나이다.

```

Client - VMware Workstation 17 Player (Non-commercial use only)
Player
Tue 09:15
pnu-201824636@ubuntu: ~
File Edit View Search Terminal Help
Unpacking dsniff (2.4bi-debian-20.1-build1) ...
Setting up libnids1.21:amd64 (1.24-4) ...
Setting up dsniff (2.4bi-debian-20.1-build1) ...
Processing triggers for man-db (2.8.3-2ubuntu1.4) ...
pnu-201824636@ubuntu:~$ frgrouter
frgrouters: command not found
pnu-201824636@ubuntu:~$ dsniff
dsniff: nids_init: ens33: You don't have permission to capture on that device (sock
ext: operation not permitted)
pnu-201824636@ubuntu:~$ sudo dsniff
[sudo] password for pnu-201824636:
dsniff: listening on ens33
10/17/22 09:15:00 tcp 192.168.0.131.57440 -> 192.168.0.126.23 (telnet)
pnu-201824636@ubuntu:~$
Server - VMware Workstation 17 Player (Non-commercial use only)
Player
Tue 09:15
pnu-201824636@ubuntu: ~
File Edit View Search Terminal Help
Setting up telnetd (0.17-41) ...
Adding user telnetd to group utmp
Processing triggers for man-db (2.8.3-2ubuntu1.4) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for systemd (237-3ubuntu10.57) ...
pnu-201824636@ubuntu:~$ sudo service telnet start
Failed to start telnet.service: Unit telnet.service not found.
pnu-201824636@ubuntu:~$ sudo vim /etc/services
pnu-201824636@ubuntu:~$ cd /etc/services
bash: cd: /etc/services: not a directory
pnu-201824636@ubuntu:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
pnu-201824636@ubuntu:~$ cd /etc/services
pnu-201824636@ubuntu:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
pnu-201824636@ubuntu:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.128 netmask 255.255.255.0 broadcast 192.168.0.255
    ether 08:00:27:28:f2:22 txqueuelen 64 scopeid 0x20<link>
    RX packets 4506 bytes 616242 (6.1 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1338 bytes 107533 (107.5 KB)
    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 214 bytes 19657 (19.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 214 bytes 19657 (19.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

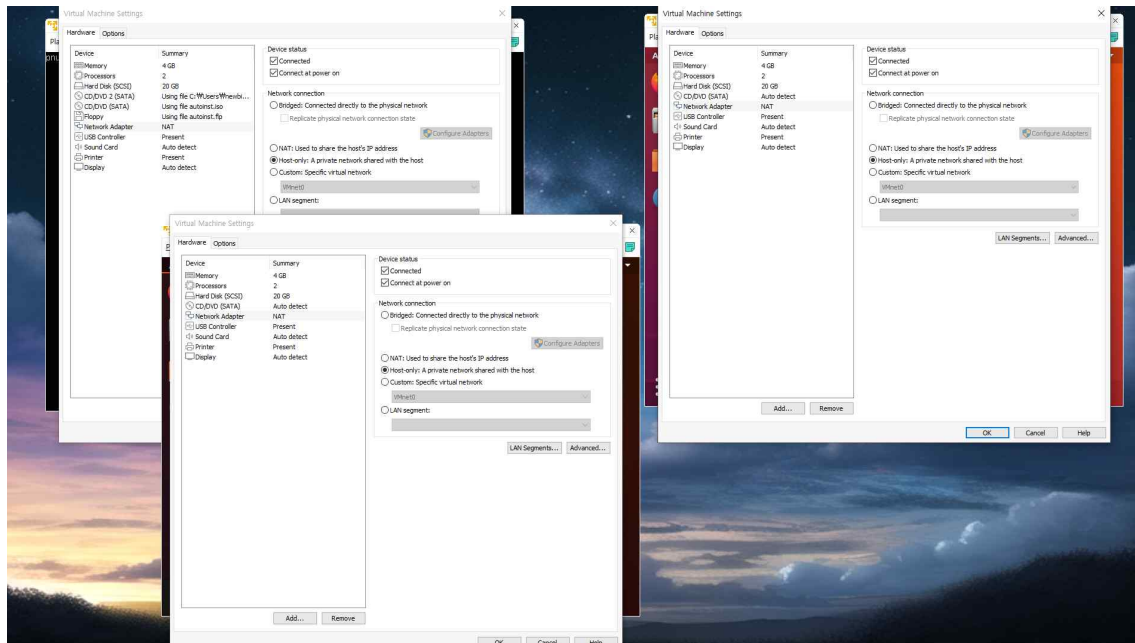
pnu-201824636@ubuntu:~$ nmap 192.168.0.128
Starting Nmap 7.60 (https://nmap.org) at 2022-10-17 08:23 PDT
Nmap scan report for ubuntu (192.168.0.128)
Host is up (0.00014s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
23/tcp    open  telnet
80/tcp    open  http
Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
pnu-201824636@ubuntu:~$
  
```

Client가 Server를 접속한 뒤 입력한 ID, Password와 모든 리눅스 명령어가 Server의

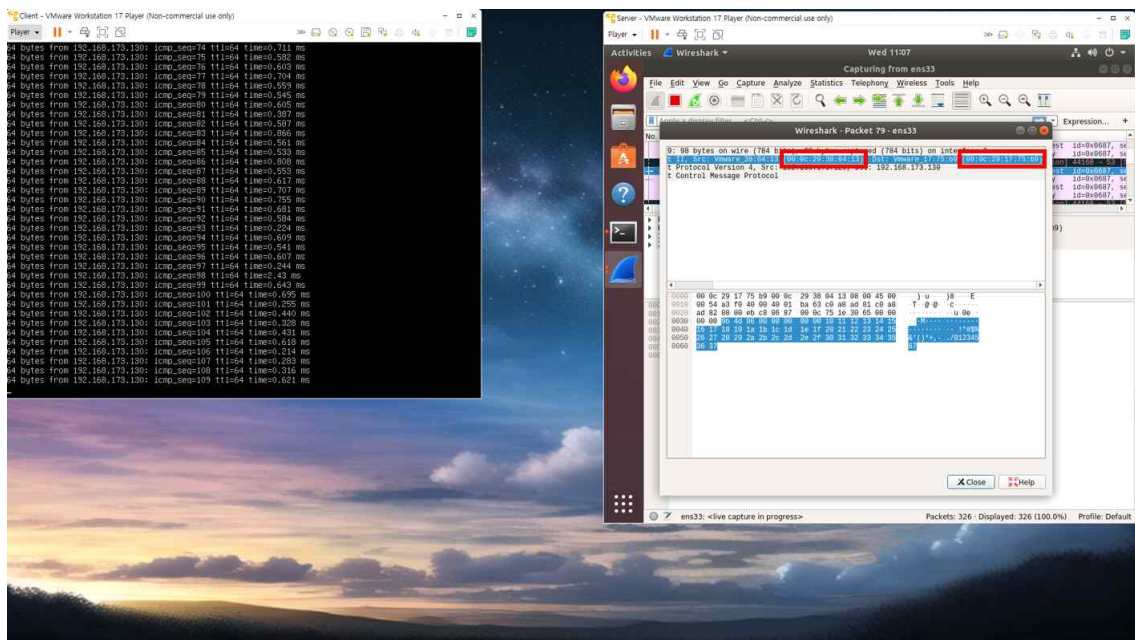
telnet 접속이 끊김과 동시에 Attacker 터미널에 출력된다.

2. ARP Spoofing 공격 수행

[실습 1] 공격 발생 전 서버에 ICMP 패킷 전송

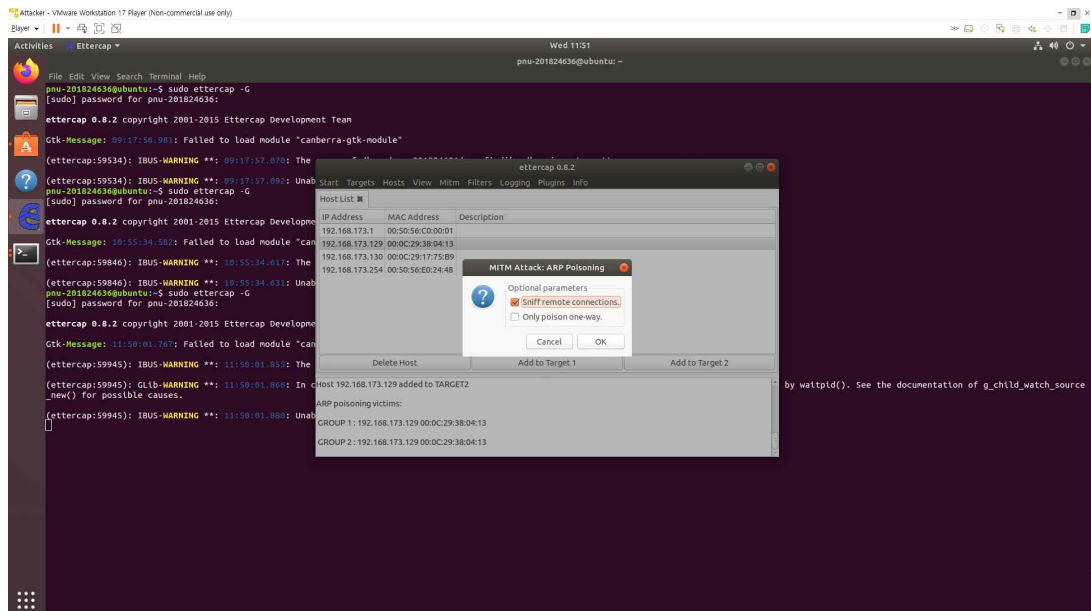


사전 설정을 위해 모든 노드의 Network를 NAT -> HOST ONLY로 변경한다.

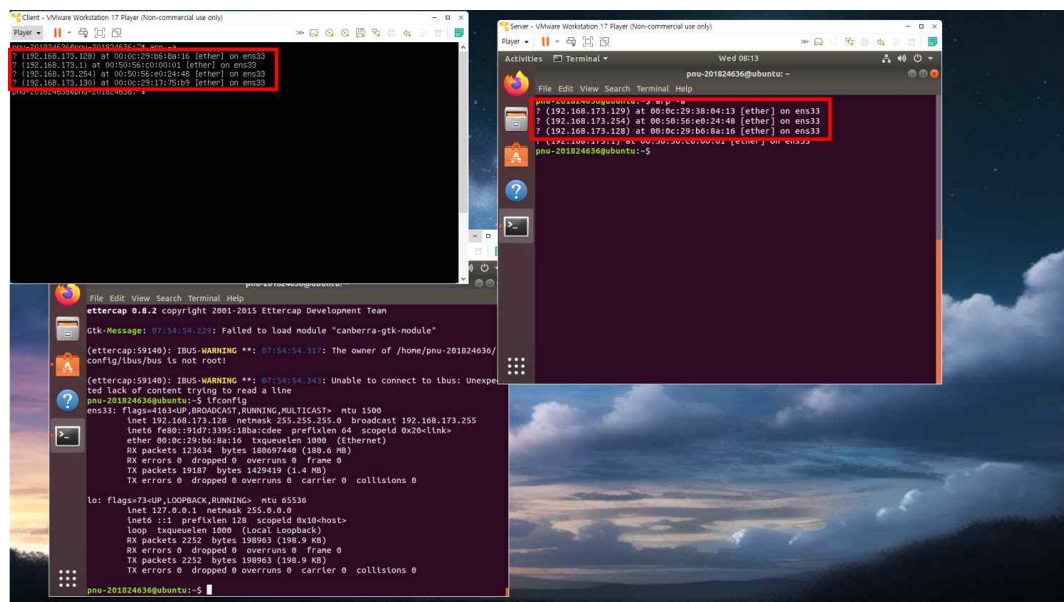


클라이언트에서 ping을 통해 서버와 통신한다. ICMP 패킷이 잘 도착하는 것을 wireshark를 통해 확인할 수 있다. MAC - Address를 보면 Client (00:0c:29:38:04:13) -> Server (00:0c:29:17:75:b9)로 잘 도착하는 것을 확인할 수 있다.

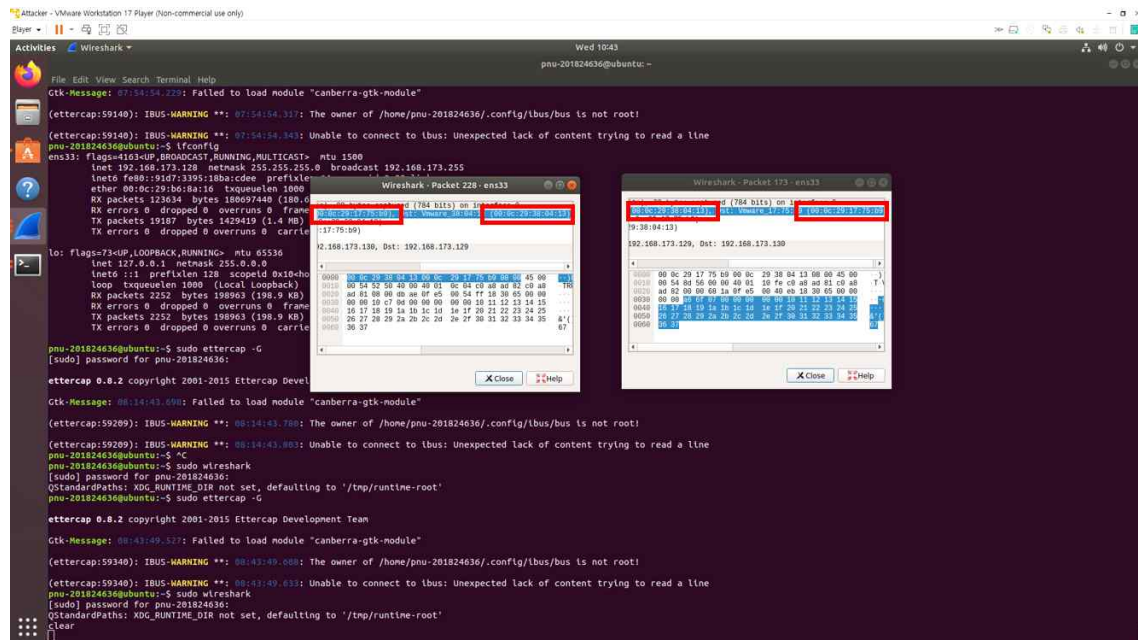
[실습 2] ettercap 툴을 통한 ARP Spoofing 공격 수행



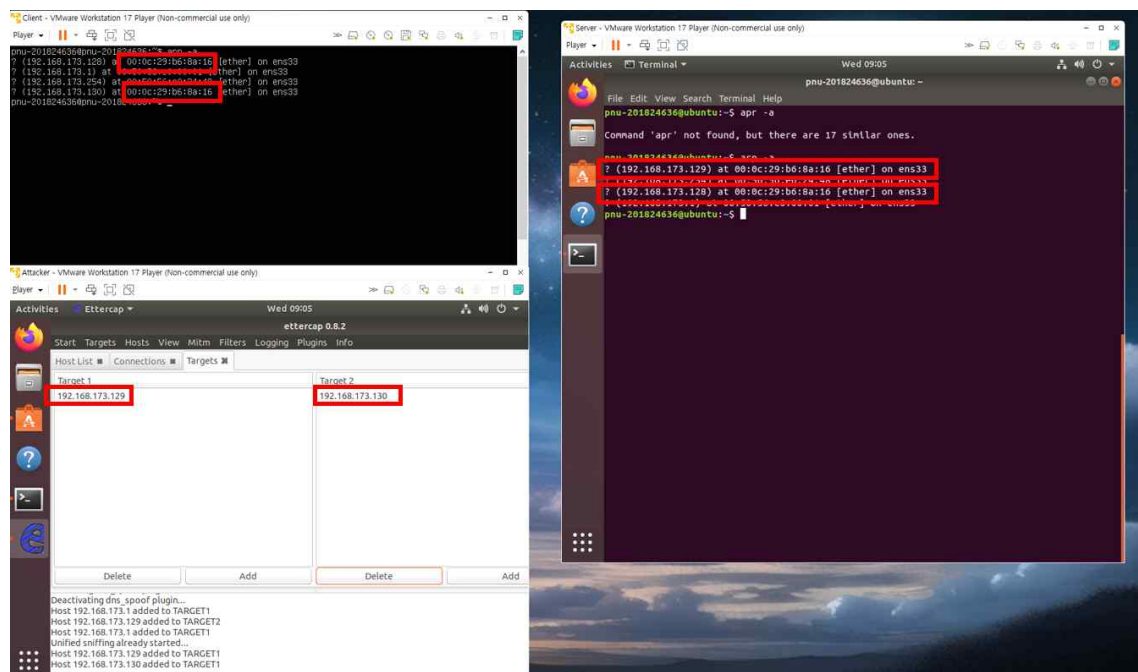
ettercap-graphical 툴을 Unified sniffing 모드로 실행하고, target1을 Client, target2를 Server로 설정하고 ARP Poison으로 Sniffing을 수행한다. 이 과정으로 arp를 조작하여 client와 server의 패킷을 조회할 수 있다. 그 전에, 정상 MAC과 IP의 Mapping을 조회해 보자.



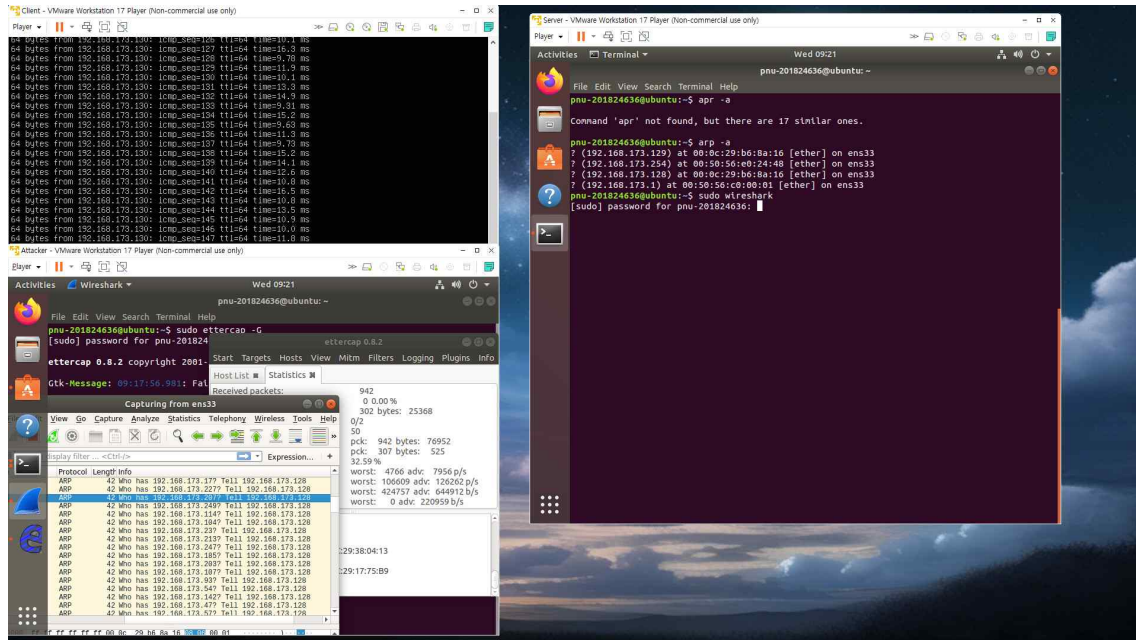
ARP 스누핑 전에 ARP -a를 이용해 MAC Address를 확인해 보면
 Client (00:0c:29:38:04:13), Server (00:0c:29:17:75:b9), Attacker (00:0c:29:b6:8a:16)
 이다.



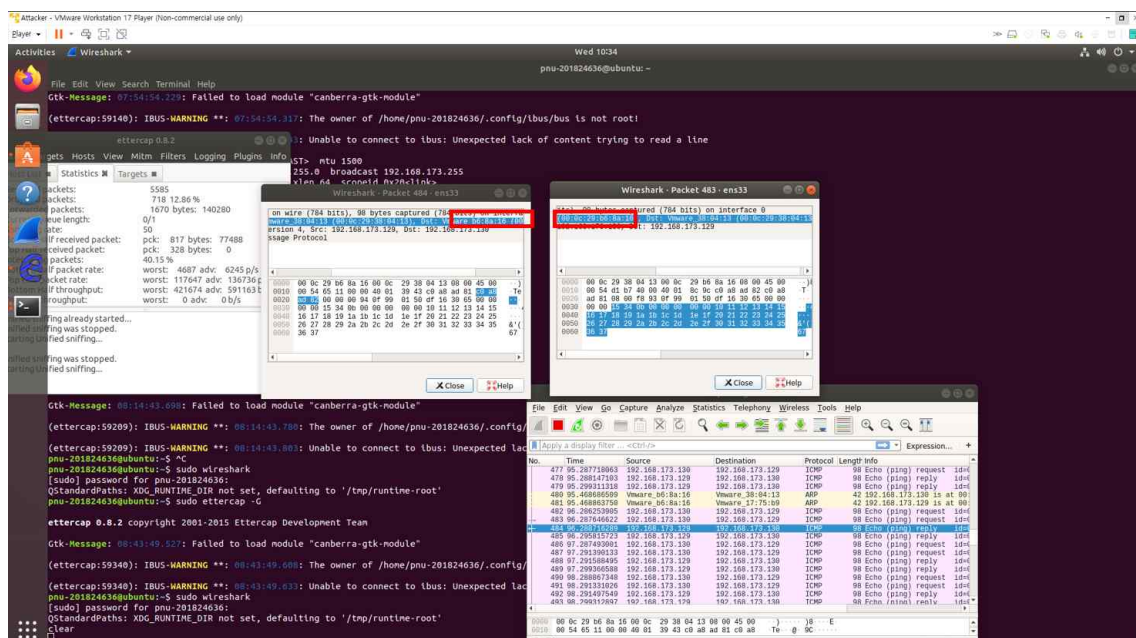
ettercap을 작동하지 않고 Client에서 Server로 ping을 보냈을 때, server -> client, client -> server의 MAC Address를 보면 본래의 설정대로 잘 나오는 것을 볼 수 있다.



ettercap을 이용해 ARP Spoofing을 실행하고 MAC Address Table을 살펴보면 Client와 Server의 MAC Address가 Attacker의 MAC (00:0c:29:b6:8a:16)으로 바뀌는 것을 볼 수 있다. client와 server로 위장하는 데 성공하였다.



Client에서 서버로 ping을 전송하고, Attacker에서 Wireshark를 통해 수집된 패킷을 보면 ARP 요청이 많이 온 것을 확인할 수 있다.



Attacker 노트에서 wireshark를 열어 패킷을 확인해 보면 Client에서 Server, Server에서

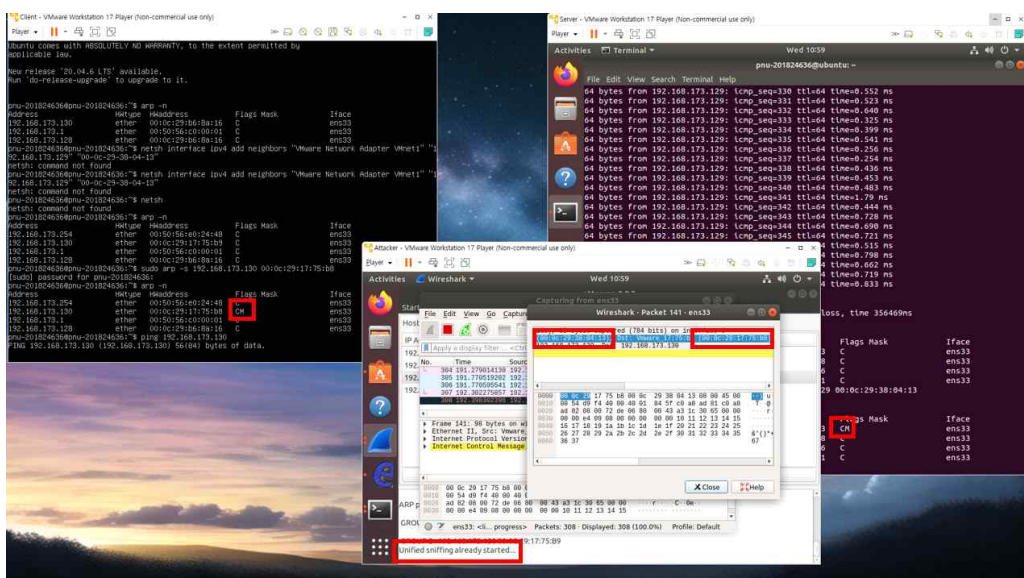
Client로 가는 패킷이 Attacker의 MAC Address인 (00:0c:29:b6:8a:16)를 경유하는 것을 확인할 수 있다. 즉, Attacker가 ARP Spoofing을 통해 중간자 공격에 성공했다는 것이다.

[실습 3] ARP spoofing 방지 기법을 적용

```

Server - VMware Workstation 17 Player (Non-commercial use only)
pnu-201824636@ubuntu: ~
File Edit View Search Terminal Help
64 bytes from 192.168.173.129: icmp_seq=330 ttl=64 time=0.552 ms
64 bytes from 192.168.173.129: icmp_seq=331 ttl=64 time=0.523 ms
64 bytes from 192.168.173.129: icmp_seq=332 ttl=64 time=0.640 ms
64 bytes from 192.168.173.129: icmp_seq=333 ttl=64 time=0.325 ms
64 bytes from 192.168.173.129: icmp_seq=334 ttl=64 time=0.399 ms
64 bytes from 192.168.173.129: icmp_seq=335 ttl=64 time=0.541 ms
64 bytes from 192.168.173.129: icmp_seq=336 ttl=64 time=0.256 ms
64 bytes from 192.168.173.129: icmp_seq=337 ttl=64 time=0.254 ms
64 bytes from 192.168.173.129: icmp_seq=338 ttl=64 time=0.436 ms
64 bytes from 192.168.173.129: icmp_seq=339 ttl=64 time=0.453 ms
64 bytes from 192.168.173.129: icmp_seq=340 ttl=64 time=0.483 ms
64 bytes from 192.168.173.129: icmp_seq=341 ttl=64 time=1.79 ms
64 bytes from 192.168.173.129: icmp_seq=342 ttl=64 time=0.444 ms
64 bytes from 192.168.173.129: icmp_seq=343 ttl=64 time=0.728 ms
64 bytes from 192.168.173.129: icmp_seq=344 ttl=64 time=0.690 ms
64 bytes from 192.168.173.129: icmp_seq=345 ttl=64 time=0.721 ms
64 bytes from 192.168.173.129: icmp_seq=346 ttl=64 time=0.515 ms
64 bytes from 192.168.173.129: icmp_seq=347 ttl=64 time=0.798 ms
64 bytes from 192.168.173.129: icmp_seq=348 ttl=64 time=0.602 ms
64 bytes from 192.168.173.129: icmp_seq=349 ttl=64 time=0.719 ms
64 bytes from 192.168.173.129: icmp_seq=350 ttl=64 time=0.833 ms
^C
--- 192.168.173.129 ping statistics ---
350 packets transmitted, 350 received, 0% packet loss, time 356469ms
rtt min/avg/max/mdev = 0.199/0.657/3.729/0.458 ms
pnu-201824636@ubuntu:~$ arp -n
Address HWtype HWaddress Flags Mask Iface
192.168.173.129 ether 00:0c:29:38:04:13 C ens33
192.168.173.254 ether 00:50:56:e0:24:48 C ens33
192.168.173.128 ether 00:0c:29:b6:8a:16 C ens33
192.168.173.1 ether 00:50:56:c0:00:01 C ens33
pnu-201824636@ubuntu:~$ sudo arp -s 192.168.173.129 00:0c:29:38:04:13
[sudo] password for pnu-201824636:
pnu-201824636@ubuntu:~$ arp -n
Address HWtype HWaddress Flags Mask Iface
192.168.173.129 ether 00:0c:29:38:04:13 CA ens33
192.168.173.254 ether 00:50:56:e0:24:48 C ens33
192.168.173.128 ether 00:0c:29:b6:8a:16 C ens33
192.168.173.1 ether 00:50:56:c0:00:01 C ens33
pnu-201824636@ubuntu:~$
  
```

ARP Spoofing은 MAC과 IP주소를 바꾸지 못하게 정적으로 설정하면 방지할 수 있다. 그림을 보면, Client 노드 (00:0c:29:38:04:13) Flag를 보면 arp -s를 통해 정적으로 설정하였다. MAC과 IP주소를 고정해 공격자가 도중에 끼어들 수 없게 방지하였다.



ettercap을 이용한 ARP spoofing이 작동되고 있고 클라이언트가 서버로 ping을 보내고 있지만, MAC Address를 정적으로 고정했기 때문에 Client 노드 (00:0c:29:38:04:13)가 Server 노드 (00:0c:29:17:75:b9)로 이동할 때 Attacker (00:0c:29:b6:8a:16)의 MAC Address를 경유하지 않는다.