동국대학교
dongguk university

제출일	2023.05.09	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

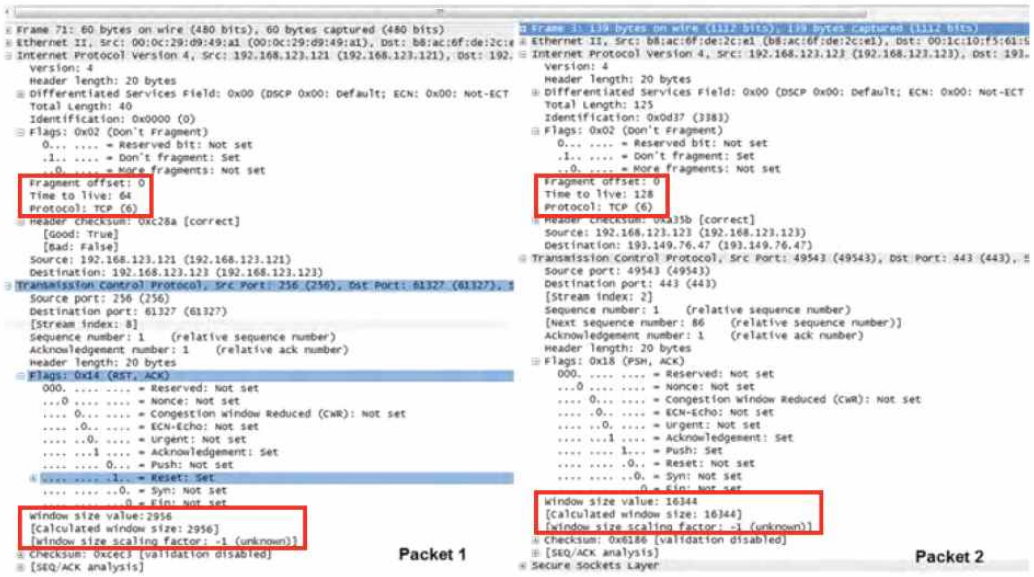
1. 문제 분석

- 와이어샤크는 자유 및 오픈 소스 패킷 분석 프로그램이며 네트워크의 문제, 분석, 소프트웨어 및 통신 프로토콜 개발, 교육에 쓰인다.
- 실시간 네트워크 연결의 유선으로부터 데이터를 포획하고, 이미 포획한 패킷을 기록해둔 파일로부터 데이터를 읽을 수 있다.
- 실시간 데이터를 이더넷, IEEE 802.11, PPP, 루프백을 포함한 수많은 네트워크로부터 읽을 수 있다.
- 이번 실습을 통해서 와이어샤크의 사용법을 익히고, 패킷의 정보를 읽어내서 다양한 문제를 풀어본다.

2. 실습

1) 패킷 가지고 놀기

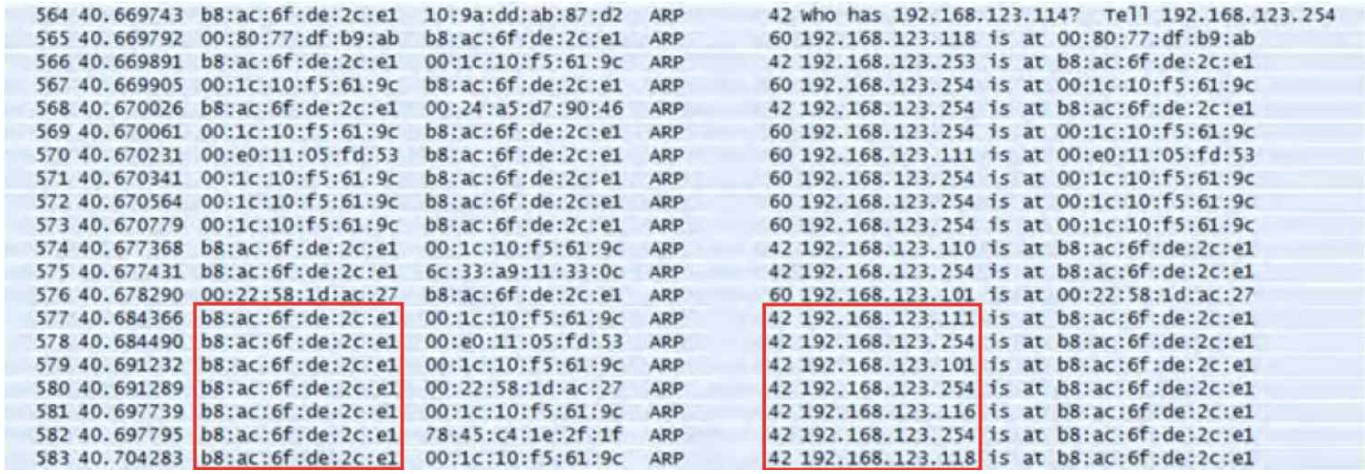
i. 아래 그림을 조사해보고 packet1과 packet2의 운영 체제가 무엇인지 추측해보시오.



- Packet 1에서 TTL이 64이므로 운영 체제가 Linux임을 알 수 있다.
- Packet 2에서 TTL이 128이므로 운영 체제가 windows임을 알 수 있다.

- Linux에서 window size는 5840(kernel 2.4 & 2.6), 5804(kernel 2.4.10)
- Windows에서 window size는 65536(XP), 8192(Vista, 7, Server 2008)
- Linux와 Windows 모두 IPv4에서 단편화를 진행하지 않는다.

ii. 아래의 그림을 조사해보고 보안 문제를 설명하여라.



564	40.669743	b8:ac:6f:de:2c:e1	10:9a:dd:ab:87:d2	ARP	42 who has 192.168.123.114? Tell 192.168.123.254
565	40.669792	00:80:77:df:b9:ab	b8:ac:6f:de:2c:e1	ARP	60 192.168.123.118 is at 00:80:77:df:b9:ab
566	40.669891	b8:ac:6f:de:2c:e1	00:1c:10:f5:61:9c	ARP	42 192.168.123.253 is at b8:ac:6f:de:2c:e1
567	40.669905	00:1c:10:f5:61:9c	b8:ac:6f:de:2c:e1	ARP	60 192.168.123.254 is at 00:1c:10:f5:61:9c
568	40.670026	b8:ac:6f:de:2c:e1	00:24:a5:d7:90:46	ARP	42 192.168.123.254 is at b8:ac:6f:de:2c:e1
569	40.670061	00:1c:10:f5:61:9c	b8:ac:6f:de:2c:e1	ARP	60 192.168.123.254 is at 00:1c:10:f5:61:9c
570	40.670231	00:e0:11:05:fd:53	b8:ac:6f:de:2c:e1	ARP	60 192.168.123.111 is at 00:e0:11:05:fd:53
571	40.670341	00:1c:10:f5:61:9c	b8:ac:6f:de:2c:e1	ARP	60 192.168.123.254 is at 00:1c:10:f5:61:9c
572	40.670564	00:1c:10:f5:61:9c	b8:ac:6f:de:2c:e1	ARP	60 192.168.123.254 is at 00:1c:10:f5:61:9c
573	40.670779	00:1c:10:f5:61:9c	b8:ac:6f:de:2c:e1	ARP	60 192.168.123.254 is at 00:1c:10:f5:61:9c
574	40.677368	b8:ac:6f:de:2c:e1	00:1c:10:f5:61:9c	ARP	42 192.168.123.110 is at b8:ac:6f:de:2c:e1
575	40.677431	b8:ac:6f:de:2c:e1	6c:33:a9:11:33:0c	ARP	42 192.168.123.254 is at b8:ac:6f:de:2c:e1
576	40.678290	00:22:58:1d:ac:27	b8:ac:6f:de:2c:e1	ARP	60 192.168.123.101 is at 00:22:58:1d:ac:27
577	40.684366	b8:ac:6f:de:2c:e1	00:1c:10:f5:61:9c	ARP	42 192.168.123.111 is at b8:ac:6f:de:2c:e1
578	40.684490	b8:ac:6f:de:2c:e1	00:e0:11:05:fd:53	ARP	42 192.168.123.254 is at b8:ac:6f:de:2c:e1
579	40.691232	b8:ac:6f:de:2c:e1	00:1c:10:f5:61:9c	ARP	42 192.168.123.101 is at b8:ac:6f:de:2c:e1
580	40.691289	b8:ac:6f:de:2c:e1	00:22:58:1d:ac:27	ARP	42 192.168.123.254 is at b8:ac:6f:de:2c:e1
581	40.697739	b8:ac:6f:de:2c:e1	00:1c:10:f5:61:9c	ARP	42 192.168.123.116 is at b8:ac:6f:de:2c:e1
582	40.697795	b8:ac:6f:de:2c:e1	78:45:c4:1e:2f:1f	ARP	42 192.168.123.254 is at b8:ac:6f:de:2c:e1
583	40.704283	b8:ac:6f:de:2c:e1	00:1c:10:f5:61:9c	ARP	42 192.168.123.118 is at b8:ac:6f:de:2c:e1

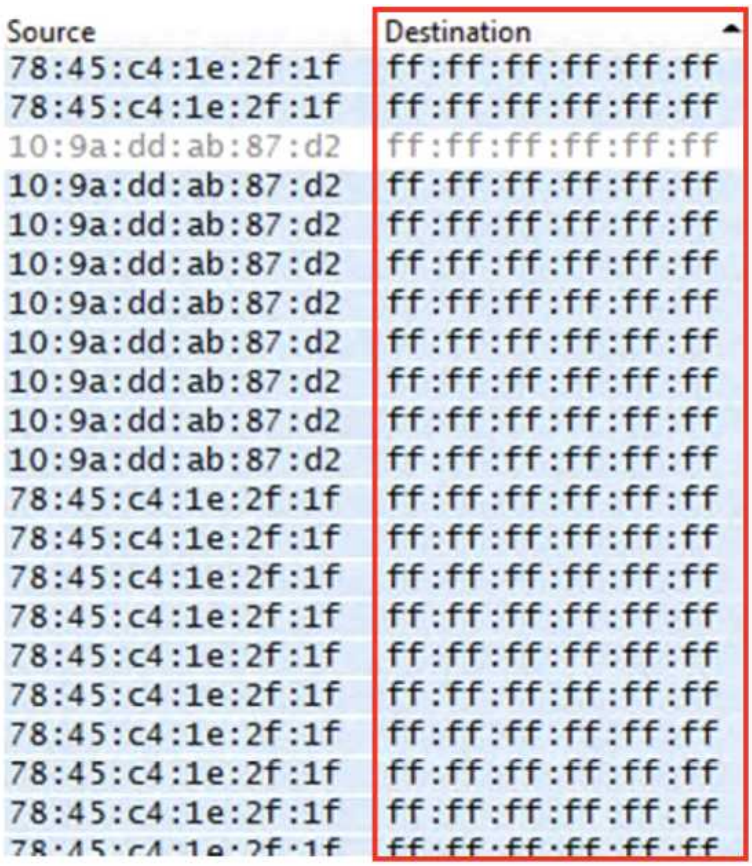
- 이것은 ARP 캐쉬 포이즈닝(cache poisoning) 공격을 캡쳐한 것이다.
- ARP 캐쉬 포이즈닝 기법과 원리에 대해 조사하고, IP 주소 101과 111, 그리고 254 모두가 어떻게 같은 물리 주소인 b8:ac:6f:de:2c:e1 을 가질 수 있게 되었는지 추측하여 발생할 수 있는 문제점에 대해 설명하시오.

ARP 캐시 포이즈닝 공격은 ARP 요청을 이용하여 해당 호스트의 MAC 주소를 조작해서 ARP 캐시에 등록하고, 조작한 ARP 캐시를 토대로 중간에서 패킷을 가로채거나, 해당 패킷을 조작하는 것이다.

ARP 캐시 포이즈닝 기법은 이런 과정을 거친다.
1. 공격자는 ARP 프로토콜을 이용하여 IP 주소와 MAC 주소를 생성한다.
2. 공격자는 생성한 IP 주소와 MAC 주소를 사용하여 네트워크상의 다른 호스트들에게 ARP 요청 패킷을 보낸다.
3. 네트워크상의 다른 호스트들은 ARP 요청 패킷에 포함된 IP 주소와 MAC 주소를 자신의 ARP 캐시에 저장하고, 갱신한다.
4. 공격자는 ARP 캐시에 저장된 IP 주소와 MAC 주소를 이용하여 공격 대상 호스트로부터 수신되는 패킷을 가로채고, 해당 패킷을 원하는 대상으로 전달한다.

서로 다른 IP 주소가 같은 물리 주소를 가지는 이유는 ARP 스푸핑을 이용하여 ARP 테이블에서 IP 주소 101, 111, 254와 연결된 MAC 주소를 공격자의 MAC 주소로 위장하기 때문이다.
그래서 101, 111, 254에 보내는 패킷은 모두 b8:ac:6f:de:2c:e1로 변환되어 공격자에게 전달되고, 공격자는 패킷을 가로채거나 조작할 수 있다.

iii. 아래 그림에서 볼 수 있듯이 새로운 보안 담당자에게 사용되지 않는 스위치의 포트에서 네트워크 캡처를 설정하도록 요청하였다. 활성화된 네트워크상에서 몇 시간의 캡처를 한 후에 확인해보니, 캡처된 모든 것이 브로드 캐스트 트래픽이었다. 무엇이 문제인가?



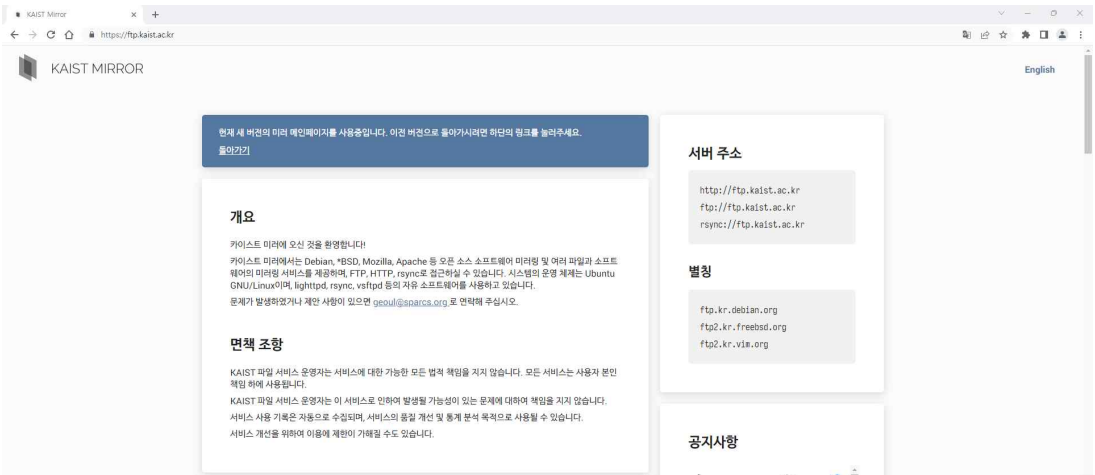
Source	Destination
78:45:c4:1e:2f:1f	ff:ff:ff:ff:ff:ff
78:45:c4:1e:2f:1f	ff:ff:ff:ff:ff:ff
10:9a:dd:ab:87:d2	ff:ff:ff:ff:ff:ff
10:9a:dd:ab:87:d2	ff:ff:ff:ff:ff:ff
10:9a:dd:ab:87:d2	ff:ff:ff:ff:ff:ff
10:9a:dd:ab:87:d2	ff:ff:ff:ff:ff:ff
10:9a:dd:ab:87:d2	ff:ff:ff:ff:ff:ff
10:9a:dd:ab:87:d2	ff:ff:ff:ff:ff:ff
10:9a:dd:ab:87:d2	ff:ff:ff:ff:ff:ff
10:9a:dd:ab:87:d2	ff:ff:ff:ff:ff:ff
10:9a:dd:ab:87:d2	ff:ff:ff:ff:ff:ff
78:45:c4:1e:2f:1f	ff:ff:ff:ff:ff:ff
78:45:c4:1e:2f:1f	ff:ff:ff:ff:ff:ff
78:45:c4:1e:2f:1f	ff:ff:ff:ff:ff:ff
78:45:c4:1e:2f:1f	ff:ff:ff:ff:ff:ff
78:45:c4:1e:2f:1f	ff:ff:ff:ff:ff:ff
78:45:c4:1e:2f:1f	ff:ff:ff:ff:ff:ff
78:45:c4:1e:2f:1f	ff:ff:ff:ff:ff:ff
78:45:c4:1e:2f:1f	ff:ff:ff:ff:ff:ff
78:45:c4:1e:2f:1f	ff:ff:ff:ff:ff:ff
78:45:c4:1e:2f:1f	ff:ff:ff:ff:ff:ff

- 포트가 확장되지 않는다.
- 스위치는 링크 계층에서 작동한다.
- 스위치는 여러 개의 포트가 존재하고, 각각 컴퓨터에 연결된다.
- 스위치는 연결된 각 컴퓨터의 MAC 주소를 알 수 있으며, 대상 컴퓨터로만 프레임이 전달한다.
- 스위치의 각 포트는 연결된 세크먼트에 있는 머신의 MAC 주소를 학습하는데, 알 수 없는 MAC 주소에 대한 Fragments는 브로드캐스트 된다. 따라서 계속 포트가 확장되지 않는다면 패킷을 브로드캐스트할 수밖에 없다.

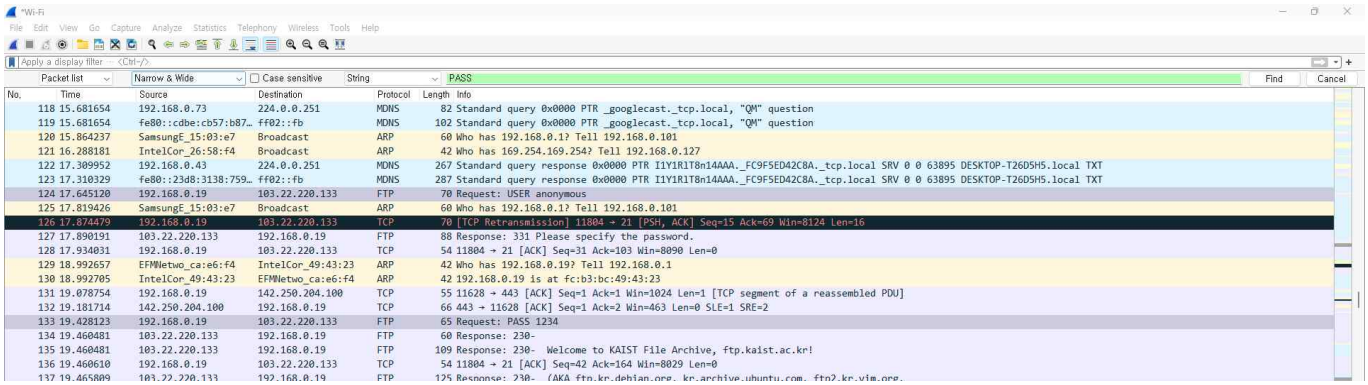
2) Wireshark를 이용한 패킷 분석

```
C:\Users\kocan>ftp ftp.kaist.ac.kr
ftp.sparcs.org에 연결되었습니다.
220 KAIST File Archive (ftp.kaist.ac.kr)
200 Always in UTF8 mode.
사용자(ftp.sparcs.org:(none)): anonymous
331 Please specify the password.
암호:
230-
230-  Welcome to KAIST File Archive, ftp.kaist.ac.kr!
230-  (AKA ftp.kr.debian.org, kr.archive.ubuntu.com, ftp2.kr.vim.org,
230-   ftp2.kr.freebsd.org)
230-
230-  We provide mirrors of open source softwares, e.g. Debian, *BSDs,
230-  Mozilla, Apache, etc. and publically available files and software.
230-  Various access methods are available: FTP, HTTP, Rsync.
230-  This system is running at SPARCS Room, KAIST, Daejeon, Korea, Asia.
230-  We are operating 18TiB RAID-6 storage on Dell PowerEdge R510
230-  (Xeon E5506, 6GiB RAM, 2 x 1Gbps) server. KAIST sponsored hardware and
230-  network connectivity. SPARCS operates the whole service.
230-
230-  Use entirely at your own risk -- no warranty is expressed or implied.
230-  * None of the service providers in any way whatsoever can be
230-    responsible for any problems that might be caused by this service.
230-  * Every access to this service is recorded and can be used and
230-    published for the purpose of improving the quality of the service.
230-  * We may limit any accesses without forewarning that may prevent
230-    operators from maintaining reasonable quality of the service.
230-
230-  Contact ftp@ftp.kaist.ac.kr for any problem or suggestion.
230-  For more information, visit: http://ftp.kaist.ac.kr/
230-
230 Login successful.
```

- ftp 프로토콜을 이용하여 ftp.kaist.ac.kr에 접속한 모습이다.
- 해당 사이트에 처음 접속하게 되면 사용자 명을 입력하게 된다.
- 사용자 명을 입력하면 패스워드를 입력한다.
- 패스워드까지 입력하면 로그인 성공 메시지가 출력된다.



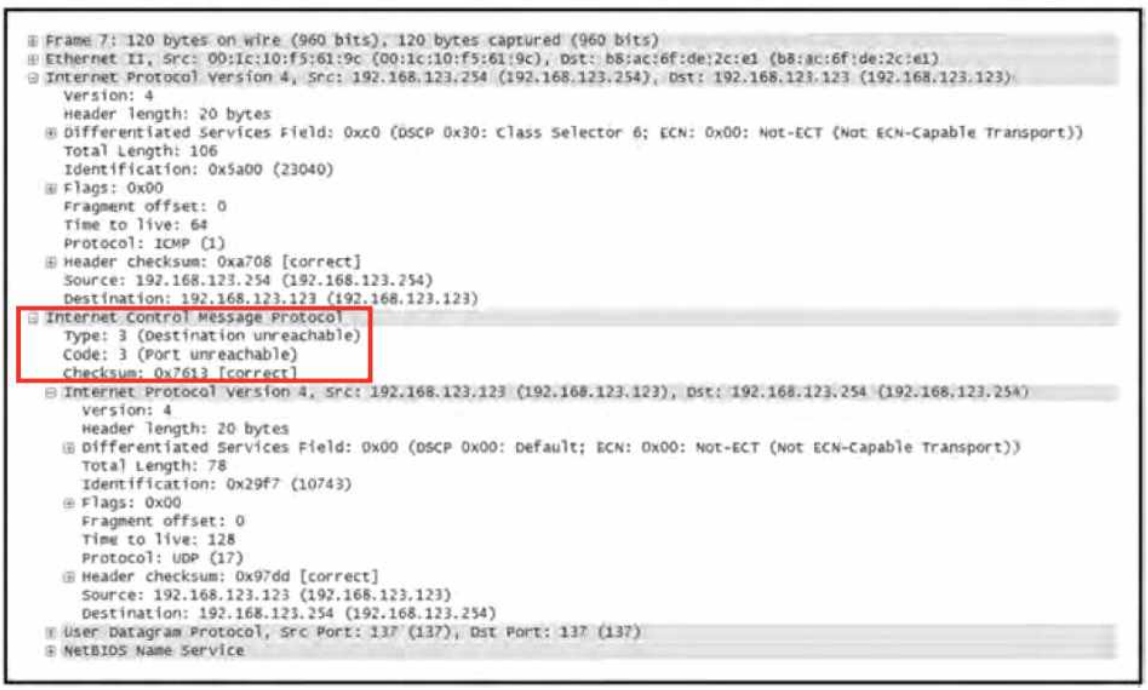
- ftp.kaist.ac.kr을 인터넷 브라우저에서 접속한 모습이다.
- 해당 사이트는 Debian, *BSD, Mozilla, Apache 등 오픈 소스 소프트웨어 미러링 및 여러 파일과 소프트웨어의 미러링 서비스를 제공하며, FTP, HTTP, rsync로 접근할 수 있다.
- 시스템의 운영 체제는 Ubuntu GNU/Linux이며, lighttpd, rsync, vsftpd 등의 자유 소프트웨어를 사용하고 있다.



- 패킷 캡처를 중지하고 String 검색을 통해 위에서 입력한 USER와 PASS의 정보를 확인한 모습이다.
- 내가 입력한 값이 그대로 패킷을 통해 쉽게 확인할 수 있었다.

- FTP는 패스워드가 평문으로 전송하기 때문에, 보안성이 매우 떨어진다. 따라서 SFTP(SSH 프로토콜을 기반으로 하여 암호화된 채널을 통해 파일 전송) 또는 FTPS(SSL/TLS 프로토콜을 사용하여 암호화된 채널을 통해 파일 전송)를 사용하여 파일을 전송한다.
- 또는 VPN을 사용하여 데이터 기밀성, 무결성, 인증을 보장한다.

3) ICMP 패킷 디코딩 수행



i. 문제가 발견된 호스트의 IP 주소는 무엇인가?
192.168.123.123

ii. 원래 호스트의 IP 주소는 무엇인가?
192.168.123.254

iii. ICMP 유형과 코드 에러는 무엇인가?
Type: 3 (Destination unreachable)
Code: 3 (Port unreachable)

iv. 송신 장치는 어느 포트를 가지고 통신을 시도하고 있는가?
137번

v. 구체적인 문제는 무엇인가?

Destination port에 접근할 수 없는 문제이다. 해당 포트에 대한 접근이 불가능하기에, 해당 포트로 접근하는 서비스에 대한 접근이 불가능하고, 해당 서비스를 이용하는 사용자는 서비스를 이용할 수 없게 된다.

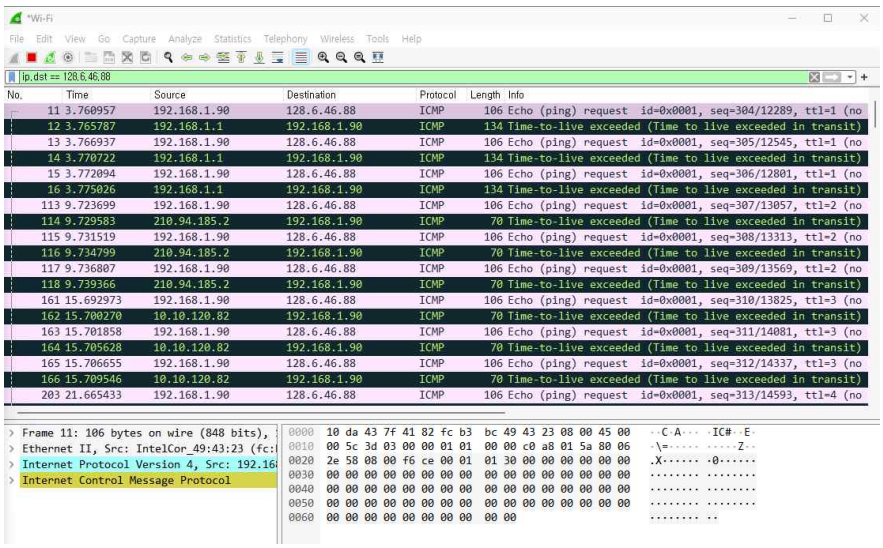
또한 이런 문제가 발생한다면 네트워크 연결에 문제가 있을 가능성이 크므로, 네트워크를 점검하고, 보수해야 한다.

4) 트레이스 라우트(Trace Route)

i. www.rutgers.edu로 향하는 경로를 찾기 위해 Traceroute를 사용한다. 이 Traceroute가 실행되는 동안 와이어샤크를 사용하는 것이 좋다.

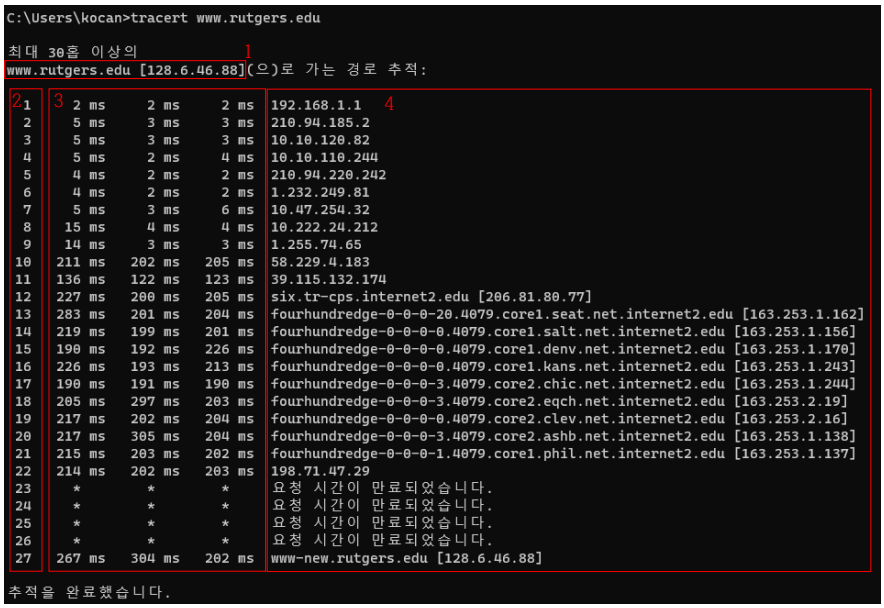
```
C:\Users\kocan>tracert www.rutgers.edu

최대 30홉 이상의
www.rutgers.edu [128.6.46.88](으)로 가는 경로 추적:

  1     2 ms     2 ms     2 ms  192.168.1.1
  2     5 ms     3 ms     3 ms  210.94.185.2
  3     5 ms     3 ms     3 ms  10.10.120.82
  4     5 ms     2 ms     4 ms  10.10.110.244
  5     4 ms     2 ms     2 ms  210.94.220.242
  6     4 ms     2 ms     2 ms  1.232.249.81
  7     5 ms     3 ms     6 ms  10.47.254.32
  8    15 ms     4 ms     4 ms  10.222.24.212
  9    14 ms     3 ms     3 ms  1.255.74.65
 10   211 ms   202 ms   205 ms  58.229.4.183
 11   136 ms   122 ms   123 ms  39.115.132.174
 12   227 ms   200 ms   205 ms  six.tr-cps.internet2.edu [206.81.80.77]
 13   283 ms   201 ms   204 ms  fourhundredge-0-0-0-20.4079.core1.seat.net.internet2.edu [163.253.1.162]
 14   219 ms   199 ms   201 ms  fourhundredge-0-0-0-0.4079.core1.salt.net.internet2.edu [163.253.1.156]
 15   190 ms   192 ms   226 ms  fourhundredge-0-0-0-0.4079.core1.denv.net.internet2.edu [163.253.1.170]
 16   226 ms   193 ms   213 ms  fourhundredge-0-0-0-0.4079.core1.kans.net.internet2.edu [163.253.1.243]
 17   190 ms   191 ms   190 ms  fourhundredge-0-0-0-3.4079.core2.chic.net.internet2.edu [163.253.1.244]
 18   205 ms   297 ms   203 ms  fourhundredge-0-0-0-3.4079.core2.eqch.net.internet2.edu [163.253.2.19]
 19   217 ms   202 ms   204 ms  fourhundredge-0-0-0-0.4079.core2.clev.net.internet2.edu [163.253.2.16]
 20   217 ms   305 ms   204 ms  fourhundredge-0-0-0-3.4079.core2.ashb.net.internet2.edu [163.253.1.138]
 21   215 ms   203 ms   202 ms  fourhundredge-0-0-0-1.4079.core1.phil.net.internet2.edu [163.253.1.137]
 22   214 ms   202 ms   203 ms  198.71.47.29
 23     *        *        *     요청 시간이 만료되었습니다.
 24     *        *        *     요청 시간이 만료되었습니다.
 25     *        *        *     요청 시간이 만료되었습니다.
 26     *        *        *     요청 시간이 만료되었습니다.
 27   267 ms   304 ms   202 ms  www-new.rutgers.edu [128.6.46.88]

추적을 완료했습니다.
```

- tracert 명령어를 통해 www.rutgers.edu로 향하는 경로를 찾는 모습이다.
- 9홉까지 RTT가 낮게 나왔으나, 10홉부터 RTT가 급격하게 증가한 모습을 볼 수 있다.
- 10홉부터 대륙 간 패킷전송을 하는 것으로 예상된다.
- 23홉부터 26홉까지 요청시간이 만료된 모습을 볼 수 있다.
- 27홉에서 목적지에 도착한 모습을 확인할 수 있다.



- 트레이스 라우트 진행 중 와이어샤크를 통해 패킷을 살펴보았다.
- 패킷을 쉽게 보기 위해 ip.dst == 128.6.46.88 조건식으로 필터를 적용하였다.
- 패킷을 보내고 TTL이 만료되는 과정이 반복된다.

ii. Traceroute 결과에서 보이는 것은 어떤 종류의 정보인가?



```
C:\Users\kocan>tracert www.rutgers.edu

최대 30홉 이상의                1
www.rutgers.edu [128.6.46.88](으)로 가는 경로 추적:

2  3                                          4
 1  2 ms     2 ms     2 ms  192.168.1.1
 2    5 ms     3 ms     3 ms  210.94.185.2
 3    5 ms     3 ms     3 ms  10.10.120.82
 4    5 ms     2 ms     4 ms  10.10.110.244
 5    4 ms     2 ms     2 ms  210.94.220.242
 6    4 ms     2 ms     2 ms  1.232.249.81
 7    5 ms     3 ms     6 ms  10.47.254.32
 8   15 ms     4 ms     4 ms  10.222.24.212
 9   14 ms     3 ms     3 ms  1.255.74.65
10  211 ms   202 ms   205 ms  58.229.4.183
11  136 ms   122 ms   123 ms  39.115.132.174
12  227 ms   200 ms   205 ms  six.tr-cps.internet2.edu [206.81.80.77]
13  283 ms   201 ms   204 ms  fourhundredge-0-0-0-20.4079.core1.seat.net.internet2.edu [163.253.1.162]
14  219 ms   199 ms   201 ms  fourhundredge-0-0-0-0.4079.core1.salt.net.internet2.edu [163.253.1.156]
15  190 ms   192 ms   226 ms  fourhundredge-0-0-0-0.4079.core1.denv.net.internet2.edu [163.253.1.170]
16  226 ms   193 ms   213 ms  fourhundredge-0-0-0-0.4079.core1.kans.net.internet2.edu [163.253.1.243]
17  190 ms   191 ms   190 ms  fourhundredge-0-0-0-3.4079.core2.chic.net.internet2.edu [163.253.1.244]
18  205 ms   297 ms   203 ms  fourhundredge-0-0-0-3.4079.core2.eqch.net.internet2.edu [163.253.2.19]
19  217 ms   202 ms   204 ms  fourhundredge-0-0-0-0.4079.core2.clev.net.internet2.edu [163.253.2.16]
20  217 ms   305 ms   204 ms  fourhundredge-0-0-0-3.4079.core2.ashb.net.internet2.edu [163.253.1.138]
21  215 ms   203 ms   202 ms  fourhundredge-0-0-0-1.4079.core1.phil.net.internet2.edu [163.253.1.137]
22  214 ms   202 ms   203 ms  198.71.47.29
23    *        *        *     요청 시간이 만료되었습니다.
24    *        *        *     요청 시간이 만료되었습니다.
25    *        *        *     요청 시간이 만료되었습니다.
26    *        *        *     요청 시간이 만료되었습니다.
27  267 ms   304 ms   202 ms  www-new.rutgers.edu [128.6.46.88]

추적을 완료했습니다.
```

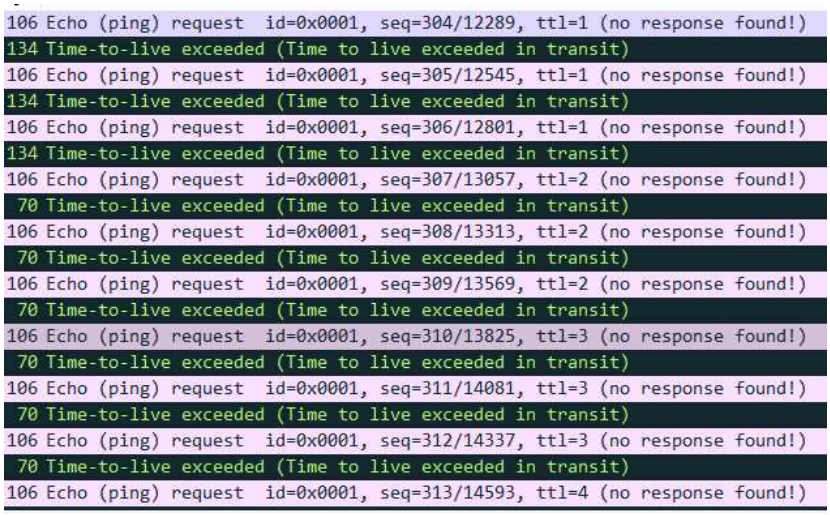
1 : 목적지 주소

2 : 본인의 시스템에서 목적지 주소의 웹 서버까지 가는데 거치는 홉 수

3 : RTT(Round Trip Time), 도착지에 도착하는 데 걸리는 시간

4 : 해당 홉의 호스트 이름과 IP 주소

iii. 와이어샤크를 사용해서 IP헤더에 있는 TTL을 관찰하여라. TTL은 3개의 패킷마다 증가한다. 왜 증가하는가?



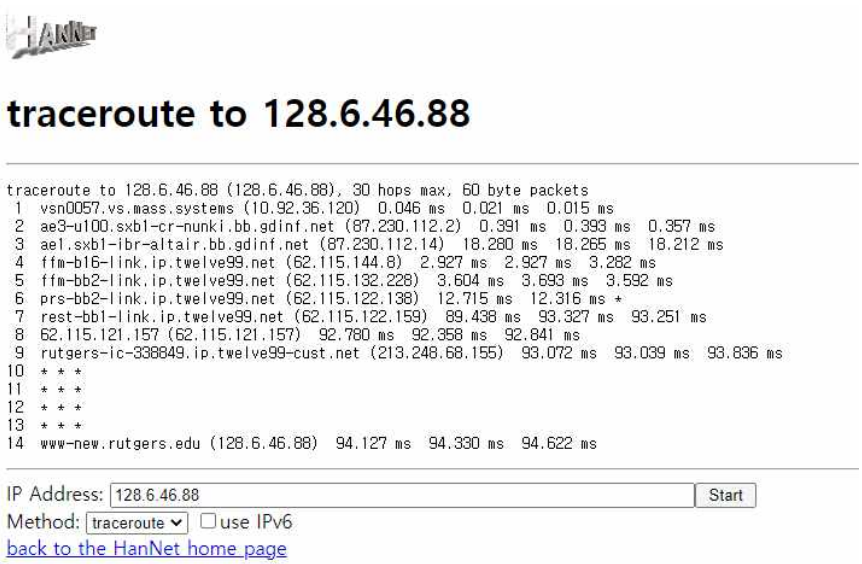
```
106 Echo (ping) request  id=0x0001, seq=304/12289, ttl=1 (no response found!)
134 Time-to-live exceeded (Time to live exceeded in transit)
106 Echo (ping) request  id=0x0001, seq=305/12545, ttl=1 (no response found!)
134 Time-to-live exceeded (Time to live exceeded in transit)
106 Echo (ping) request  id=0x0001, seq=306/12801, ttl=1 (no response found!)
134 Time-to-live exceeded (Time to live exceeded in transit)
106 Echo (ping) request  id=0x0001, seq=307/13057, ttl=2 (no response found!)
 70 Time-to-live exceeded (Time to live exceeded in transit)
106 Echo (ping) request  id=0x0001, seq=308/13313, ttl=2 (no response found!)
 70 Time-to-live exceeded (Time to live exceeded in transit)
106 Echo (ping) request  id=0x0001, seq=309/13569, ttl=2 (no response found!)
 70 Time-to-live exceeded (Time to live exceeded in transit)
106 Echo (ping) request  id=0x0001, seq=310/13825, ttl=3 (no response found!)
 70 Time-to-live exceeded (Time to live exceeded in transit)
106 Echo (ping) request  id=0x0001, seq=311/14081, ttl=3 (no response found!)
 70 Time-to-live exceeded (Time to live exceeded in transit)
106 Echo (ping) request  id=0x0001, seq=312/14337, ttl=3 (no response found!)
 70 Time-to-live exceeded (Time to live exceeded in transit)
106 Echo (ping) request  id=0x0001, seq=313/14593, ttl=4 (no response found!)
```

- 와이어샤크를 사용해서 IP 헤더에 있는 TTL을 확인한 모습이다. 3개의 패킷마다 TTL이 증가한다.
- 3개의 패킷마다 TTL이 증가하는 이유는 홉마다 3개의 UDP 패킷을 전달하여 RTT를 측정하게 된다. 따라서 해당 홉에서 3개의 패킷을 보내어 RTT를 측정하고, 다음 홉으로 진행하게 되는데 홉이 증가할 때마다 TTL을 증가시켜 해당 홉까지만 패킷이 전달될 수 있도록 해야 한다. 따라서 3개의 패킷마다 TTL을 증가시키게 된다.

iv. www.traceroute.org(Germany – HanNet 권장)에서 선택한 공개 Trace route 서버를 사용하여라. 그곳에서 www.rutgers.edu로 Trace route를 실행하여라. 로컬 Trace route의 경로와 traceroute.org에서 수행한 경로가 같은가? 왜 같거나 다른가?

```
C:\Users\kocan>tracert www.rutgers.edu

최대 30홉 이상의
www.rutgers.edu [128.6.46.88](으)로 가는 경로 추적:

  1     2 ms     2 ms     2 ms  192.168.1.1
  2     5 ms     3 ms     3 ms  210.94.185.2
  3     5 ms     3 ms     3 ms  10.10.120.82
  4     5 ms     2 ms     4 ms  10.10.110.244
  5     4 ms     2 ms     2 ms  210.94.220.242
  6     4 ms     2 ms     2 ms  1.232.249.81
  7     5 ms     3 ms     6 ms  10.47.254.32
  8    15 ms     4 ms     4 ms  10.222.24.212
  9    14 ms     3 ms     3 ms  1.255.74.65
 10   211 ms   202 ms   205 ms  58.229.4.183
 11   136 ms   122 ms   123 ms  39.115.132.174
 12   227 ms   200 ms   205 ms  six.tr-cps.internet2.edu [206.81.80.77]
 13   283 ms   201 ms   204 ms  fourhundredge-0-0-0-20.4079.core1.seat.net.internet2.edu [163.253.1.162]
 14   219 ms   199 ms   201 ms  fourhundredge-0-0-0-0.4079.core1.salt.net.internet2.edu [163.253.1.156]
 15   190 ms   192 ms   226 ms  fourhundredge-0-0-0-0.4079.core1.denv.net.internet2.edu [163.253.1.170]
 16   226 ms   193 ms   213 ms  fourhundredge-0-0-0-0.4079.core1.kans.net.internet2.edu [163.253.1.243]
 17   190 ms   191 ms   190 ms  fourhundredge-0-0-0-3.4079.core2.chic.net.internet2.edu [163.253.1.244]
 18   205 ms   297 ms   203 ms  fourhundredge-0-0-0-3.4079.core2.eqch.net.internet2.edu [163.253.2.19]
 19   217 ms   202 ms   204 ms  fourhundredge-0-0-0-0.4079.core2.clev.net.internet2.edu [163.253.2.16]
 20   217 ms   305 ms   204 ms  fourhundredge-0-0-0-3.4079.core2.ashb.net.internet2.edu [163.253.1.138]
 21   215 ms   203 ms   202 ms  fourhundredge-0-0-0-1.4079.core1.phil.net.internet2.edu [163.253.1.137]
 22   214 ms   202 ms   203 ms  198.71.47.29
 23     *        *        *     요청 시간이 만료되었습니다.
 24     *        *        *     요청 시간이 만료되었습니다.
 25     *        *        *     요청 시간이 만료되었습니다.
 26     *        *        *     요청 시간이 만료되었습니다.
 27   267 ms   304 ms   202 ms  www-new.rutgers.edu [128.6.46.88]

추적을 완료했습니다.
```

- 로컬 Trace route의 결과이다.
- 27홉을 거쳐 목적지에 도착하였다.



- 공개 Trace route 서버에서 같은 목적지를 대상으로 trace route를 행한 모습이다.
- 로컬 Trace route와 달리 14홉을 거쳐 목적지에 도착하였다.

- 로컬 Trace route와 공개 Trace route 서버에서 행한 Trace route 결과가 다른 이유는 로컬 Trace route의 경우에는 패킷의 출발지가 한국이나, 공개 Trace route 서버의 경우 독일의 HanNet에서 진행하였기에 출발지가 독일이다. 따라서 출발지가 달라서 거쳐 가는 홉(경유지)도 달라지므로 결과가 달라질 수 밖에 없다.

v. 다음 예에서 장비의 종류와 포트, 또는 각 홉마다 가진 다른 속성을 확인할 수 있는가?
- Trace Route로는 목적지 호스트까지의 경로와 각 홉에서의 지연시간을 확인할 수 있지만, 장비의 종류와 포트, 각 홉마다 가진 다른 속성은 확인할 수 없다.

vi. 동일한 예를 사용하여 마지막 라인들이 왜 공백이며 일반적으로 Traceroute가 목적지에 도달했는지 여부를 어떻게 알려주는가?

- 마지막 라인들이 공백인 이유는 목적지로의 마지막 패킷이 도착하고 ICMP Echo Reply가 반환되었기 때문이다.
- 목적지에 도달했는지에 대한 여부는 마지막 라인에서 목적지 IP 주소, 호스트 이름, 패킷전송에 든 시간 등의 표시 여부로 Trace Route가 완료되었는지 알 수 있다.

3) 느낀 점

- 이번 실습에서는 와이어샤크를 이용하여 네트워크에 전송되는 패킷을 관찰하고, 정보를 읽는 작업을 수행했다. 네트워크에는 수많은 패킷이 전송되고 있으며, 패킷마다 계층 구조로 되어 있는 것을 와이어샤크로 확인할 수 있었다. 이렇게 와이어샤크로 패킷을 손쉽게 볼 수 있는 것을 보면 중간에 누군가가 패킷을 가로채기는 쉽다는 생각이 들었다. 물론 네트워크에 수많은 패킷이 존재하고 있고, 분할되어 여러 패킷으로 전송되는 일도 있기에 원하는 정보를 찾기는 조금 어려울 것으로 생각은 하는데 그래도 개인 정보가 담긴 패킷을 가로챌 수 있다는 것이 참으로 무서웠다. 이번 실습으로 처음 알게 된 정보로는 운영 체제에 따라서 패킷의 정보가 다르다는 것이었다. 운영 체제가 다르더라도 같은 프로토콜을 이용해서 정보를 전송하면 패킷의 TTL이나 window size, 단편화 여부가 같을 줄 알았는데 그게 아니었다는 것을 알고는 새삼 놀랐다. 운영 체제마다 패킷의 TTL이나 windows size가 다르다면 만약 다른 운영 체제끼리 네트워크 통신을 진행한다면 문제가 생기지 않을지 궁금해진다. 물론 지금까지 그런 문제가 발생했다는 이야기를 들은 적이 없는 것을 보면 분명 호환되도록 처리하지 않았을까 싶다. 그리고 와이어샤크로 패킷을 하나 보면 내용이 바이너리 코드로 이루어져 있는데 이걸 역 컴파일하면 내용을 볼 수 있지 않을까 생각한다. 그리고 값을 변경하기도 용이하다고 생각한다. 헥사 코드 또는 바이너리 코드를 보고 이해할 수만 있다면 조작하기 쉽다고 생각한다. 나중에 이런 기술을 배울지 모르겠지만 개인적으로 흥미가 생긴다. 개인적으로 이런 기술에 대해 알아봐야겠다.