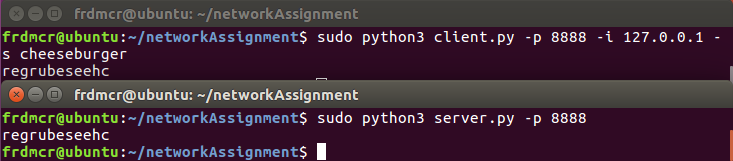
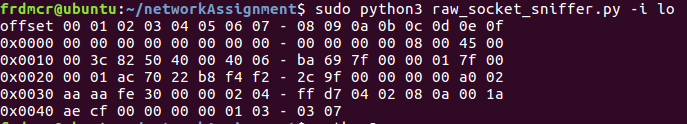
# raw\_sniffer.py를 사용한 패킷 분석

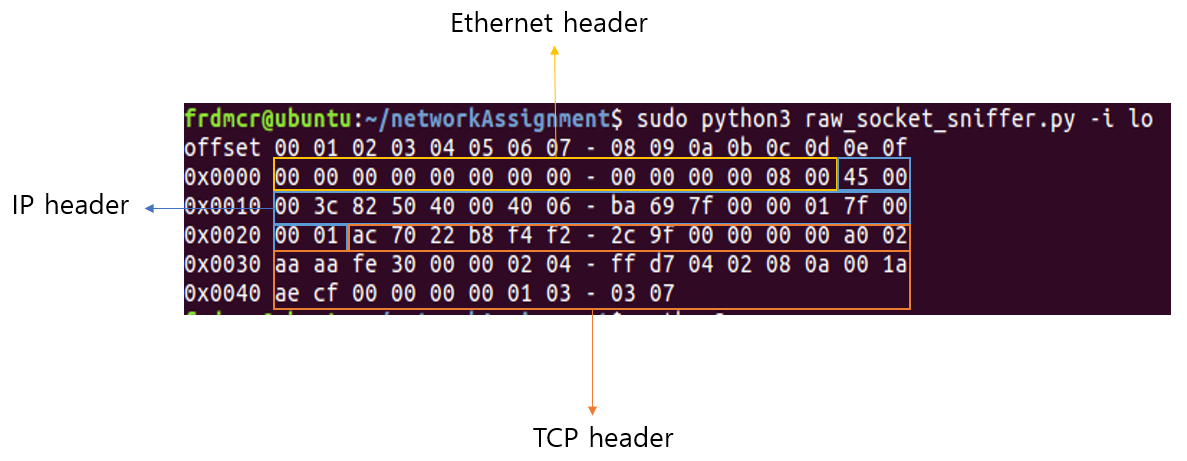
* **문자열 전송**



* **패킷 캡처**



* **패킷 영역 나누기**



* Ethernet header

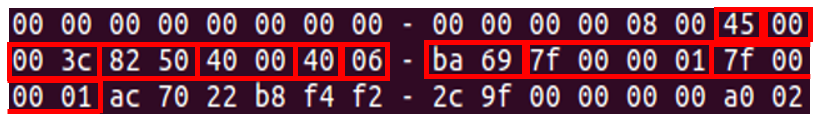


1.Destination(MAC address) : 00:00:00:00:00:00 (loop back)

2.Source(MAC address) : 00:00:00:00:00:00 (loop back)

3.Type : IPv4 0x0800 (해당 영역의 값이 0x0600 이상이면 Ethernet 2.0의 Type으로 해석하고, 0x0600 미만이면 IEEE 802.3의 Length로 해석한다. 따라서 패킷은 Ethernet 2.0 프레임의 IP 프로토콜)

* IP header



1.Version : 4 (45의 첫 번째 16진수 ‘4’로 IPv4를 나타냄)

2.Header Length : 20 bytes (45의 두 번째 16진수 ‘5’로 단위가 4바이트라 4x5 = 20 바이트)

3.DSCP : CS0 (패킷이 받을 서비스의 종류 또는 등급을 나타내는데, 기본 값이 0이다)

4.Total Length : 60 (IP header와 IP payload를 포함하는 IP 패킷의 전체 크기, 최대 65535 bytes)

5.Identification : 0x8250 (패킷의 ID 번호)

6.Flags : 0x4000, Don’t fragment (단편화(fragment)가 진행되어 있지 않다는 의미이다)

7.Time To Live : 64 (패킷이 살아 있는 시간으로 라우터를 기준으로 한 TTL 값이라고 한다)

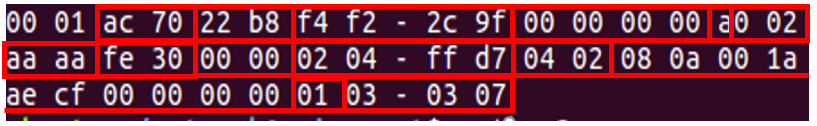
8.Protocol : TCP (6)

9.Header checksum : 0xba69 (IP header의 에러를 검사하는 부분)

10.Source : 127.0.0.1 (loop back)

11.Destination : 127.0.0.1 (loop back)

* TCP header



1.Source Port : 44144 (0xac70)

2.Destination Port : 8888 (0x22b8)

3.Sequence number : 0(3-way-handshaking에서 사용하는 번호로 보안적 측면을 위해 랜덤 생성)

4.Acknowledgement number : 0(3-way-handshaking에서 상대방으로부터 받기를 기대하는 일련번호)

5.Header Length : 40 bytes (a)

6.Flags : 0x002 (SYN으로 송수신자간 일련번호로써 세션을 동기화하기를 요청)

7.Window size value : 43690 (수신자로부터 응답을 기다리지 않고 연속적으로 보낼 수 있는 byte)

8.Checksum : 0xfe30 (에러 확인)

9.Urgent pointer : 0

10.Maximum segment size : 65495 bytes (TCP Option : 0x0204ffd7)

11.SACK permitted (TCP Option : 0x0402 = 선택 확인응답 옵션 허용)

12.Timestamps : TSval 1748687, TSecr 0 (TCP Option : 0x080a001aaecf00000000)

13.No-Operation (TCP Option : NOP = 필요시 송신 측에서 필드를 4 byte의 배수가 되도록 채움)

14.Window scale : 7 (TCP Option : 윈도우 크기 확장 값 7)