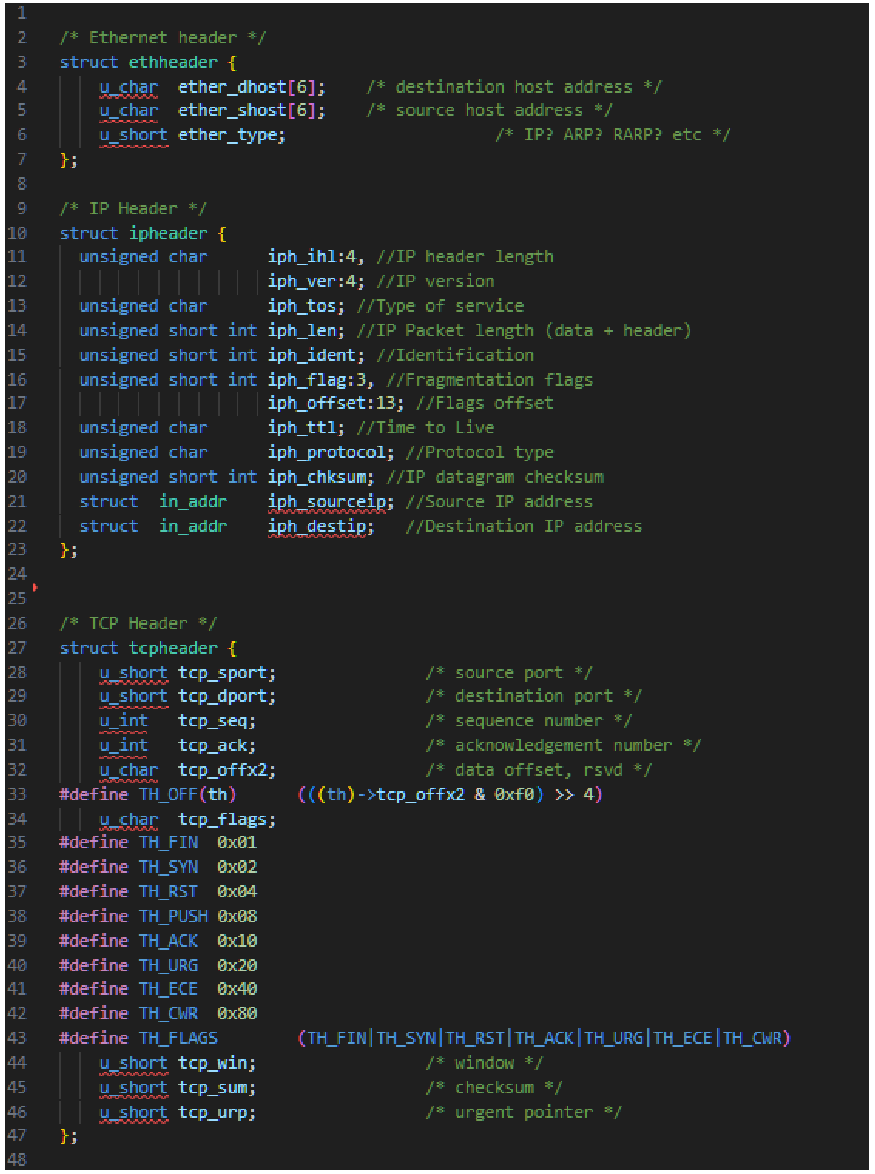
PCAP Programming 보고서

[4반]김정훈_6342

코드 설명

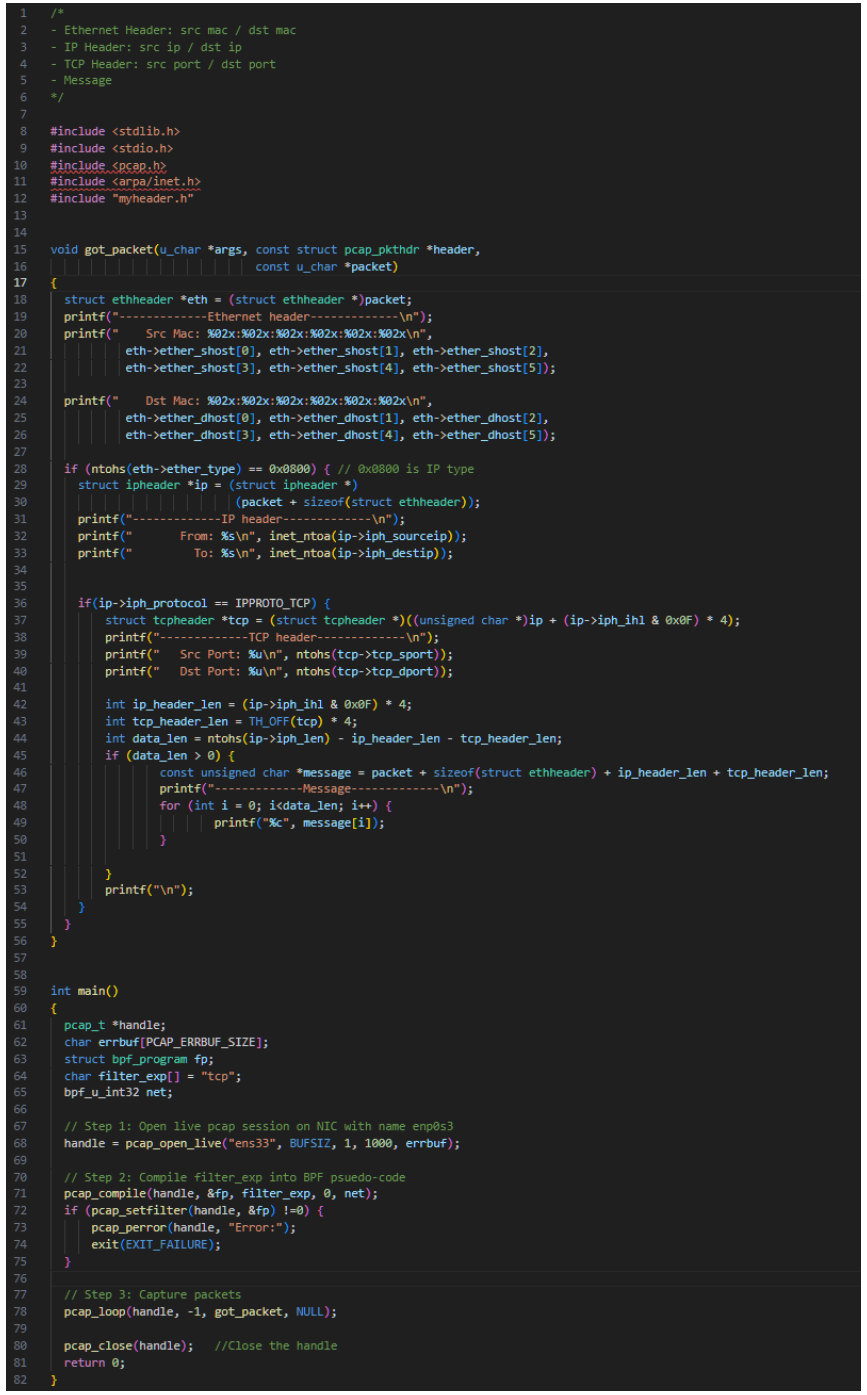


```c
/* Ethernet header */
struct ethheader {
    u_char  ether_dhost[6];    /* destination host address */
    u_char  ether_shost[6];    /* source host address */
    u_short ether_type;                     /* IP? ARP? RARP? etc */
};

/* IP Header */
struct ipheader {
  unsigned char      iph_ihl:4, //IP header length
                     iph_ver:4; //IP version
  unsigned char      iph_tos; //Type of service
  unsigned short int iph_len; //IP Packet length (data + header)
  unsigned short int iph_ident; //Identification
  unsigned short int iph_flag:3, //Fragmentation flags
                     iph_offset:13; //Flags offset
  unsigned char      iph_ttl; //Time to Live
  unsigned char      iph_protocol; //Protocol type
  unsigned short int iph_chksum; //IP datagram checksum
  struct  in_addr    iph_sourceip; //Source IP address
  struct  in_addr    iph_destip;   //Destination IP address
};


/* TCP Header */
struct tcpheader {
    u_short tcp_sport;               /* source port */
    u_short tcp_dport;               /* destination port */
    u_int   tcp_seq;                 /* sequence number */
    u_int   tcp_ack;                 /* acknowledgement number */
    u_char  tcp_offx2;               /* data offset, rsvd */
#define TH_OFF(th)      (((th)->tcp_offx2 & 0xf0) >> 4)
    u_char  tcp_flags;
#define TH_FIN  0x01
#define TH_SYN  0x02
#define TH_RST  0x04
#define TH_PUSH 0x08
#define TH_ACK  0x10
#define TH_URG  0x20
#define TH_ECE  0x40
#define TH_CWR  0x80
#define TH_FLAGS        (TH_FIN|TH_SYN|TH_RST|TH_ACK|TH_URG|TH_ECE|TH_CWR)
    u_short tcp_win;                 /* window */
    u_short tcp_sum;                 /* checksum */
    u_short tcp_urp;                 /* urgent pointer */
};
```

myheader.h

Ethernet header,
IP header,
TCP header
헤더 구조가 있는 헤더파일.

코드 설명



```c
/*
- Ethernet Header: src mac / dst mac
- IP Header: src ip / dst ip
- TCP Header: src port / dst port
- Message
*/

#include <stdlib.h>
#include <stdio.h>
#include <pcap.h>
#include <arpa/inet.h>
#include "myheader.h"


void got_packet(u_char *args, const struct pcap_pkthdr *header,
                              const u_char *packet)
{
  struct ethheader *eth = (struct ethheader *)packet;
  printf("------------Ethernet header------------\n");
  printf("    Src Mac: %02x:%02x:%02x:%02x:%02x:%02x\n",
            eth->ether_shost[0], eth->ether_shost[1], eth->ether_shost[2],
            eth->ether_shost[3], eth->ether_shost[4], eth->ether_shost[5]);

  printf("    Dst Mac: %02x:%02x:%02x:%02x:%02x:%02x\n",
            eth->ether_dhost[0], eth->ether_dhost[1], eth->ether_dhost[2],
            eth->ether_dhost[3], eth->ether_dhost[4], eth->ether_dhost[5]);

  if (ntohs(eth->ether_type) == 0x0800) { // 0x0800 is IP type
    struct ipheader *ip = (struct ipheader *)
                            (packet + sizeof(struct ethheader));
    printf("------------IP header------------\n");
    printf("       From: %s\n", inet_ntoa(ip->iph_sourceip));
    printf("         To: %s\n", inet_ntoa(ip->iph_destip));


    if(ip->iph_protocol == IPPROTO_TCP) {
        struct tcpheader *tcp = (struct tcpheader *)((unsigned char *)ip + (ip->iph_ihl & 0x0F) * 4);
        printf("------------TCP header------------\n");
        printf("   Src Port: %u\n", ntohs(tcp->tcp_sport));
        printf("   Dst Port: %u\n", ntohs(tcp->tcp_dport));

        int ip_header_len = (ip->iph_ihl & 0x0F) * 4;
        int tcp_header_len = TH_OFF(tcp) * 4;
        int data_len = ntohs(ip->iph_len) - ip_header_len - tcp_header_len;
        if (data_len > 0) {
                const unsigned char *message = packet + sizeof(struct ethheader) + ip_header_len + tcp_header_len;
                printf("------------Message------------\n");
                for (int i = 0; i<data_len; i++) {
                        printf("%c", message[i]);
                }

        }
        printf("\n");
    }
  }
}


int main()
{
  pcap_t *handle;
  char errbuf[PCAP_ERRBUF_SIZE];
  struct bpf_program fp;
  char filter_exp[] = "tcp";
  bpf_u_int32 net;

  // Step 1: Open live pcap session on NIC with name enp0s3
  handle = pcap_open_live("ens33", BUFSIZ, 1, 1000, errbuf);

  // Step 2: Compile filter_exp into BPF psuedo-code
  pcap_compile(handle, &fp, filter_exp, 0, net);
  if (pcap_setfilter(handle, &fp) !=0) {
      pcap_perror(handle, "Error:");
      exit(EXIT_FAILURE);
  }

  // Step 3: Capture packets
  pcap_loop(handle, -1, got_packet, NULL);

  pcap_close(handle);   //Close the handle
  return 0;
}
```

PCAP_programming.c

Ethernet header 구조체 생성 및
src Mac / Dst Mac 출력

packet + ethheader의 크기 = ip header의 시작위치
Src ip / dst ip 출력

packet + ethheader크기 + ipheader크기 = tcpheader의 시작위치
(ip -> iph_ihl에 ipheader의 길이정보)
Src port / dst port 출력

packet + ethheader크기 + ipheader크기 + tcpheader크기 = message위치
Message의 길이가 1이상일 경우 출력

Main함수의 Filter에 tcp정보만 받도록 설정

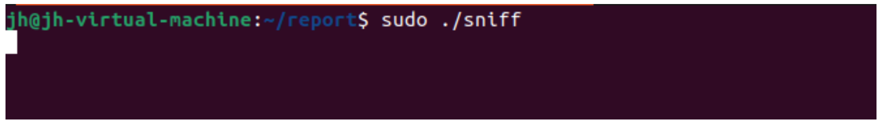
※

1. ip_header_len
'Ip->iph_ihl'은 ip헤더 길이를 32비트 단위로 나타냄. -> *5를 해줌

2. tcp_header_len
'TH_OFF(tcp)'는 ipv4와 마찬가지로 32비트 단위로 나타냄. -> *5

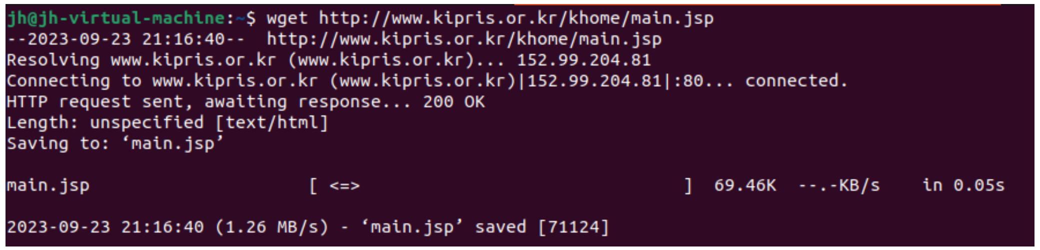
3. data_len
'ip->iph_len'는 header + data의 길이로 네트워크 바이트 순서(big endian)를 호스트 바이트순서(little endian)으로 바꾸기 위해 ntohs를 사용함. 총 길이에서 ip헤더의 길이와 tcp헤더의 길이를 빼면 데이터의 길이가 나온다.

실행 화면



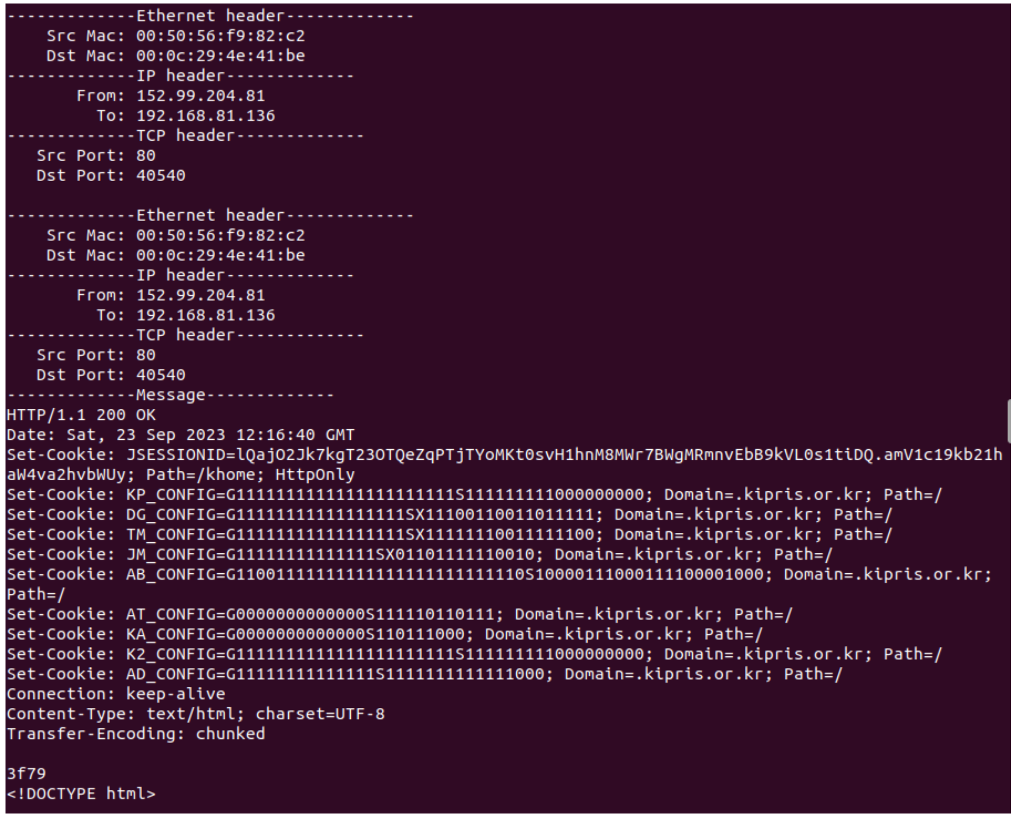
```
jh@jh-virtual-machine:~/report$ sudo ./sniff
```

프로그램을 실행하고 대기합니다.



```
jh@jh-virtual-machine:~$ wget http://www.kipris.or.kr/khome/main.jsp
--2023-09-23 21:16:40--  http://www.kipris.or.kr/khome/main.jsp
Resolving www.kipris.or.kr (www.kipris.or.kr)... 152.99.204.81
Connecting to www.kipris.or.kr (www.kipris.or.kr)|152.99.204.81|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: ‘main.jsp’

main.jsp                     [ <=>                                ]  69.46K  --.-KB/s    in 0.05s

2023-09-23 21:16:40 (1.26 MB/s) - ‘main.jsp’ saved [71124]
```

다른 터미널로 wget 명령어를 사용해 웹페이지를 호출합니다.
Message의 내용을 보기위해 http페이지를 호출합니다.



```
------------Ethernet header------------
    Src Mac: 00:50:56:f9:82:c2
    Dst Mac: 00:0c:29:4e:41:be
------------IP header------------
       From: 152.99.204.81
         To: 192.168.81.136
------------TCP header------------
   Src Port: 80
   Dst Port: 40540

------------Ethernet header------------
    Src Mac: 00:50:56:f9:82:c2
    Dst Mac: 00:0c:29:4e:41:be
------------IP header------------
       From: 152.99.204.81
         To: 192.168.81.136
------------TCP header------------
   Src Port: 80
   Dst Port: 40540
------------Message------------
HTTP/1.1 200 OK
Date: Sat, 23 Sep 2023 12:16:40 GMT
Set-Cookie: JSESSIONID=lQajO2Jk7kgT23OTQeZqPTjTYoMKt0svH1hnM8MWr7BWgMRmnvEbB9kVL0s1tiDQ.amV1c19kb21h
aW4va2hvbWUy; Path=/khome; HttpOnly
Set-Cookie: KP_CONFIG=G1111111111111111111111S111111111000000000; Domain=.kipris.or.kr; Path=/
Set-Cookie: DG_CONFIG=G11111111111111111SX11100110011011111; Domain=.kipris.or.kr; Path=/
Set-Cookie: TM_CONFIG=G11111111111111111SX11111110011111100; Domain=.kipris.or.kr; Path=/
Set-Cookie: JM_CONFIG=G11111111111111SX01101111110010; Domain=.kipris.or.kr; Path=/
Set-Cookie: AB_CONFIG=G11001111111111111111111111110S10000111000111100001000; Domain=.kipris.or.kr;
Path=/
Set-Cookie: AT_CONFIG=G0000000000000S111110110111; Domain=.kipris.or.kr; Path=/
Set-Cookie: KA_CONFIG=G0000000000000S110111000; Domain=.kipris.or.kr; Path=/
Set-Cookie: K2_CONFIG=G1111111111111111111111S111111111000000000; Domain=.kipris.or.kr; Path=/
Set-Cookie: AD_CONFIG=G11111111111111S1111111111111000; Domain=.kipris.or.kr; Path=/
Connection: keep-alive
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked

3f79
<!DOCTYPE html>
```

Ethernet header / IP header / TCP header
그리고 Message가 있을 경우,
message의 내용까지 전부 확인할 수 있습니다.