<수업내용>

- Insert below #define statements BEFORE #include <stdio.h> (not "pcp.h")

#define WIN32

#define WPCAP

#define HAVE\_REMOTE

#define \_CRT\_SECURE\_NO\_WARNINGS

you should define following functions: print\_raw\_packet(), print\_ether\_header(), print\_ip\_header(), print\_tcp\_header(), print\_data()

The data structures (ip\_hdr, tcp\_hdr) you need are in the lecture note. Just define them in the beginning of your program before any function and use them in above functions.

실제 패킷 출력하는 함수 (print\_raw\_packet, print\_ether\_header,print\_ip\_header, print\_tcp\_header, print\_data)는 직접 정의해야 합니다. 패킷 출력시 필요한 자료구조는 강의노트에 있으므로 프로그램 시작 지점에 정의한 후 사용하기 바랍니다.

<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.

텍스트이(가) 표시된 사진

자동 생성된 설명

sniffer.c 로 available 한 dev 를 확인하였습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

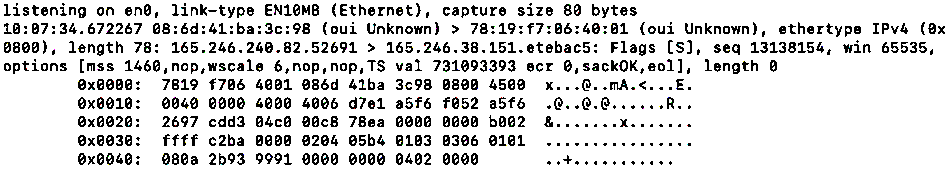
Raw packet 은 위와같이 패킷데이터와, length 을 사용하여 16진수로 출력할 수 있도록 합니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dest MAC Addr | | | | | | Src MAC Addr | | | | | |
| Protocol Type | | | | | | Version | Head Length | | Service Field | | |
| Total Length | | | | | | Identification | | | | | |
| Flags  40 | | | | Flag Offset  00 | | TTL  40 | | | Protocol  06 | | |
| Header Checksum | | | | Source IP Address | | | | | | | |
| Source IP Address | | | | Destination IP Address | | | | | | | |
| Destination IP Address | | | | Source Port | | | | Destination Port | | | |
| Destination Port | | | | Sequence Number | | | | | | | |
| Sequence Number | | | | Acknowledgment Number | | | | | | | |
| Acknowledgment Number | | | | Header Length | Reserved | | N | C | E | U | A |
| P | R | S | F | Window Size | | | | Checksum | | | |
| Checksum | | | | Urgent Pointer | | | | Options | | | |

뭔가 문제가 있지만, 각 값은 대략적으로 패킷은 위치(길이)에 따라 위와 같은 의미를 가집니다



tcpdump 로 가져온 값과 앞서 출력된 값을.비교해보면, 다른 환경에서 실행했기에 MAC 주소가 다르고, IP address, port number가 한 곳만 다르며, window size나 TCP checksum은 때에따라 변화하는 것을 확인할 수 있습니다.

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.

테이블이(가) 표시된 사진

자동 생성된 설명

각 속성값을 가져오기 전에 strct 로 각 헤더의 내부 구조 속성값을 정의합니다.

1. 이더넷 헤더

텍스트이(가) 표시된 사진

자동 생성된 설명

2. IP 헤더

텍스트이(가) 표시된 사진

자동 생성된 설명

3. TCP 헤더

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명 텍스트이(가) 표시된 사진

자동 생성된 설명

출력된 전체 패킷의 값 입니다. ehternet 헤더 부분입니다.

텍스트이(가) 표시된 사진

자동 생성된 설명 텍스트이(가) 표시된 사진

자동 생성된 설명

IP 헤더 부분입니다. tcp 헤더 부분입니다.

앞서 정의한 대로 각 속성값들이 보기좋게 잘 출력되는 것을 확인할 수 있었습니다.