정보보호론 lect6_first

12141163 이욱진

homework

1) Make a packet sniffer using winpcap library. Let it dump the raw byte stream of the packet. Compare it with the output of windump.

```
~/De/정보보호론/lect6 sudo ./sniffer
Password:
1-th dev:en0  (No description available)
2-th dev:p2p0  (No description available)
3-th dev:awdl0  (No description available)
4-th dev:llw0  (No description available)
5-th dev:utun0  (No description available)
6-th dev:utun1  (No description available)
7-th dev:lo0  (No description available)
8-th dev:bridge0  (No description available)
9-th dev:en1  (No description available)
10-th dev:en2  (No description available)
11-th dev:en3  (No description available)
12-th dev:en4  (No description available)
13-th dev:gif0  (No description available)
14-th dev:stf0  (No description available)
15-th dev:ap1  (No description available)

~/De/정/lect6 sudo tcpdump -D
1.en0 [Up, Running]
2.p2p0 [Up, Running]
3.awdl0 [Up, Running]
4.llw0 [Up, Running]
5.utun0 [Up, Running]
6.utun1 [Up, Running]
7.lo0 [Up, Running, Loopback]
8.bridge0 [Up, Running]
9.en1 [Up, Running]
10.en2 [Up, Running]
11.en3 [Up, Running]
12.en4 [Up, Running]
13.gif0 [none]
14.stf0 [none]
15.ap1 [none]
```

```c
int main(){
        pcap_if_t *alldevs=NULL;
        char errbuf[PCAP_ERRBUF_SIZE];

        //find all network
        if (pcap_findalldevs(&alldevs, errbuf)==-1){
                printf("dev find failed\n");
                return (-1);
        }
        if (alldevs==NULL){
                printf("no devs found\n");
                return (-1);
        }
        pcap_if_t *d;
        int i;
        for(d=alldevs,i=0; d!=NULL;d=d->next){
                printf("%d-th dev:%s ",++i,d->name);
                if (d->description){
                        printf(" (%s)\n", d->description);

                }
                else
                        printf(" (No description available)\n");
        }
        return (0);
}
```

tcpdump -D 옵션과 동일하게 sniffer코드를 구현하였을 경우 동일하게 네트워크 인터페이스를 확인할 수 있었습니다.

이후 원하는 인터페이스번호를입력받고 파일을 여는작업까지 진행해보았습니다.

```c
 37         printf("enter the interfaace number. ");
 38         scanf("%d",&inum);
 39         for (d=alldevs,i=0;i<inum-1;d=d->next,i++);
 40         if ((fp = pcap_open_live(d->name,
 41                         65536,
 42                         1,
 43                         20,
 44                         errbuf
 45                         ))==NULL){
 46             printf("pcap open failed\n");
 47             pcap_freealldevs(alldevs);
 48             return (-1);
 49         }
 50         printf("pcap oepn successful\n");
 51         return (0);
 52 }
```

2) Improve your sniffer such that it also prints all the fields in ethernet header, ip header, and tcp header. Use ntohs for "short" data type and ntohl for "int" data type in order to display them correclty. For the data part, just show them in hexadecimal numbers.

dest MAC: ......
src MAC: ....
protocol type: ....
IP version: ...
IP header length: .....
.....

Use following structures.

```
struct ether_addr {
    unsigned char ether_addr_octet[6];
};

struct  ether_header {
    struct  ether_addr ether_dhost;
    struct  ether_addr ether_shost;
    unsigned short ether_type;          // 0x0800 for IP
};
struct ip_hdr{
   unsigned char ip_header_len:4;
   unsigned char ip_version:4;
   unsigned char ip_tos;
   unsigned short ip_total_length;
   unsigned short ip_id;
   unsigned char ip_frag_offset:5;
   unsigned char ip_more_fragment:1;
   unsigned char ip_dont_fragment:1;
   unsigned char ip_reserved_zero:1;
   unsigned char ip_frag_offset1;
   unsigned char ip_ttl;
   unsigned char ip_protocol;
   unsigned short ip_checksum;
   unsigned int ip_srcaddr;
   unsigned int ip_destaddr;
};
struct tcp_hdr{
   unsigned short source_port;
   unsigned short dest_port;
   unsigned int sequence;
   unsigned int acknowledge;
   unsigned char ns:1;
   unsigned char reserved_part1:3;
   unsigned char data_offset:4;
   unsigned char fin:1;
   unsigned char syn:1;
   unsigned char rst:1;
   unsigned char psh:1;
   unsigned char ack:1;
   unsigned char urg:1;
   unsigned char ecn:1;
   unsigned char cwr:1;
   unsigned short window;
```

```
  unsigned short checksum;
  unsigned short urgent_pointer;
};
```