동국대학교
dongguk university

제출일	2023.05.16	학과	컴퓨터공학전공
과목	컴퓨터보안	학번	2018112007
담당교수	김영부 교수님	이름	이승현

1) 실습 환경

(1)
운영 체제: Microsoft Windows 11 Home 64bit
프로세서 : Intel(R) Core(TM) i7-10510U @ 1.80GHz (8 CPUs), ~ 2.3GHz
메모리 : DDR4 16GB 2,667MHz
그래픽 카드 : Intel UHD Graphics

(2)
운영 체제: Microsoft Windows 10 Home 64bit
프로세서 : Intel(R) Core(TM) i7-7700HQ @ 2.80GHz (8 CPUs), ~ 2.8GHz
메모리 : DDR4 8GB 2,133MHz
그래픽 카드 : Intel HD Graphics 630, NVIDIA GeForce GTX 1050

2) 실습 진행

1. 문제 분석

i. 스노트(Snort) 윈도우 시스템 구축하기

- 스노트(Snort)는 네트워크 침입 차단 시스템(NIPS:Network Intrusion Prevention System)이자, 동시에 네트워크 침입 탐지 시스템(NIDS: Network Intrusion Detection System)인 오픈소스 프로그램으로서, 마틴로시가 1998년에 개발하였다.
- 스노트의 네트워크 기반 침입 탐지 시스템(NIDS)은 실시간 트래픽 분석과 IP에서의 패킷 로깅을 수행하는 능력을 가지며, 프로토콜 분석, 내용 검색 그리고 매칭을 수행한다.
- 스노트는 조사나 공격을 탐지하는데 사용될 수 있다. 이러한 조사나 공격으로는 공용 TCP/IP 스택 핑거프린팅, 공용 게이트웨이 인터페이스, 버퍼 오버플로, 서버 메시지 블록 조사 그리고 스텔스 포트 스캔 등이 있다.
- 스노트는 3가지의 주요 모드로 설정될 수 있다. 스니퍼, 패킷 로거 그리고 네트워크 침입 탐지 모드가 있다. 스니퍼 모드에서 프로그램은 네트워크 패킷을 읽고 콘솔에 보여준다. 패킷 로거 모드에서 프로그램은 패킷을 디스크에 기록한다. 침입 탐지 모드에서 프로그램은 네트워크 트래픽을 모니터하고 사용자에 의해 정의된 규칙에 반하는지를 분석한다. 프로그램은 그 후 특정한 동작을 수행한다.
- 이번 실습에서는 스노트(Snort) 시스템을 윈도우 기반의 PC에 구축하고, 스노트를 이용해 패킷 정보를 확인해 본다.

ii. 무선 네트워크 분석을 위한 WiFi 탐색툴 Vistumbler 사용하기

- 무선랜(WLAN)은 실제로 스니핑 프로그램을 통해 암호가 걸려있지 않은 백화점 결제 시스템의 AP에 접속하여 결재정보를 살펴보거나 접속 ID를 도용하여 정상적으로 접속한 후 이를 악의적으로 이용하는 경우가 종종 발생하고 있다. 그러나 이러한 사실보다 더 심각한 것은 해커들의 정보 습득 사례가 빈번하더라고 특징적인 증상이 나타나지 않기 때문에 사용자 스스로가 피해 사실을 인지하지 못한다.
- 무선랜이 가진 보안 취약점으로 인해 네트워크 관리 비용은 훨씬 증가하고 있다. 따라서 무선랜의 현상을 분석하고 관리하는 문제는 매우 중요하다. 현재 관리의 목적이 아니더라도 무선랜을 긍정적으로 이용하기 위해 이를 탐색하기 위한 다양한 툴이 존재한다.
- 이번 실습에서는 비즈텀블러(Vistumbler)를 이용하여 활성화된 AP를 스캔해본다.

2. 실습

i. 스노트(Snort) 윈도우 시스템 구축하기

```
# Setup the network addresses you are protecting
ipvar HOME_NET localhost/24
```

- 스노트 최신 버전을 설치한 다음 C:\Snort\etc 폴더로 이동한다.
- snort.conf를 메모장으로 오픈한 다음, 변수 구문 ipvar HOME_NET Any를 해당 네트워크에 맞게 세팅한다.
- 본 실습의 경우 localhost/24로 세팅하였다.

```
# metadata reference data.  do not modify these lines
include c:\snort\etc\classification.config
include c:\snort\etc\reference.config
```

- 문구 include classification.conf를 검색하고 include c:\snort\etc\classification.config로 변경한다.
- 문구 include reference.config를 검색하고 include c:\snort\etc\reference.config로 변경한다.
- 수정한 snort.conf 파일을 저장한다.

```
C:₩Snort₩bin>snort -W

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.20-WIN64 GRE (Build 82)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using PCRE version: 8.10 2010-06-25
           Using ZLIB version: 1.2.11

Index   Physical Address        IP Address      Device Name     Description
-----   ----------------        ----------      -----------     -----------
    1   00:00:00:00:00:00       disabled        ₩Device₩NPF_{861C8970-015C-4ED6-AA57-8409436CE1CE}      NdisWan Adapter
    2   00:00:00:00:00:00       0000:0000:fe80:0000:0000:0000:4c80:9771 ₩Device₩NPF_{57FB0820-DCE6-42A7-A717-F4E08CE2380E}      Microsoft
    3   00:00:00:00:00:00       0000:0000:fe80:0000:0000:0000:dc1f:e253 ₩Device₩NPF_{DB4A61B5-4F5D-4CD6-B155-90E6FB6C25BC}      Microsoft
    4   00:00:00:00:00:00       0000:0000:fe80:0000:0000:0000:1cbb:85c4 ₩Device₩NPF_{A4FF8AC9-8B9B-486A-BFBE-77A7F13DECC4}      Microsoft
    5   00:00:00:00:00:00       0000:0000:fe80:0000:0000:0000:ad51:e6ca ₩Device₩NPF_{E18EC423-B3D3-4E19-AF49-EFAD7355B7A7}      Microsoft
    6   00:00:00:00:00:00       disabled        ₩Device₩NPF_{ED4FC9C2-417E-4070-B8E2-5BB4F5867F58}      NdisWan Adapter
    7   00:00:00:00:00:00       disabled        ₩Device₩NPF_{5E40D117-0EB7-448E-BE32-8BDC3458D883}      NdisWan Adapter
    8   00:00:00:00:00:00       disabled        ₩Device₩NPF_Loopback    Adapter for loopback traffic capture
```

- C:\Snort\bin 폴더로 이동한 다음 snort -W를 입력한다.
- 센서에 설치할 수 있는 어댑터 목록을 확인할 수 있다.
- 유선랜의 경우 Controller 문구가 붙어있다.
- wifi인 경우 Controller 문구가 빠져있으며, Microsoft 문구를 가진 어댑터 중 하나이다.

```
C:\Users\kocan>ipconfig

Windows IP 구성


무선 LAN 어댑터 로컬 영역 연결* 4:

   미디어 상태 . . . . . : 미디어 연결 끊김
   연결별 DNS 접미사. . . . :

무선 LAN 어댑터 로컬 영역 연결* 5:

   미디어 상태 . . . . . : 미디어 연결 끊김
   연결별 DNS 접미사. . . . :

무선 LAN 어댑터 Wi-Fi:

   연결별 DNS 접미사. . . . :
   링크-로컬 IPv6 주소 . . . . : fe80::dc1f:e253:9b8e:6df1%19
   IPv4 주소 . . . . . : 192.168.0.169
   서브넷 마스크 . . . . . : 255.255.255.0
   기본 게이트웨이 . . . . . : 192.168.0.1

이더넷 어댑터 Bluetooth 네트워크 연결:

   미디어 상태 . . . . . : 미디어 연결 끊김
   연결별 DNS 접미사. . . . :
```

- ipconfig 명령어로 windows ip 구성을 확인한 모습이다.
- 현재 랩톱에서 wifi를 이용해 인터넷 통신을 하므로 무선 LAN 어댑터 Wi-Fi를 살펴본다.
- 링크-로컬 IPv6 주소를 이용해 snort -W 목록에 있는 어댑터 중 wifi 어댑터를 선택한다.
- 기본 게이트웨이 주소 192.168.0.1에 ping 명령어를 이용해 패킷을 보내고 확인하는 실습을 가질 것이다.

```
Index   Physical Address        IP Address      Device Name     Description
-----   ----------------        ----------      -----------     -----------
    1   00:00:00:00:00:00       disabled        \Device\NPF_{861C8970-015C-4ED6-AA57-8409436CE1CE}      NdisWan Adapter
    2   00:00:00:00:00:00       0000:0000:fe80:0000:0000:0000:4c80:9771 \Device\NPF_{57FB0820-DCE6-42A7-A717-F4E08CE2380E}     Microsoft
    3   00:00:00:00:00:00       0000:0000:fe80:0000:0000:0000:dc1f:e253 \Device\NPF_{DB4A61B5-4F5D-4CD6-B155-90E6FB6C25BC}     Microsoft
    4   00:00:00:00:00:00       0000:0000:fe80:0000:0000:0000:1cbb:85c4 \Device\NPF_{A4FF8AC9-8B9B-486A-BFBE-77A7F13DECC4}     Microsoft
    5   00:00:00:00:00:00       0000:0000:fe80:0000:0000:0000:ad51:e6ca \Device\NPF_{E18EC423-B3D3-4E19-AF49-EFAD7355B7A7}     Microsoft
    6   00:00:00:00:00:00       disabled        \Device\NPF_{ED4FC9C2-417E-4070-B8E2-5BB4F5867F58}      NdisWan Adapter
    7   00:00:00:00:00:00       disabled        \Device\NPF_{5E40D117-0EB7-446E-BE32-8BDC3458D883}      NdisWan Adapter
    8   00:00:00:00:00:00       disabled        \Device\NPF_Loopback    Adapter for loopback traffic capture

C:\Snort\bin>snort -v -i3
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "\Device\NPF_{DB4A61B5-4F5D-4CD6-B155-90E6FB6C25BC}".
Decoding Ethernet

        --== Initialization Complete ==--

           -*> Snort! <*-
  ,,_      Version 2.9.20-WIN64 GRE (Build 82)
 o"  )~    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
   ''''    Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using PCRE version: 8.10 2010-06-25
           Using ZLIB version: 1.2.11

Commencing packet processing (pid=27928)
WARNING: No preprocessors configured for policy 0.
05/10-11:21:33.047551 210.94.221.215:62889 -> 239.255.255.250:1900
UDP TTL:127 TOS:0x0 ID:26223 IpLen:20 DgmLen:208
Len: 175
```

- 어댑터 중 위에서 본 링크-로컬 IPv4 주소를 가진 어댑터는 index 3번이다.
- 따라서 snort -v -i3 명령어를 통해 스노트를 진행한다.

```
C:\Users\kocan>ping 192.168.0.1

Ping 192.168.0.1 32바이트 데이터 사용:
192.168.0.1의 응답: 바이트=32 시간=7ms TTL=64
192.168.0.1의 응답: 바이트=32 시간=1ms TTL=64
192.168.0.1의 응답: 바이트=32 시간=1ms TTL=64
192.168.0.1의 응답: 바이트=32 시간=1ms TTL=64

192.168.0.1에 대한 Ping 통계:
    패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
왕복 시간(밀리초):
    최소 = 1ms, 최대 = 7ms, 평균 = 2ms
```

- ping 명령어를 이용해 기본 게이트웨이 주소 192.168.0.1에 패킷을 전송하고, 응답을 기다린다.
- 전송한 4개의 패킷에 대해 모두 응답을 받았다.

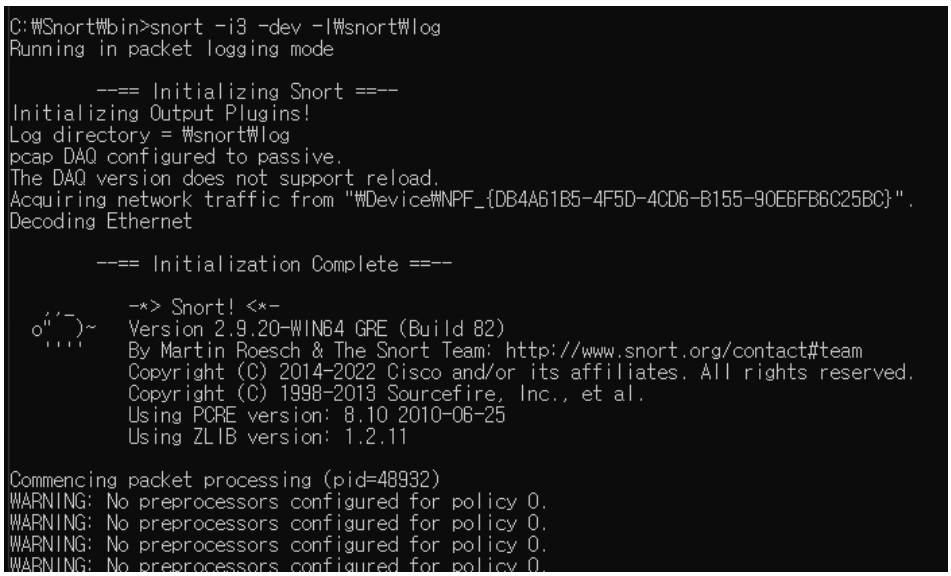
```
C:\Users\kocan>nslookup naver.com
서버:    ns.dgu.ac.kr
Address:  210.94.190.7

권한 없는 응답:
이름:    naver.com
Addresses:  223.130.200.104
          223.130.195.200
          223.130.195.95
          223.130.200.107


C:\Users\kocan>ping 223.130.195.200

Ping 223.130.195.200 32바이트 데이터 사용:
요청 시간이 만료되었습니다.
요청 시간이 만료되었습니다.
요청 시간이 만료되었습니다.
요청 시간이 만료되었습니다.

223.130.195.200에 대한 Ping 통계:
    패킷: 보냄 = 4, 받음 = 0, 손실 = 4 (100% 손실),
```

- 네이버에 대한 ip 주소를 nslookup 명령어를 이용해 확인하였다.
- ping 명령어로 네이버의 ip 주소 223.130.195.200에 패킷을 보내고 응답을 기다렸다.
- 전송한 4개의 패킷에 대하여 응답 요청 시간이 만료되어 응답을 받지 못했다.
- 따라서, 100%의 손실을 결과로 확인하였다.

```
===============================================================================
Run time for packet processing was 93.851000 seconds
Snort processed 35463 packets.
Snort ran for 0 days 0 hours 1 minutes 33 seconds
   Pkts/min:        35463
   Pkts/sec:          381
===============================================================================
Packet I/O Totals:
   Received:        36960
   Analyzed:        35463 ( 95.950%)
    Dropped:         1472 (  3.830%)
   Filtered:            0 (  0.000%)
Outstanding:         1497 (  4.050%)
   Injected:            0
===============================================================================
Breakdown by protocol (includes rebuilt packets):
        Eth:        35463 (100.000%)
       VLAN:            0 (  0.000%)
        IP4:        35457 ( 99.983%)
       Frag:            0 (  0.000%)
       ICMP:           12 (  0.034%)
        UDP:           36 (  0.102%)
        TCP:        35353 ( 99.690%)
        IP6:            2 (  0.006%)
    IP6 Ext:            2 (  0.006%)
   IP6 Opts:            0 (  0.000%)
      Frag6:            0 (  0.000%)
      ICMP6:            0 (  0.000%)
       UDP6:            2 (  0.006%)
       TCP6:            0 (  0.000%)
     Teredo:            0 (  0.000%)
    ICMP-IP:            0 (  0.000%)
      EAPOL:            0 (  0.000%)
    IP4/IP4:            0 (  0.000%)
    IP4/IP6:            0 (  0.000%)
    IP6/IP4:            0 (  0.000%)
    IP6/IP6:            0 (  0.000%)
        GRE:            0 (  0.000%)
    GRE Eth:            0 (  0.000%)
   GRE VLAN:            0 (  0.000%)
    GRE IP4:            0 (  0.000%)
    GRE IP6:            0 (  0.000%)
GRE IP6 Ext:            0 (  0.000%)
   GRE PPTP:            0 (  0.000%)
    GRE ARP:            0 (  0.000%)
    GRE IPX:            0 (  0.000%)
   GRE Loop:            0 (  0.000%)
       MPLS:            0 (  0.000%)
        ARP:            4 (  0.011%)
        IPX:            0 (  0.000%)
   Eth Loop:            0 (  0.000%)
   Eth Disc:            0 (  0.000%)
   IP4 Disc:           55 (  0.155%)
   IP6 Disc:            0 (  0.000%)
   TCP Disc:            0 (  0.000%)
   UDP Disc:            0 (  0.000%)
  ICMP Disc:            0 (  0.000%)
All Discard:           55 (  0.155%)
      Other:            1 (  0.003%)
Bad Chk Sum:         2240 (  6.316%)
    Bad TTL:            0 (  0.000%)
     S5 G 1:            0 (  0.000%)
     S5 G 2:            0 (  0.000%)
      Total:        35463
===============================================================================

Memory Statistics for File at:Wed May 10 13:51:33 2023

Total buffers allocated:          0
Total buffers freed:              0
Total buffers released:           0
Total file mempool:               0
Total allocated file mempool:     0
Total freed file mempool:         0
Total released file mempool:      0

Heap Statistics of file:
         Total Statistics:
              Memory in use:              0 bytes
               No of allocs:              0
                No of frees:              0
===============================================================================
Snort exiting
```

- ctrl + c 입력으로 진행하던 스노트를 중단한다. 스노트를 중단하는 순간 스노트 도중 전송되었던 패킷에 대한 정보들이 출력된다.

- 약 94초 동안 스노트가 진행되었음을 알 수 있다.
- 전체 패킷의 수는 36,960개이며, 이 중 35,463개가 분석되었고 1,497개가 분석하지 않았거나, 분석을 위해 대기하는 패킷이다.
- 1,472개의 패킷이 소실된 모습을 볼 수 있다.
- 분석된 패킷 전체가 Ethernet 프로토콜을 통해 전송된 것을 볼 수 있다. (100%)
- 분석된 패킷 대부분이 IPv4 프로토콜로 전송되었음을 확인할 수 있다. (99.983%)
- ICMP 프로토콜이 12번 전송된 모습을 볼 수 있다. 이는 기본 게이트웨이 주소로 전송할 때 4개의 패킷 각각에 대하여 ICMP가 2번 전송되어(패킷의 전송과 응답) 8번의 ICMP 프로토콜 전송이 있었고, 네이버 ip 주소로 패킷을 보냈을 때 응답이 없었으므로 전송에 대한 4번의 ICMP 프로토콜만 존재하게 된다. 따라서 12번의 ICMP 프로토콜 전송이 존재하게 된다.
- 분석된 패킷 대부분이 TCP 프로토콜로 전송된 모습을 볼 수 있다. (99.690%)
- 4번의 ARP 사용이 있었으며, 55개의 IPv4 프로토콜의 패킷이 손실되었고, 2,240개의 패킷이 변조되어 Bad Check Sum이 발생하였다.



- snort -i3 -dev -l \snort\log 명령어를 통해 스노트를 진행하면서 \snort\log 폴더에 헤더를 기록한다.
- 이전 스노트와 마찬가지로 index 3번에 있는 어댑터를 이용해 진행한다.

```
C:\Users\kocan>ipconfig

Windows IP 구성


무선 LAN 어댑터 로컬 영역 연결* 4:

   미디어 상태 . . . . . : 미디어 연결 끊김
   연결별 DNS 접미사. . . . :

무선 LAN 어댑터 로컬 영역 연결* 5:

   미디어 상태 . . . . . : 미디어 연결 끊김
   연결별 DNS 접미사. . . . :

무선 LAN 어댑터 Wi-Fi:

   연결별 DNS 접미사. . . . :
   링크-로컬 IPv6 주소 . . . . : fe80::dc1f:e253:9b8e:6df1%19
   IPv4 주소 . . . . . : 192.168.1.85
   서브넷 마스크 . . . . . : 255.255.255.0
   기본 게이트웨이 . . . . . : 192.168.1.1

이더넷 어댑터 Bluetooth 네트워크 연결:

   미디어 상태 . . . . . : 미디어 연결 끊김
   연결별 DNS 접미사. . . . :
```

- ipconfig 명령어를 통해 ip 구성을 확인한다.
- 접속한 Wi-Fi를 변경했기 때문에, 기본 게이트웨이가 변경된 모습을 볼 수 있다.

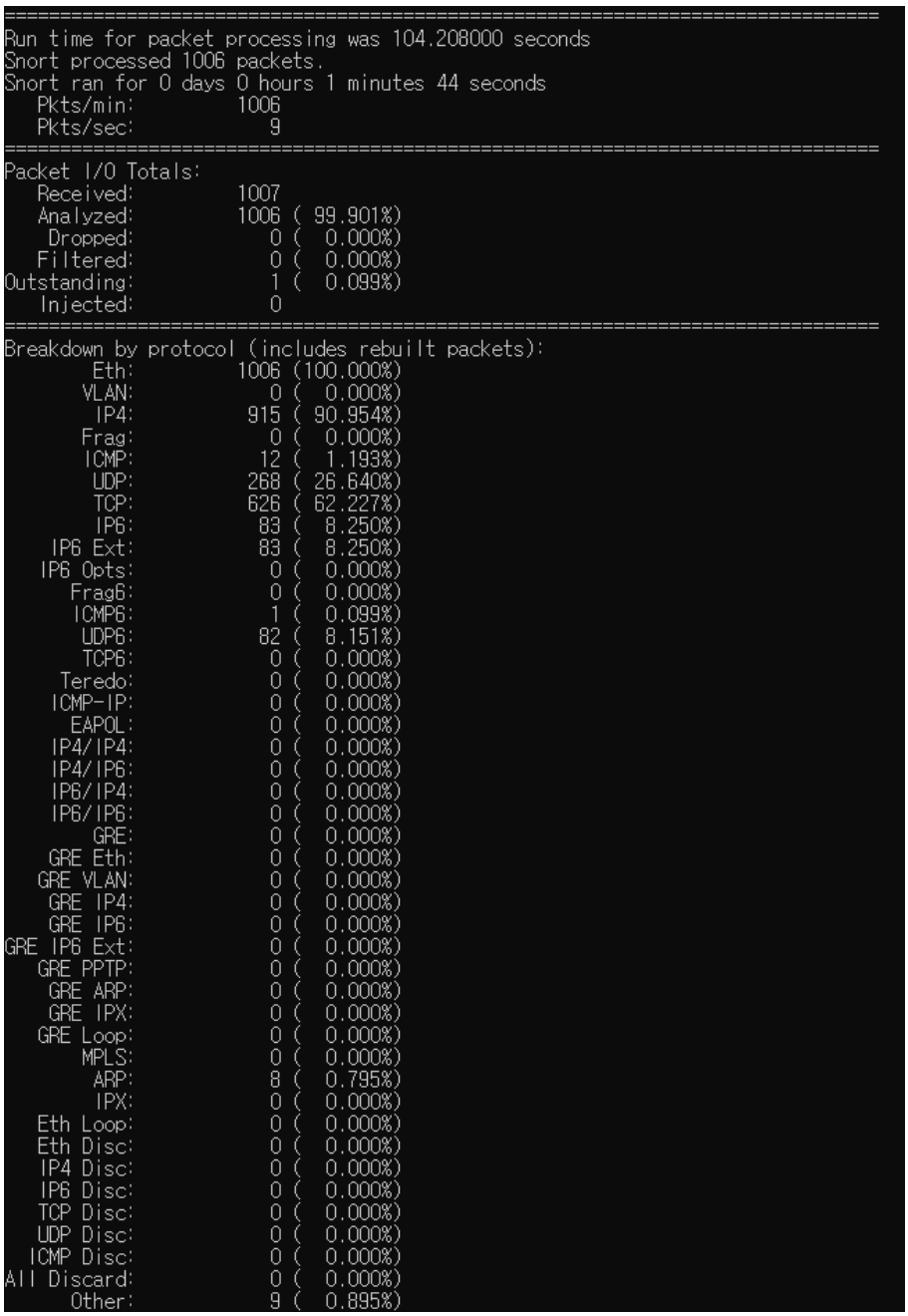
```
C:\Users\kocan>ping 192.168.1.1

Ping 192.168.1.1 32바이트 데이터 사용:
192.168.1.1의 응답: 바이트=32 시간=7ms TTL=64
192.168.1.1의 응답: 바이트=32 시간=9ms TTL=64
192.168.1.1의 응답: 바이트=32 시간=4ms TTL=64
192.168.1.1의 응답: 바이트=32 시간=5ms TTL=64

192.168.1.1에 대한 Ping 통계:
    패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
왕복 시간(밀리초):
    최소 = 4ms, 최대 = 9ms, 평균 = 6ms
```

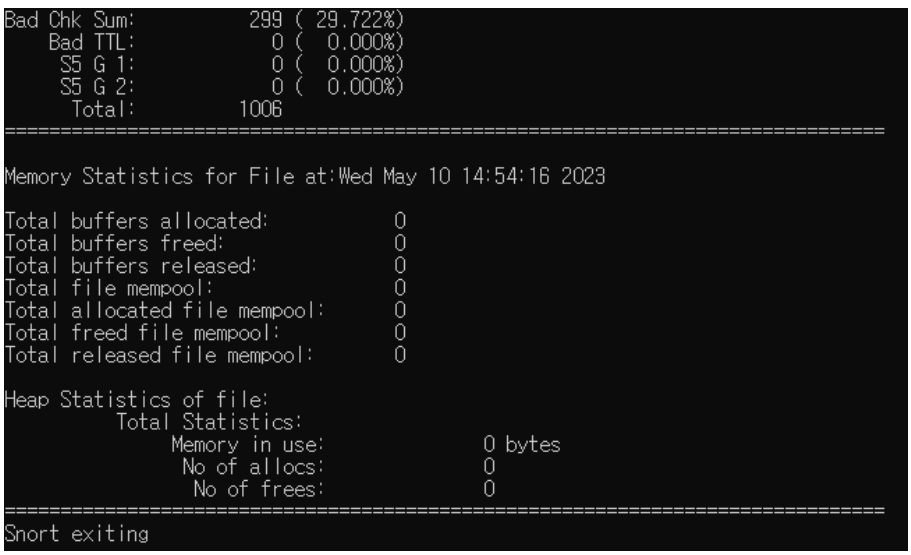
- ping 명령어를 이용해 기본 게이트웨이 주소 192.168.1.1에 패킷을 전송하고, 응답을 기다린다.
- 전송한 4개의 패킷에 대해 모두 응답을 받았다.

```
C:\Users\kocan>ping 223.130.195.200

Ping 223.130.195.200 32바이트 데이터 사용:
요청 시간이 만료되었습니다.
요청 시간이 만료되었습니다.
요청 시간이 만료되었습니다.
요청 시간이 만료되었습니다.

223.130.195.200에 대한 Ping 통계:
    패킷: 보냄 = 4, 받음 = 0, 손실 = 4 (100% 손실),
```

- ping 명령어로 네이버의 ip 주소 223.130.195.200에 패킷을 보내고 응답을 기다렸다.
- 전송한 4개의 패킷에 대하여 응답 요청 시간이 만료되어 응답을 받지 못했다.

- 따라서, 100%의 손실을 결과로 확인하였다.



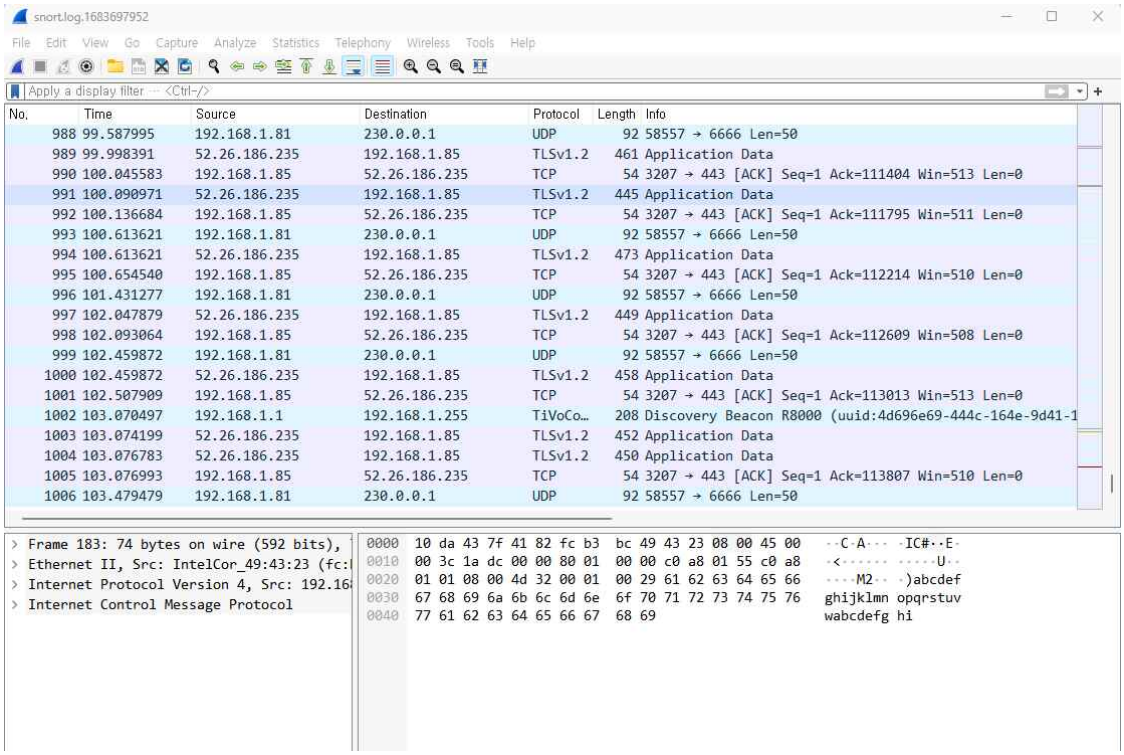


- 앞선 스노트의 결과와 거의 비슷한 모습을 볼 수 있다.

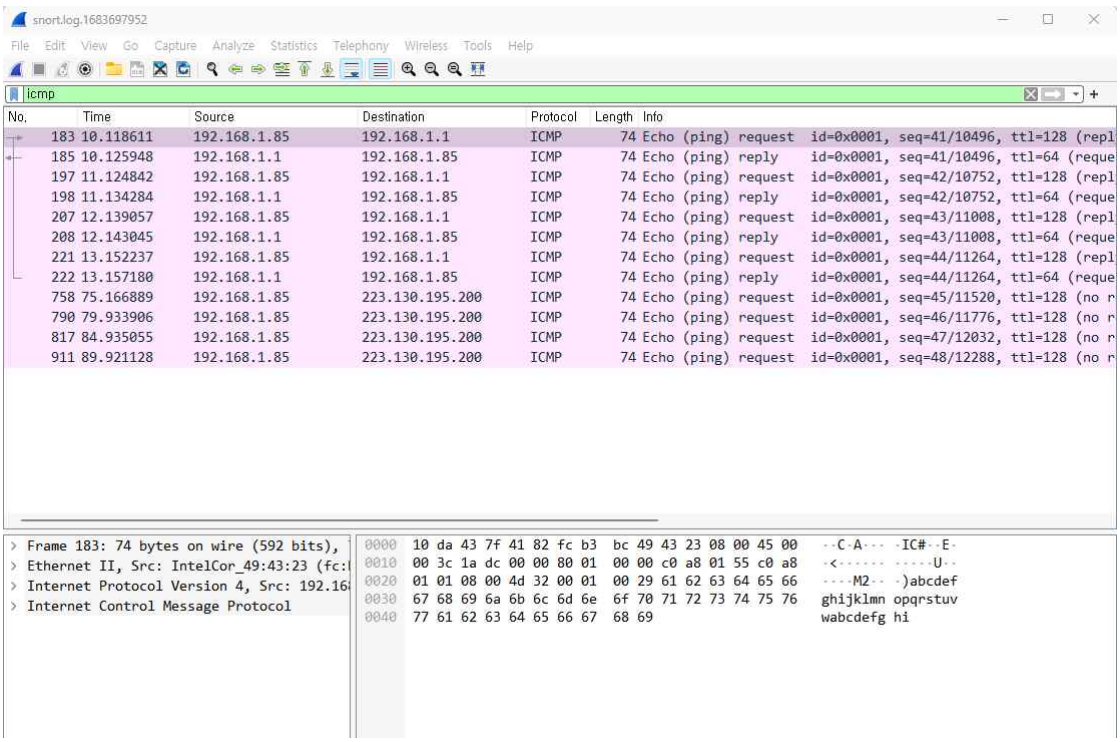
- ctrl + c 입력으로 진행하던 스노트를 중단한다. 스노트를 중단하는 순간 스노트 도중 전송되었던 패킷에 대한 정보들이 출력된다.
- 약 104초 동안 스노트가 진행되었음을 알 수 있다.
- 전체 패킷의 수는 1,007개이며, 이 중 1,006개가 분석되었고 1개가 분석하지 않았거나, 분석을 위해 대기하는 패킷이다.
- 분석된 패킷 전체가 Ethernet 프로토콜을 통해 전송된 것을 볼 수 있다. (100%)
- 분석된 패킷 대부분이 IPv4 프로토콜로 전송되었음을 확인할 수 있다. (90.954%)
- ICMP 프로토콜이 12번 전송된 모습을 볼 수 있다. 이는 기본 게이트웨이 주소로 전송할 때 4개의 패킷 각각에 대하여 ICMP가 2번 전송되어(패킷의 전송과 응답) 8번의 ICMP 프로토콜 전송이 있었고, 네이버 ip 주소로 패킷을 보냈을 때 응답이 없었으므로 전송에 대한 4번의 ICMP 프로토콜만 존재하게 된다. 따라서 12번의 ICMP 프로토콜 전송이 존재하게 된다.
- 분석된 패킷의 절반 이상이 TCP 프로토콜로 전송된 모습을 볼 수 있다. (62.227%)
- 8번의 ARP 사용이 있었으며, 299개의 패킷이 변조되어 Bad Check Sum이 발생하였다.

- \snort\log 폴더에 저장된 로그 파일을 메모장으로 열어보면 인코딩 문제로 글자가 깨지게 된다.

- \snort\log 폴더에 저장된 로그 파일을 vscode로 열어보면 메모장과 마찬가지로 인코딩 문제로 글자가 깨지게 된다.



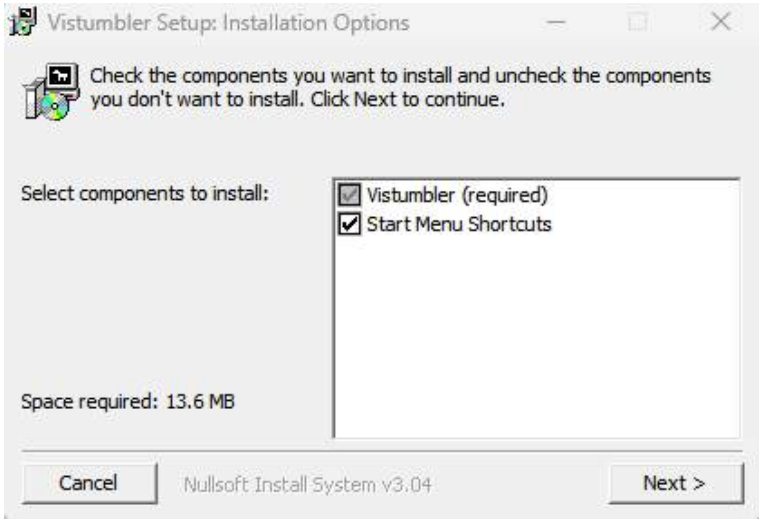
- 와이어샤크로 로그 파일을 열어보면 메모장과 vscode와 달리 캡처된 패킷 정보들이 목록에 출력된다.
- 패킷의 총 개수는 스노트에서 분석된 패킷의 개수와 같다. 스노트에서 분석된 패킷의 개수가 1,006개였는데, 와이어샤크에 출력된 패킷의 개수 또한 1,006임을 위의 이미지를 통해 확인할 수 있다.



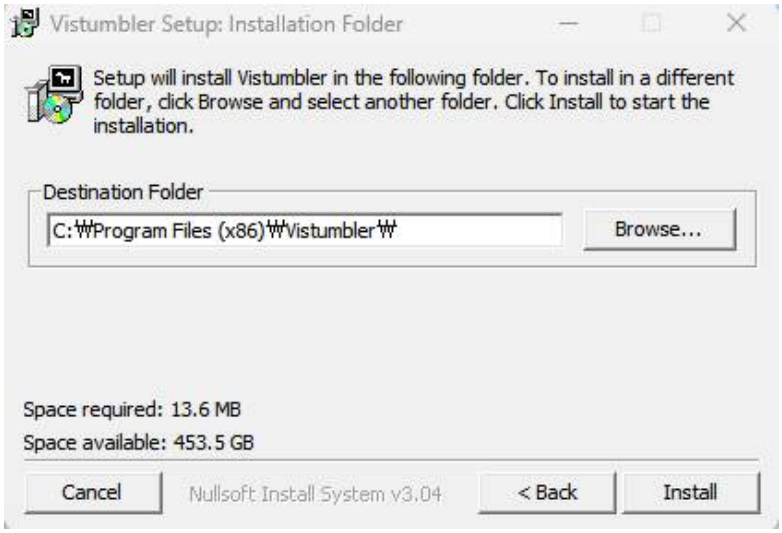
- icmp 프로토콜만 출력되도록 필터링한 모습이다.
- 총 icmp 프로토콜의 개수는 12개이다.
- 앞의 8개는 이 랩톱에서 기본 게이트웨이에 패킷을 4개 전송할 때, 각각의 패킷에 대해서 전송과 응답을 받게 되고, 전송과 응답에 대해 icmp 프로토콜이 전송되어 총 8개가 된다.
- 뒤의 4개는 이 랩톱에서 네이버 ip 주소에 패킷을 4개 전송할 때, 각각의 패킷에 대해서 전송만 하고, 응답은 받지 못하므로, 전송에 대해서만 icmp 프로토콜이 전송되어 총 4개가 된다.
- 따라서 12개의 icmp 프로토콜이 전송되는 것이다.

ii. 무선 네트워크 분석을 위한 WiFi 탐색 투를 Vistumbler 사용하기

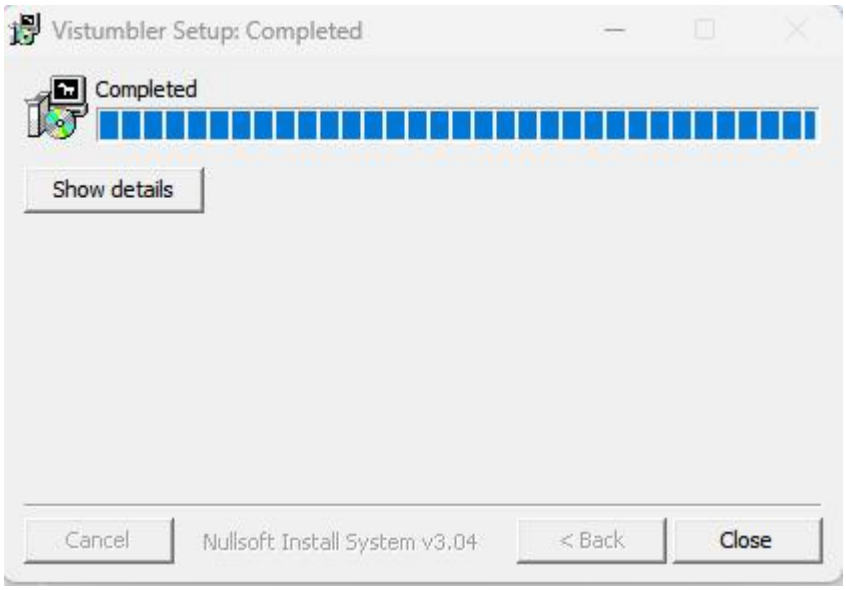
1) 해당 WiFi 탐색 툴이 자신의 노트북에서 제대로 동작하도록 설치파일을 다운로드하여 올바르게 설치한다.



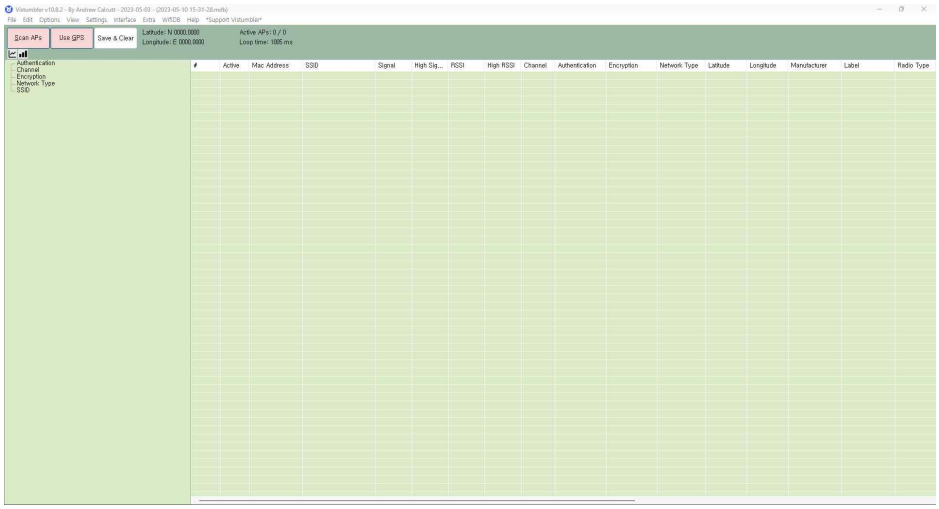
- 설치파일을 실행하고, 시작메뉴에 바로가기를 만들 것인지 선택한다.



- 설치 경로를 선택한다.
- 나는 C:\ProgramFiles(x86) 폴더에 설치하였다.

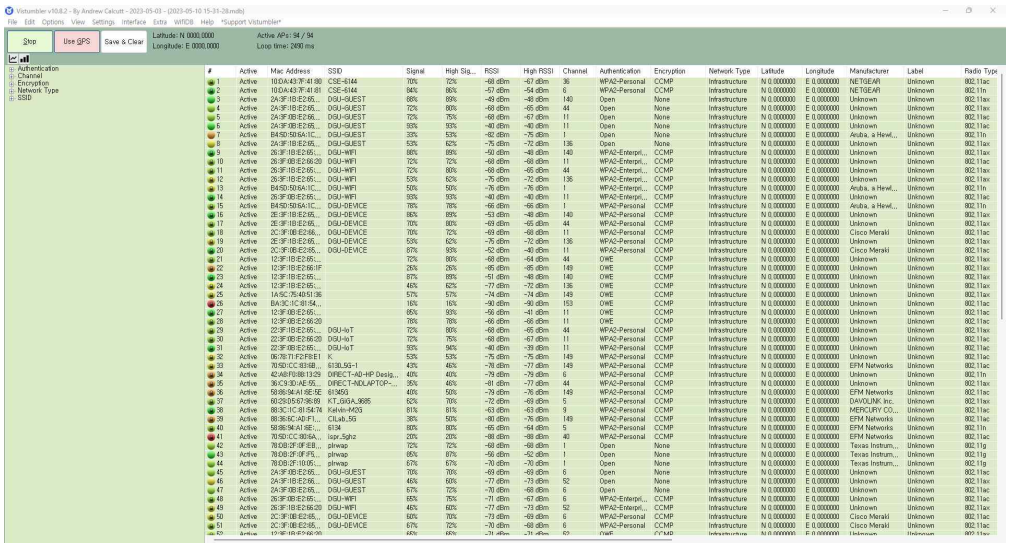


- 설치가 진행되며, 설치가 완료되었을 때 Close 버튼을 클릭하여 종료한다.

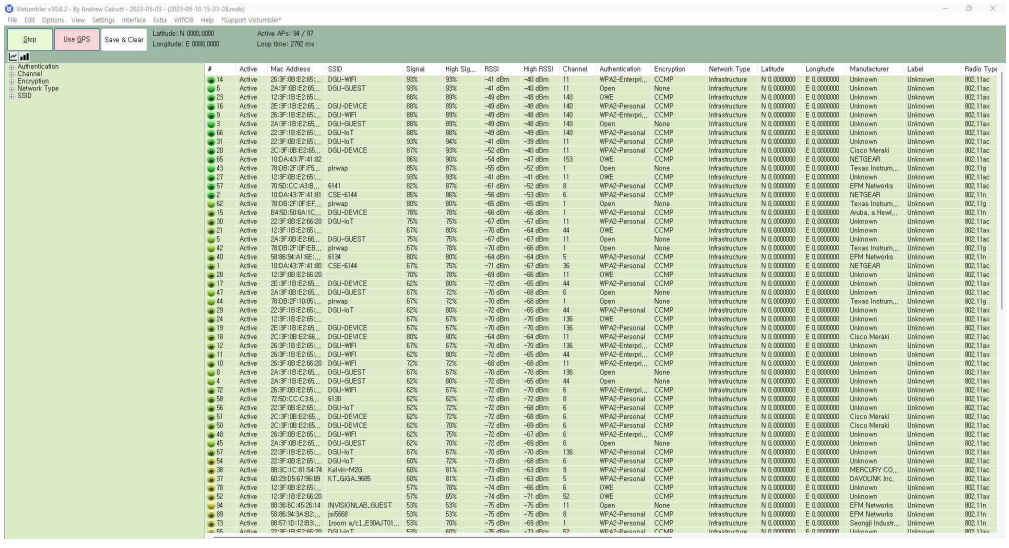


- 설치한 Vistumbler를 실행한 모습이다.
- 초록색 배경을 띠고 있으며, AP를 스캔하지 않았기에 목록에는 아무것도 나타나지 않는다.

2) 설치한 WiFi 탐색 툴을 실행하여 현재 내 주변에 탐지되는 무선랜 SSID는 몇 가지나 있으며 접속하여 사용하기에 가장 신호 상태가 양호한 무선랜은 어떤 것인가?



- 내 주변에 탐지되는 AP를 스캔한 모습이다.
- 현재 94개의 AP가 스캔이 된 모습을 볼 수 있다.



- Signal 컬럼을 클릭하여 신호 세기에 따라 정렬한 모습이다.

- 신호의 세기에 따라 오름차순과 내림차순으로 정렬할 수 있으며, 가장 신호가 양호한 AP를 찾아야 하기에 내림차순으로 정렬하였다.
- 가장 신호가 양호한 AP는 DGU-WIFI, DGU-GUEST이며, 둘 다 93%의 신호 세기를 가지고 있다.

3) 주로 어떤 암호화 기법이 사용되는가? 또 그 기법은 어떠한 기법인가?
- 주로 CCMP를 사용하고 있다.
- CCMP(Cipher Block Chaining Message Authentication Code Protocol)는 Wi-Fi 보안 프로토콜인 WPA2에서 사용되는 암호화 및 인증 방법이다.
- CCMP는 고급 암호화 표준인 AES(Advanced Encryption Standard)를 기반으로 하며, 데이터의 기밀성과 무결성을 보장하기 위해 사용된다.
- CCMP는 다음과 같은 기능을 제공한다.

(1) 데이터 기밀성: CCMP는 AES 알고리즘을 사용하여 Wi-Fi 네트워크에서 데이터를 안전하게 암호화한다. 데이터는 키를 사용하여 암호화되고, 수신 측에서는 해당 키를 사용하여 데이터를 복호화한다. 이를 통해 제3자가 데이터를 엿들을 수 없도록 보호한다.

(2) 데이터 무결성: CCMP는 데이터 무결성을 위해 인증 기능을 제공한다. 데이터를 송신하기 전에 CCMP는 MAC(Message Authentication Code)을 생성하여 데이터에 첨부하고, 수신 측에서 데이터를 수신한 후 동일한 MAC을 계산하여 송신 측이 보낸 MAC와 비교한다. MAC가 일치하지 않으면 데이터의 무결성이 손상되었다는 것을 의미한다.

(3) 리플레이 어택 방지: CCMP는 패킷 리플레이 어택(replay attack)을 방지하기 위해 고유한 시퀀스 번호를 사용한다. 이를 통해 공격자가 이전에 캡처한 패킷을 재전송하여 네트워크를 침해하는 것을 방지한다.

4) 해당 WiFi 탐색 툴이 제공하는 데이터와 기능은 무엇이며 각각 무엇을 의미하는가?
    • Authentication, Channel, Encryption, Network Type, SSID, RSSI 등

- 데이터

Active	해당 무선 네트워크의 활성 여부를 나타낸다.
Mac Address	무선 액세스 포인트(AP)의 MAC 주소를 나타낸다.
SSID	무선 네트워크의 SSID(네트워크 이름)를 나타낸다. SSID는 사용자가 네트워크를 식별할 수 있는 이름이다.
Signal	무선 네트워크의 신호 세기를 % 단위로 나타낸다. 높은 숫자는 더 강한 신호를 나타낸다.
High Signal	현재 무선 네트워크의 신호 세기가 이전에 측정된 최고 신호 세기보다 더 높은지 여부를 나타낸다.
RSSI	Received Signal Strength Indicator의 약자로, 무선 네트워크의 수신 신호 강도를 dBm 단위로 나타낸다. 높은 숫자는 더 강한 신호를 나타낸다.
High RSSI	현재 무선 네트워크의 수신 신호 강도가 이전에 측정된 최고 수신 신호 강도보다 더 높은지 여부를 나타낸다.
Channel	무선 네트워크가 사용하는 무선 채널 번호를 나타낸다.
Authentication	무선 네트워크의 인증 방법을 나타낸다.
Encryption	무선 네트워크의 암호화 유형을 나타낸다.
Network Type	무선 네트워크의 유형을 나타낸다.
Latitude	무선 네트워크 위치의 위도를 나타낸다. 좌표 형식에 따라 다른 형식으로 표시될 수 있다.
Longitude	무선 네트워크 위치의 경도를 나타낸다. 마찬가지로 좌표 형식에 따라 다른 형식으로 표시될 수 있다.
Manufacturer	무선 액세스 포인트(AP)의 제조사 정보를 나타낸다. AP의 MAC 주소를 기반으로 제조사를 식별하여 표시한다.
Label	무선 네트워크에 대한 추가 정보나 레이블을 나타낸다. 사용자가 무선 네트워크에 특정 레이블을 지정할 수 있다.
Radio Type	무선 네트워크의 무선 기술 유형을 나타낸다.
Latitude (DDMMSS)	위도를 도, 분, 초 형식으로 표시한다.
Longitude (DDMMSS)	경도를 도, 분, 초 형식으로 표시한다.
Latitude (DDMMMM)	위도를 도 및 소수점 분 형식으로 표시한다.
Longitude (DDMMMM)	경도를 도 및 소수점 분 형식으로 표시한다.
Basic Transfer Rates	무선 네트워크의 기본 전송 속도를 나타낸다. 이는 네트워크의 최소 전송 속도를 나타낸다.
Other Transfer Rates	무선 네트워크의 기타 전송 속도를 나타낸다. 이는 네트워크가 지원하는 다른 전송 속도를 나타낸다.
First Active	무선 네트워크의 처음으로 활성화된 시간을 나타낸다.
Last Active	무선 네트워크의 마지막으로 활성화된 시간을 나타낸다.

- 기능

(1) 무선 네트워크 탐지: Vistumbler는 주변에 있는 무선 네트워크(AP)를 스캔하여 탐지한다. 이를 통해 무선 네트워크의 신호 세기, SSID(네트워크 이름), MAC 주소, 무선 채널, 보안 설정 등을 확인할 수 있다.

(2) 신호 세기 및 히트맵: Vistumbler는 무선 네트워크의 신호 세기를 표시하고 히트맵으로 시각화한다. 이를 통해 무선 네트워크의 강도와 범위를 시각적으로 확인할 수 있다.

(3) GPS 위치 기능: Vistumbler는 GPS 기능을 지원하여 무선 네트워크의 위치 정보를 수집할 수 있다. 이를 통해 무선 네트워크의 위치와 거리를 확인할 수 있다.

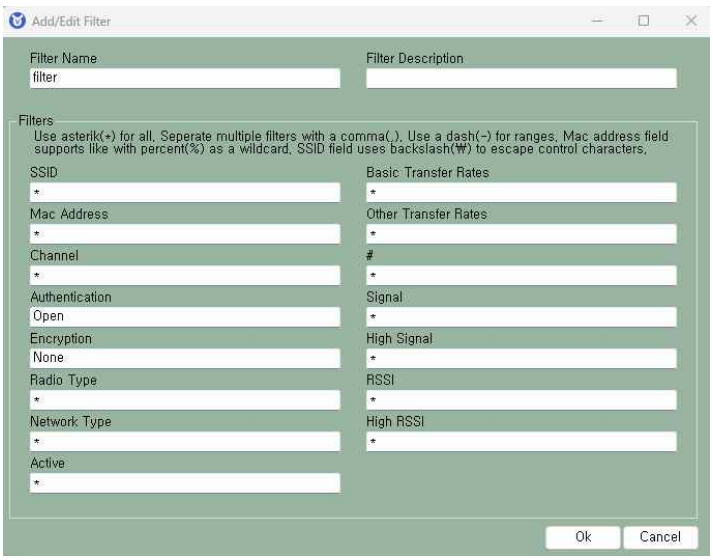
(4) 네트워크 분석: Vistumbler는 무선 네트워크의 보안 설정, 암호화 유형, 무선 채널 사용 여부, 신호 세기 그래프 등을 분석한다. 이를 통해 네트워크의 보안 설정을 평가하고 최적의 무선 채널을 선택할 수 있다.

(5) 파일 내보내기: Vistumbler는 탐지한 무선 네트워크 정보를 CSV(Comma-Separated Values) 파일로 내보낼 수 있다. 이를 통해 나중에 데이터를 분석하거나 다른 도구에서 사용할 수 있다.

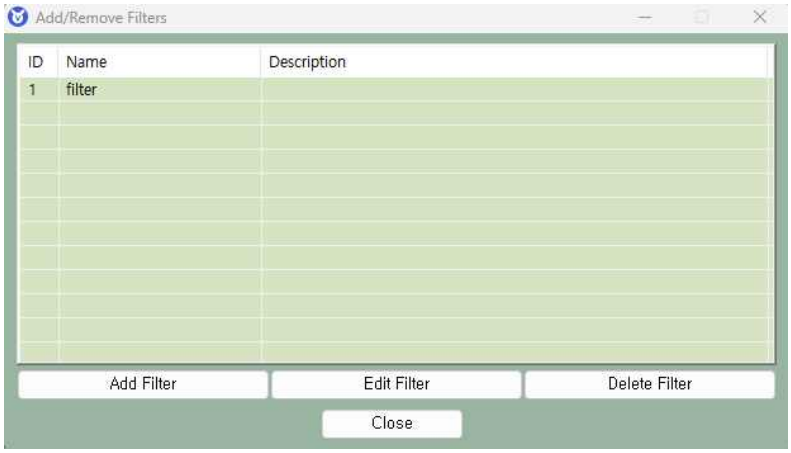
5) 탐지된 무선랜 SSID 중 가장 취약하다고 판단되는 SSID는 무엇이며 그 이유는 무엇인가?

- 가장 취약한 무선랜 SSID를 찾기 위해 필터를 사용하였다.

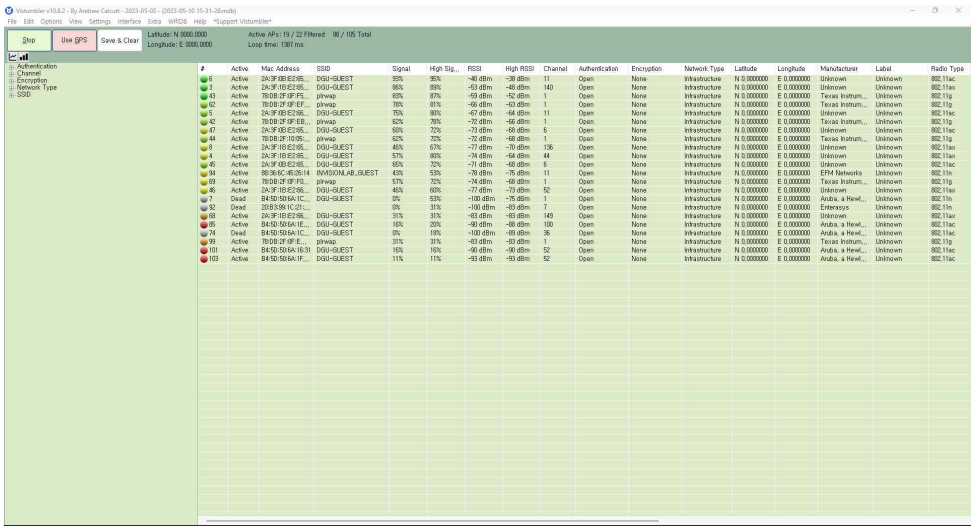
- 메뉴에서 View - Filters - Add/Remove Filters에 들어간다.
- 다음과 같은 화면이 나오며, 필터를 추가하거나 수정, 또는 삭제할 수 있다.



- Add Filter를 클릭한 모습이다. 항목마다 제약 조건을 줄 수 있다.
- Authentication은 Open, Encryption은 None으로 지정한 다음, Filter Name을 입력하고 Ok 버튼을 클릭한다.



- 필터가 추가된 모습이다. 추가된 것을 확인하면 Close 버튼을 클릭하여 화면에서 나간다.



- 메뉴에서 View - Filters - (자기가 만든 필터)를 클릭한다. 클릭한 순간 자신이 지정한 제약 조건에 따라 필터링이 된 결과를 보여준다.
- Authentication은 Open, Encryption은 None인 AP가 22개가 있는 모습을 확인할 수 있다.
- 가장 취약하다고 생각하는 무선랜 SSID는 DGU-GUEST라고 생각한다. 왜냐하면, 탐지된 SSID 중 대부분을 차지하고, 신호도 대체적 강할 정도로 학교에서 누구나 접속하고 이용할 수 있는데, 정작 인증과 암호화가 제대로 이루어지지 않기에 많은 학생의 정보가 중간에 탈취될 위험성이 존재한다.

3) 느낀 점

- 지난주 실습에 이어서 이번 주 실습에는 패킷을 이용한 실습을 진행하게 되었다. 지난주에는 단순히 와이어샤크로 패킷의 정보만 봤다면 이번 실습에는 ping 명령어로 패킷을 전송하고, 스노트로 전송한 패킷의 정보를 확인하거나 로그 파일을 저장하여 와이어샤크에서 열어보고 패킷의 정보를 보는, 좀 더 심화된 실습을 진행하였다. 와이어샤크만큼 패킷의 정보를 자세하게 볼 순 없지만 스노트로도 패킷 전송에 쓰인 프로토콜을 볼 수 있는 만큼 유용한 도구라는 것을 깨닫게 되었다. 그리고 기본 게이트웨이와 네이버 ip 주소에 패킷을 보내고 ICMP 프로토콜을 확인하는 과정에서 나는 원래 ICMP 프로토콜이 패킷 전송에 오류가 발생할 때만 사용되는 줄 알았는데, 패킷을 전송하고 응답받는 과정에서 ICMP 프로토콜이 사용되는 것을 알게 되었다. 그래서 기본 게이트웨이와 네이버 ip 주소에 패킷을 보냈을 때 ICMP 프로토콜이 4번만 사용될 것이라고 처음에 생각했었다. 그러나 직접 기본 게이트웨이에만 보내보고, 네이버 ip 주소에만 보내보고 결과를 비교해본 결과 패킷전송에 기본적으로 ICMP 프로토콜이 사용되고, 상대에게 응답을 받았을 때도 ICMP 프로토콜을 사용하는 것을 깨달았다. 지금이라도 잘못된 지식을 바로 잡아서 다행이라는 생각이 들었고, 개념을 확실히 해야겠다는 생각이 들었다. 이제 벌써 10주차에 접어들었는데, 다음에는 어떤 실습을 하게 될지 궁금해진다.