네트워크 패킷 포렌식 4장 FTP 분석

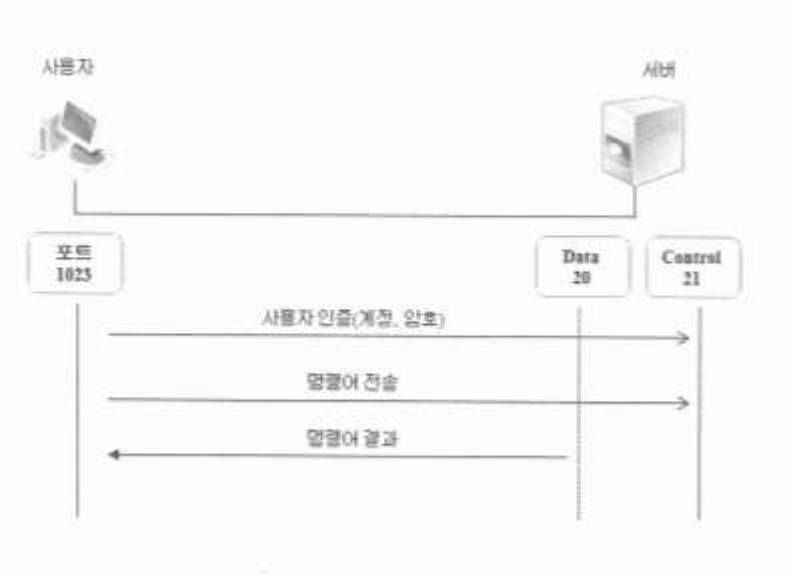
FTP(File Transfer Protocol, FTP):

파일전송 프로토콜로, TCP/IP 프로토콜을 가지고 서버와 클라이언트 사이의 파일 전송을 하기 위한 프로토콜

FTP는 TCP/IP계층 중 응용계층에 속함

FTP 통신방식

- 2개의 포트번호를 사용 : TCP 21, 20번
- 21번 : 사용자 인증 및 명령어 전달에 사용
- 20번 : 사용자 명령을 처리하여 결과를 전달

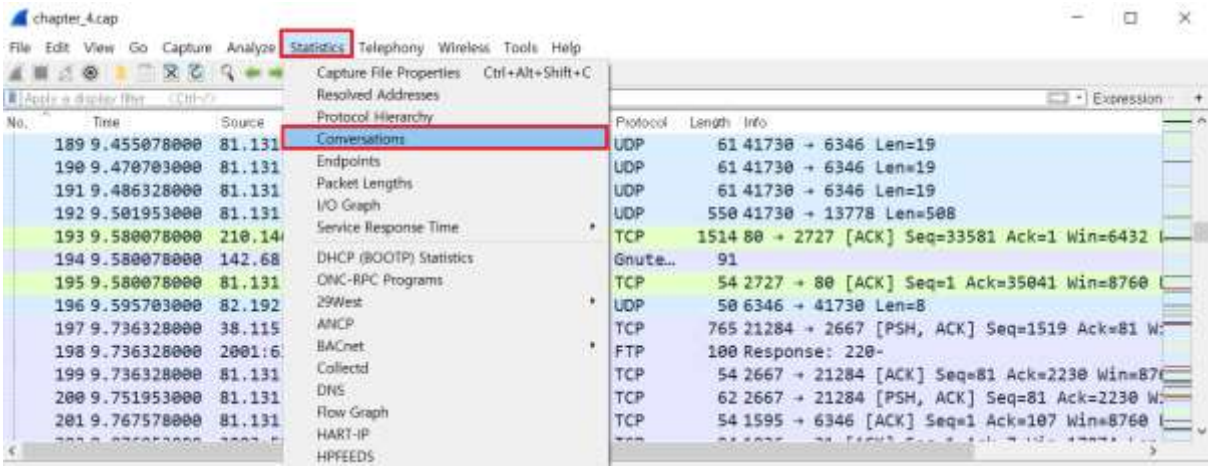


FTP 패킷 분석

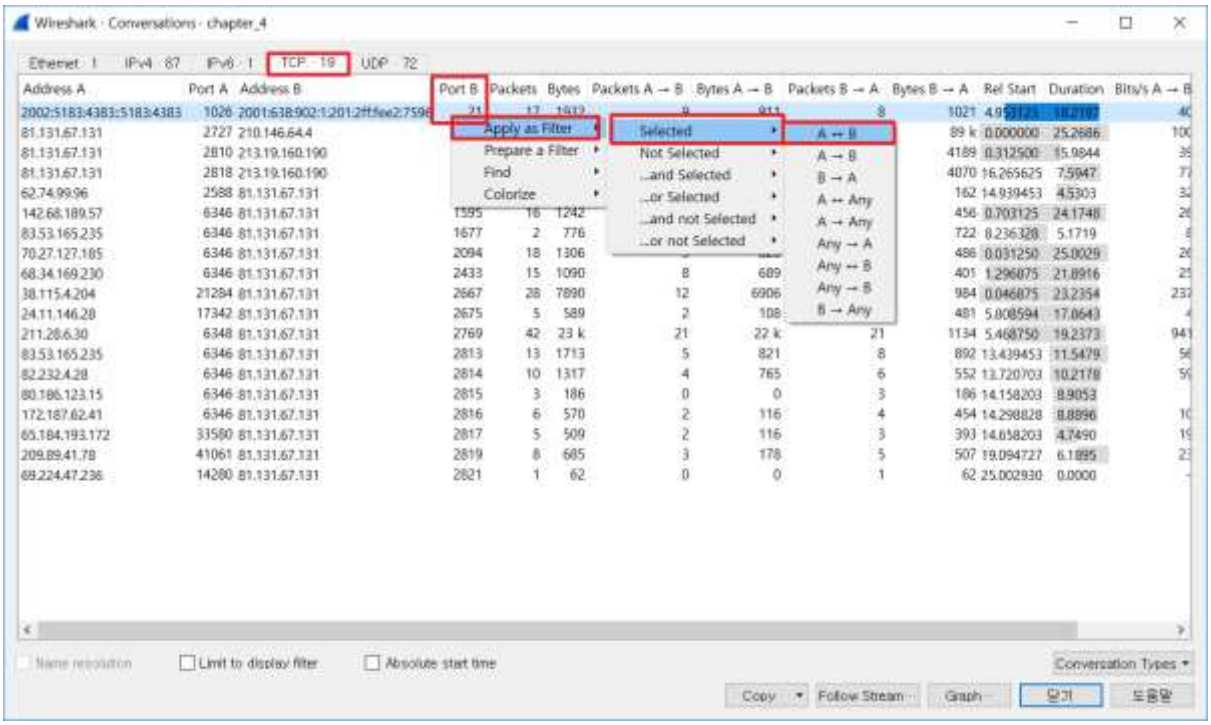
⇨ 패킷 분석에서 가장 먼저 해야 할 것?

: 어떤 IP에서 어떤 대상으로 접속했는지에 대한 전체 정보 확인

1. Statistics(통계)를 통해 캡쳐된 패킷들의 정보를 확인 : Statistics -> Conversation



2. 분석할 패킷 대상을 필터링하여 메인 화면에서 확인 : Apply as Filter -> Selected



➔ FTP는 TCP를 사용하며 포트번호가 21인 프로토콜이다. PortA, PortB 컬럼을 클릭하여 정렬후 포트번호가 21번인 패킷 통계를 찾는다.

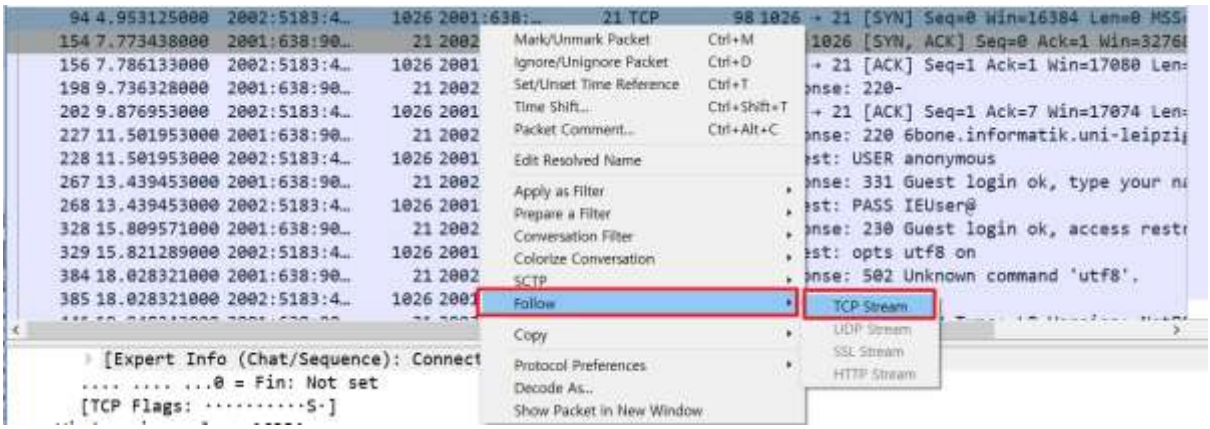
➔ Apply as Filter->Selected -> (A <-> B) 를 클릭, 양방향 필터링을 통해 사용자와 서버 간의 통신 패킷들을 필터링한다.

1) Apply as Filter를 통한 ftp프로토콜의 통신 양방향 필터가 입력됨을 확인

2) TCP 세션 확인(3-handshake)

3) FTP와 사용자간의 통신과정들이 포함된 패킷

3. 패킷의 개별 세션별로 정보 수집 : Follow Stream -> TCP Stream



사용자 – FTP 응용프로그램 간의 통신내용 확인 가능.

사용자의 커맨드 입력 : 빨간색, FTP가 보낸 내용 : 파란색

네트워크 포렌식 5장 Telnet 패킷 분석

Telnet(Telecommunication Network Protocol) :

자신이 사용권한을 가지고 있다는 전제하에 다른 사람의 호스트 컴퓨터를 원격지에서 접근할 수 있는 기능 제공

Telnet 통신 방식

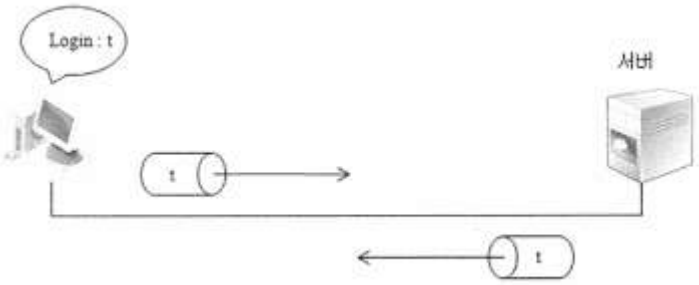
```
① Kernel 2.4.31 on an i686
②③ login: test
④⑤ Password:
⑥ Last login: Tue Nov 24 19:31:05 from 192.168.0.128
[test@LWEB01 test]$
```

① 접속 시 서버 배너정보 리턴 – kernel 2.4.31 on an i686

② 접속 계정을 요구하는 login: 프롬프트가 떨어진다

③ 사용자가 계정을 입력한다. – test

계정이 test 일 경우, 계정의 첫 번째 문자열 t를 입력하면 t 문자열이 서버로 전송되며 다시 서버에서 동일한 문자 t를 리턴하면서 화면에 입력한 t 문자열이 보여지는 echo 현상이 발생.



④ 패스워드 요구하는 Password: 프롬프트가 떨어진다.

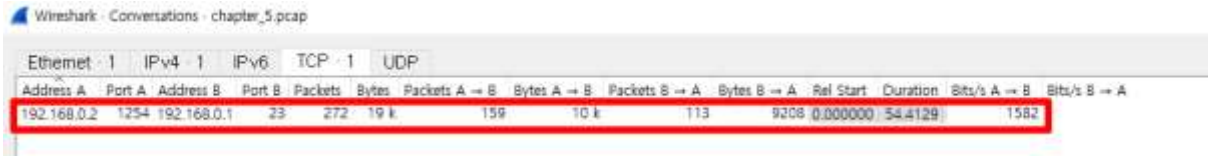
패스워드는 login과 같이 echo 현상이 일어나지 않는다.

⑤ 사용자가 패스워드를 입력한다.(해당 정보는 보이지 않음)

⑥ 접속시간과 함께 프롬프트가 떨어진다. – Last login: 날짜, 시간

Telnet 패킷분석

1. Statistics -> Conversations : TCP 탭에 Telnet만 보이며 필터링 적용(Apply as filter)



2. 필터링 후 메인 화면을 보면 TCP세션을 연결하는 3-handshaking을 확인 할 수 있다.

Source	Destination	Protocol	Length	Info
192.168.0.2	192.168.0.1	TCP	74	1254 → 23 [SYN] Seq=0 Win=32120 Len=
192.168.0.1	192.168.0.2	TCP	74	23 → 1254 [SYN, ACK] Seq=0 Ack=1 Wir
192.168.0.2	192.168.0.1	TCP	66	1254 → 23 [ACK] Seq=1 Ack=1 Win=3212

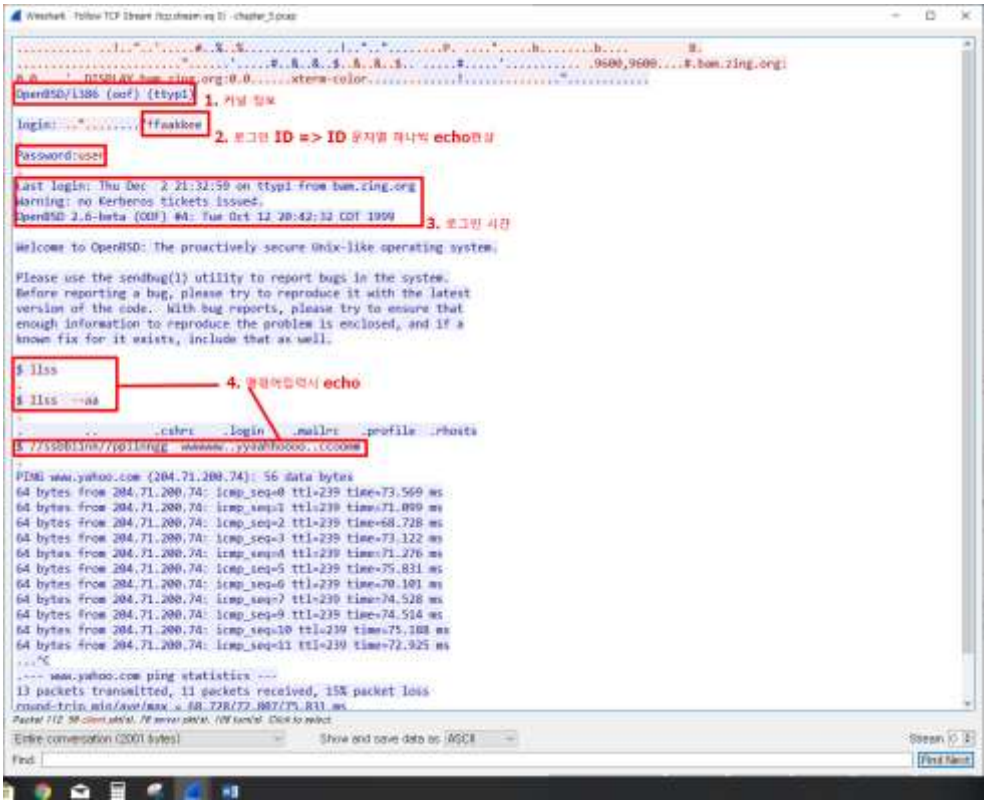
3. 그 이후에는 Telnet의 로그인 명령어 등을 사용한 내역이 있다.

4. 그리고 마지막 패킷4개를 관찰하면, 송신자 192.168.0.1에서 192.168.0.2에게 [Fin,Ack]를 보내 통신종료를 요청하며, 수신자는 [ACK]를 응답하고,

   192.168.0.2에서 192.168.0.1에게 [FIN,ACK]를 전송하고, 응답으로 [ACK]를 전송한다.

192.168.0.1	192.168.0.2	TCP	66	23 → 1254 [FIN, ACK] Seq=1743 Ack=260 W
192.168.0.2	192.168.0.1	TCP	66	1254 → 23 [ACK] Seq=260 Ack=1744 Win=32
192.168.0.2	192.168.0.1	TCP	66	1254 → 23 [FIN, ACK] Seq=260 Ack=1744 W
192.168.0.1	192.168.0.2	TCP	66	23 → 1254 [ACK] Seq=1744 Ack=261 Win=17

5. Telnet 데이터를 TCP Stream으로 한번에 확인하여 보자.



네트워크 포렌식 6장 파일 Magic Number

1. 패킷 내 파일 분석
    - [Conversations]를 통한 전체 내용 파악
    - 세션별 [Follow Stream]을 통해 통신 내용 분석
    - 세션별[Follow Stream]으로 확인되지 않는 정보는 필터링을 통해 상세 분석
2. 포트 번호 범위(알아야 할 것들)
    - 0 ~ 1023 : 잘 알려진 포트
    - 1024 ~ 49151 : 등록된 포트
    - 49152 ~ 65535 : 동적 포트
    -
3. 주요 포트 번호

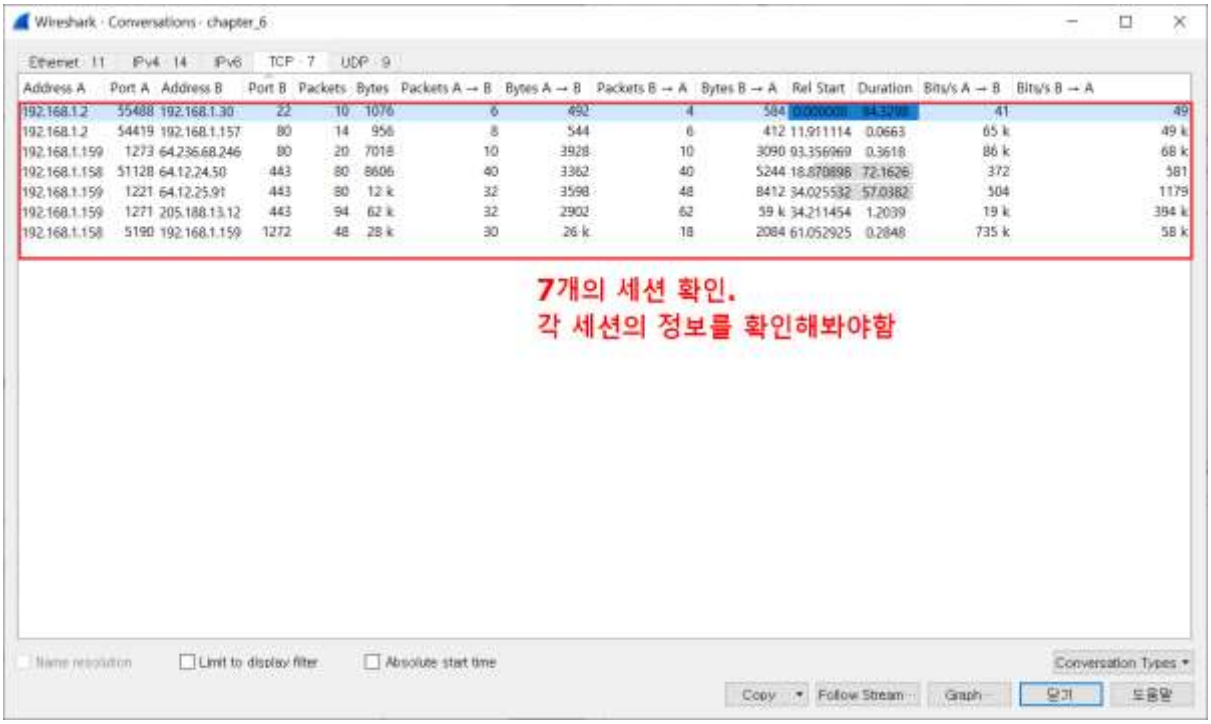
20	TCP		FTP (파일 전송 프로토콜) - 데이터 포트
21	TCP		FTP - 제어 포트
22	TCP		SSH (Secure Shell) - ssh scp, sftp 같은 프로토콜 및 포트 포워딩
23	TCP		텔넷 프로토콜 - 암호화되지 않은 텍스트 통신
24	TCP		개인메일 시스템
25	TCP		SMTP (Simple Mail Transfer Protocol) - 이메일 전송에 사용
53	TCP	UDP	DNS(Domain Name System)
80	TCP	UDP	HTTP(HyperText Transfer Protocol) – 웹페이지 전송
443	TCP		HTTPS – SSL 위의 HTTP (암호화 전송)

4. 패킷 분석(1번 내용의 순서에 따라서..)

- [Conversations]를 통한 전체내용 파악 => 세션 확인

주의할 것 : Follow Stream시에 Apply as Filter 적용후 메인화면에서 Follow Stream 확인 할 것!

Conversations에서 Follow Stream을 하면 최초의 확인한 Stream만 확인되는 것 같음.

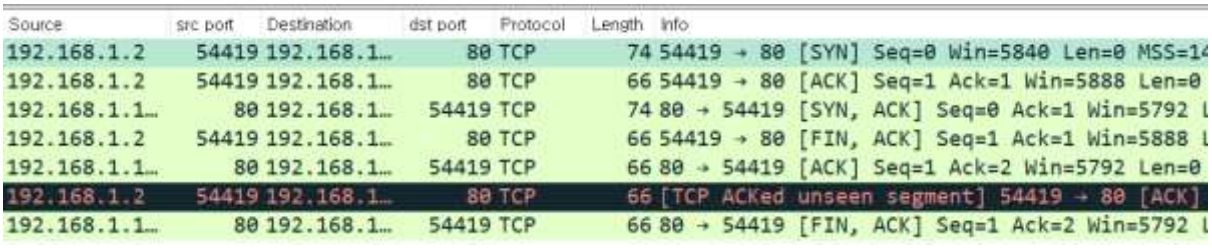


1) 첫번째 세션 : Port B = 22 -> Telnet 등의 평문데이터를 암호화 처리하는 프로토콜.

➔ Follow Stream을 통해 내용을 확인해보면, 암호화 데이터이기 때문에 알아볼 데이터는 알아볼 수 없음.

2) 두번째 세션 : Port B = 80 -> HTTP 프로토콜을 사용한다.

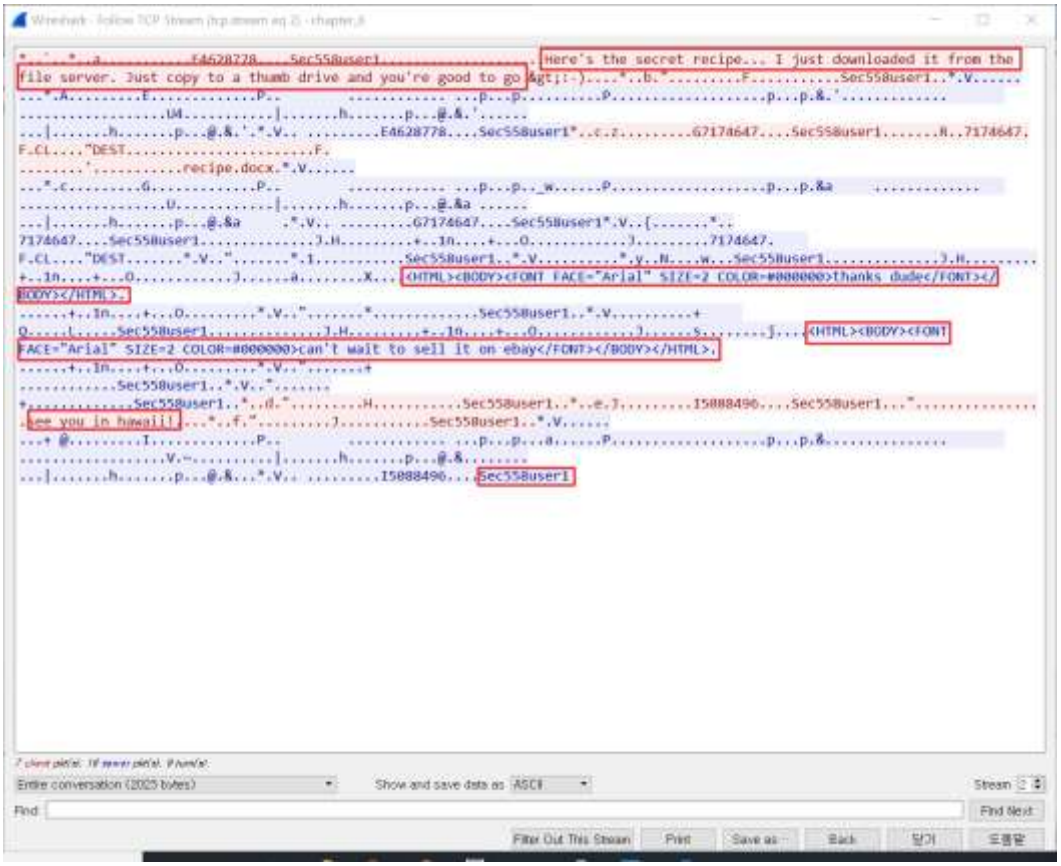
➔ Follow Stream을 클릭하여 확인하면 아무런 데이터가 뜨지 않는다. Apply as filter 적용하여 패킷을 확인 한 결과는 아래와 같다.



즉, 세션이 [SYN],[ACK],[SYN,ACK]로 세션이 연결되고, [FIN,ACK],[ACK],[FIN,ACK]를 통해 곧바로 세션이 종료되었다. 이 세션은 단순히 송신자 192.168.1.2가 포트를 스캔하여 포트 80번 오픈정도만 확인한 것으로 가정할 수 있다.

3) 세번째 세션 : Port B = 443 이므로 SSL 프로토콜(HTTPS)를 사용함을 알 수 있다.

➔ Follow Stream을 통해 확인하여 보자.



암호화 통신임에도 불구하고 일부 확인할 수 있는 문자열들이 보인다.

4) 네 번째 세션 : Port B = 443이므로 HTTPS로 판단할 수 있다.

➔ Follow Stream으로 확인하면 3번의 내용처럼 평문데이터를 확인할 수 없다.

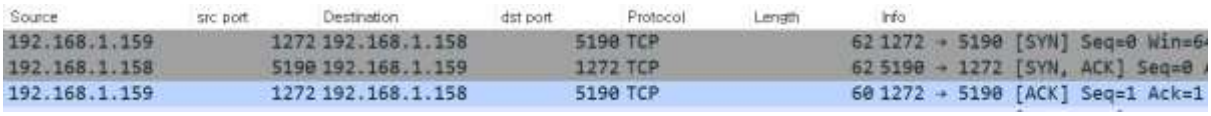
5) 다섯 번째 세션 : Port B = 443이므로 HTTPS로 판단 할 수 있다.

➔ Follow Stream으로 확인하여도 평문데이터를 확인 할 수 없다.

6) 여섯 번째 세션 : Port B =443이므로 HTTPS로 판단 할 수 있다

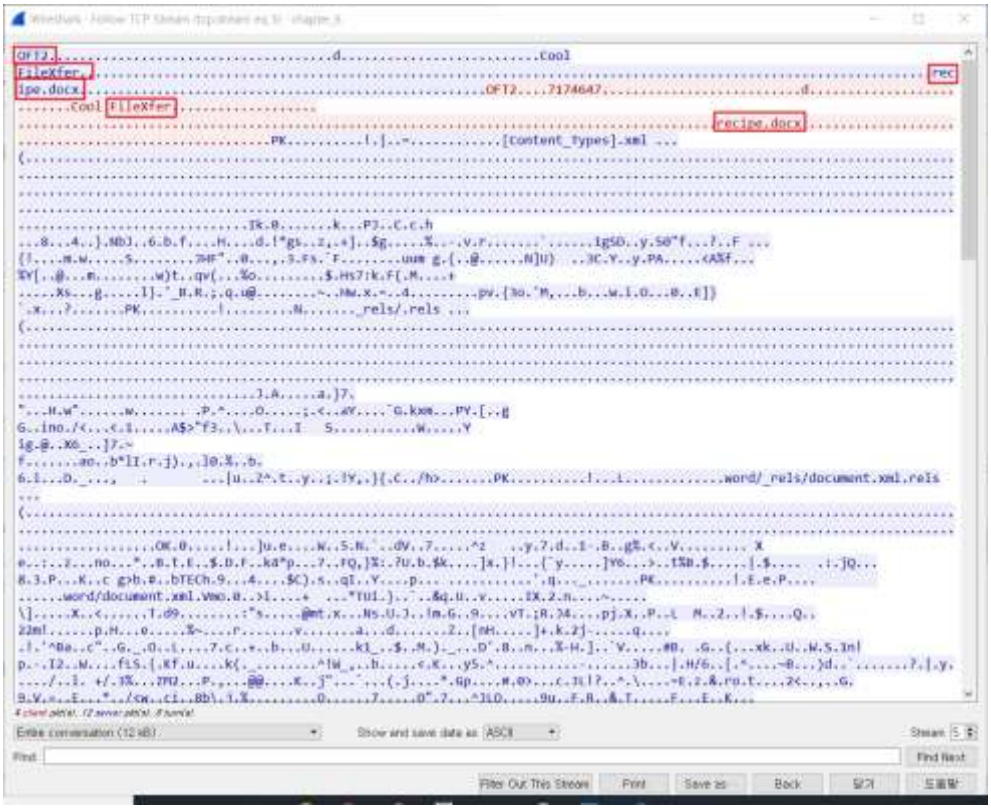
➔ Follow Stream으로 확인 시 3)의 내용과 같은 평문데이터를 확인할 수 없다.

7) 일곱 번째 세션 : Port B = 1272 이므로 특정 프로토콜을 사용한다고 단정지을 수 없다. Apply as Filter 적용 후 패킷을 살펴보자.



통신방향이 192.168.1.159 -> 192.168.1.158 임을 확인할 수 있다.

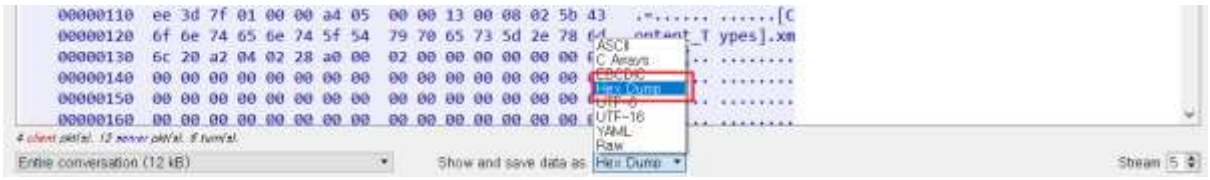
➔ Follow Stream을 통해 데이터의 내용을 확인해 보면



OFT2와 filexFer, recipe.docx등의 내용을 확인 할 수 있는데,

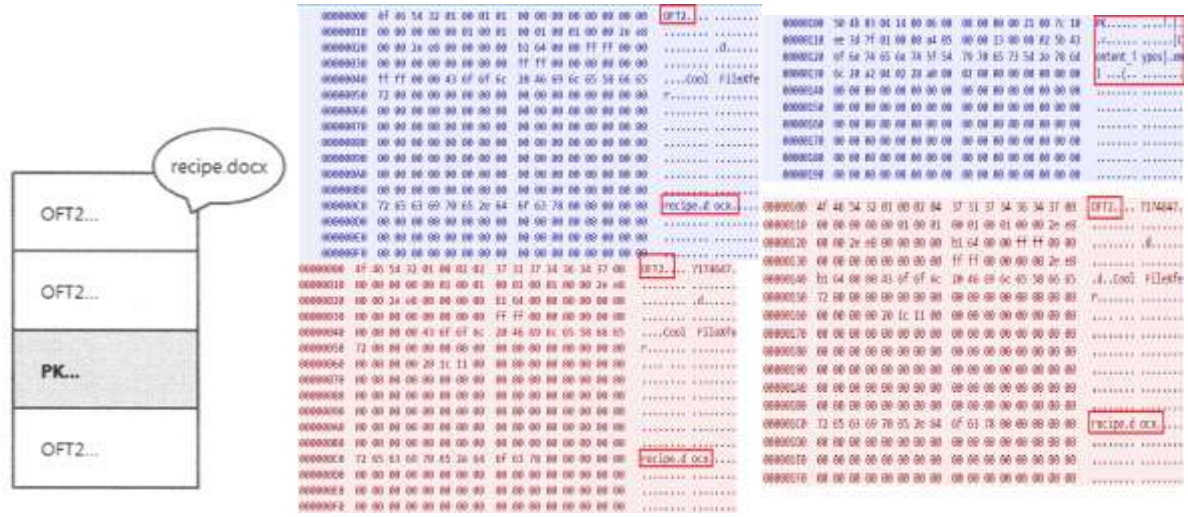
OFT2 packet을 검색해 보면, AOL Instant Massenger 라는 정보를 확인 할 수 있고, 이를 통해 메신저 프로그램을 통해 recipe.docx 파일을 전송했을 수 있겠다는 생각을 해 볼 수 있다.

또한 Follow Stream 화면의 아래에 Ascii를 HexDump로 바꾸어 보자.



바꾸어보면 [파-빨-파-빨]로 4개의 영역으로 구분되어 있음을 알 수 있다.

구조를 간단하게 요약하여 나타내어 보면 아래 그림과 같다.



각 OFT2 메신저 통신 내용 안에는 recipe.docx 라는 문자열이 확인됨을 알 수 있다.

"PK"라는 문자열은 Magic Number 이다.(시그니처 값 이라고도 한다.)

Magic Number란, 파일 포맷마다 고유의 넘버값을 가지고 있어, 그 값으로 파일종류를 파악할 수 있고, 패킷 내부에서는 이를 이용하여 파일을 추출할 수 있다.

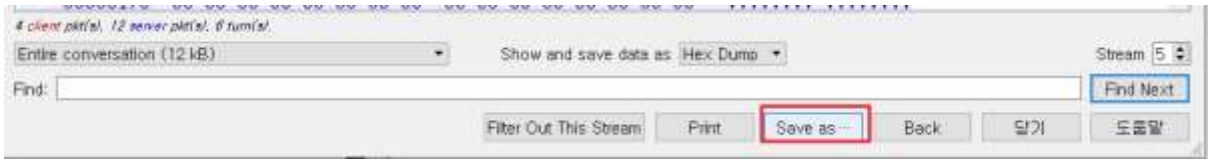
"PK"는 ZIP, OFFICE등이 사용하는 파일 포맷임을 알 수 있다.

```
50 4b 03 04 14 00 06 00  08 00 00 00 21 00 7c 10
```

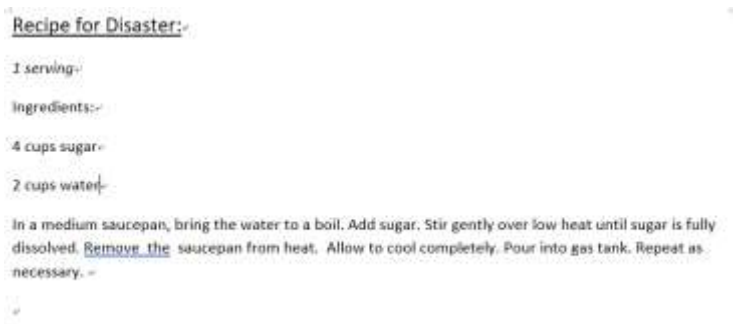
위와 같은 구조를 보고, OFFICE 파일임을 알 수 있으며, OFT2에서 확인된 문자열이

recipe.docx임을 보고 docx파일이라고 추측할 수 있다. 그리고 Hexdump를 Raw로 변경 후,

해당 값을 [Save as]클릭, recipe.docx로 저장하고 OFT2 내용을 [HxD]프로그램을 통해 삭제해 주자



편집 후 recipe.docx의 내용을 확인해 보면 다음과 같다.



Recipe for Disaster:

1 serving

Ingredients:

4 cups sugar

2 cups water

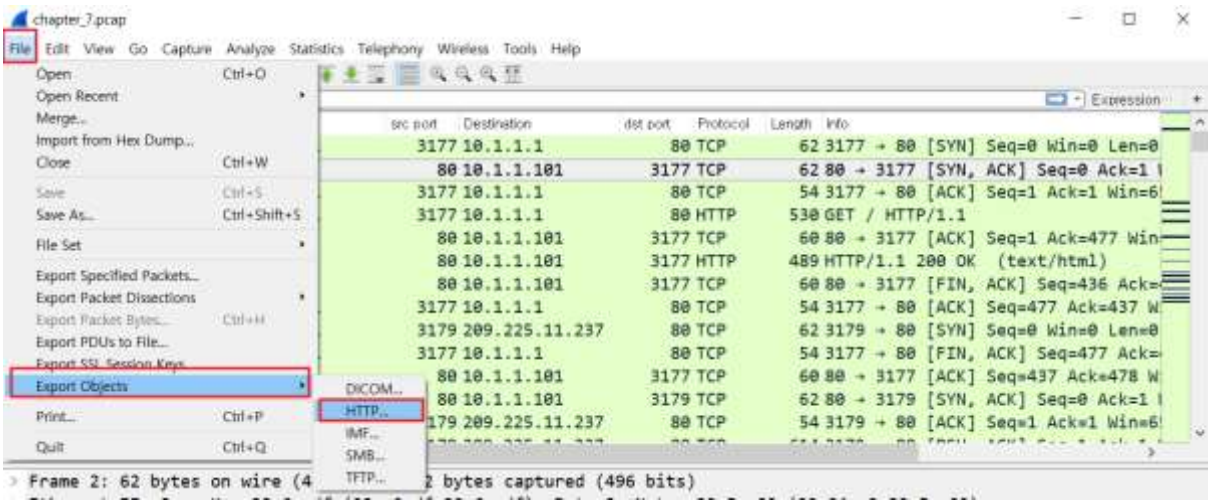
In a medium saucepan, bring the water to a boil. Add sugar. Stir gently over low heat until sugar is fully dissolved. Remove the saucepan from heat. Allow to cool completely. Pour into gas tank. Repeat as necessary.

네트워크 포렌식 7장 HTTP 컨텐츠 분석

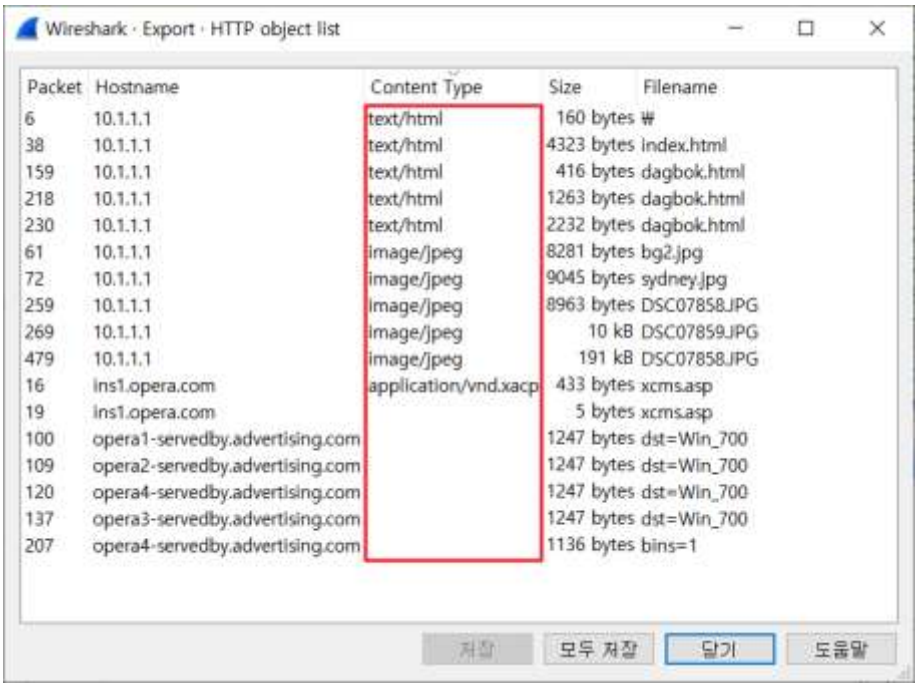
HTTP 패킷 내에 데이터를 추출하는방법에는 두가지가 있음.

1. 6장의 magic number로 찾아내어 추출하는 방법

   1) File -> Export Objects -> HTTP



   2) Content type을 확인하여 추출가능



2. NetworkMiner을 통해 추출