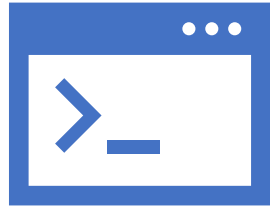




SCP 2021 Summer Project - 2주차

이지훈

2주차_목차



Pcap-test

Last week

Code Review



Beacon flooding

환경 구축

예제

Next week

Pcap-test

Last week

1. Ethernet Header의 src mac / dst mac
2. IP Header의 src ip / dst ip
3. TCP Header의 src port / dst port
4. Payload Data

```
root@ubuntu:~/build-untitled-Desktop-Debug# ./untitled ens33
66 bytes captured, len : 66, res : 1, 0th packet
type : 8
dst : 70:5d:cc:5f:28:c0
src : 00:0c:29:cc:b2:6e
```

```
▶ Frame 1: 84 bytes on wire (672 bits), 84 bytes captured (672 bits) on interface ens33, id 0
▶ Ethernet II, Src: VMware_cc:b2:6e (00:0c:29:cc:b2:6e), Dst: ArubaaHe_01:42:e8 (00:1a:1e:01:42:e8)
▶ Internet Protocol Version 4, Src: 172.31.7.213, Dst: 168.126.63.1
▶ User Datagram Protocol, Src Port: 48300, Dst Port: 53
▶ Domain Name System (query)
```



Pcap-test Code Review

<pcap.h> : pcap 라이브러리 캡처를 위한 헤더파일

<stdbool.h> : true, false, NULL 과 같은 bool 연산자 헤더파일
(컴파일 오류로 13~15줄에 별도 정의)

<stdint.h> : 정수 자료형 정의 헤더파일 (uint8_t)

6~8줄 : 이더넷, IP, TCP 헤더파일 (18~20줄에 구조체 정의)

```
1  #include <pcap.h>
2  // #include <stdbool.h>
3  #include <stdio.h>
4
5  #include <stdint.h>
6  #include <netinet/ip.h>
7  #include <netinet/tcp.h>
8  #include <net/ethernet.h>
9  #include <string.h>
10
11
12
13  #define true 1
14  #define false 0
15  #define NULL 0
16
17
18  struct ether_header *ep;
19  struct ip *iph;
20  struct tcphdr *tcph;
21
```



Pcap-test Code Review

usage() : 사용법을 알려주기 위한 함수

28~34줄 : 매개변수 구조체 정의

36~45줄 : argc 가 2가 아닐 경우, usage 함수 호출
: argv[1] 값을 매개변수에 저장 (인터페이스 값)

```
22 |  
23 ▼ void usage() {  
24     printf("syntax: pcap-test <interface>\n");  
25     printf("sample: pcap-test wlan0\n");  
26 }  
27  
28 ▼ typedef struct {  
29     char* dev_;  
30 } Param;  
31  
32 ▼ Param param = {  
33     .dev_ = NULL  
34 };  
35  
36 ▼ int parse(Param* param, int argc, char* argv[]) {  
37     if (argc != 2) {  
38         usage();  
39         return false;  
40     }  
41  
42     param->dev_ = argv[1];  
43     return true;  
44 }  
45
```



Pcap-test Code Review

48~50줄 : parse 함수가 false 일 때 main 함수 종료

53줄 : pcap 캡처 시작

55~58줄 : 에러 문구 출력

60~62줄 : 패킷 개수, ip주소를 담을 변수

```
46 int main(int argc, char* argv[]) {
47     if (!parse(&param, argc, argv))
48         return -1;
49
50     char errbuf[PCAP_ERRBUF_SIZE];
51
52     pcap_t* pcap = pcap_open_live(param.dev_, BUFSIZ, 1, 1000, errbuf);
53
54     if (pcap == NULL) {
55         fprintf(stderr, "pcap_open_live(%s) return null - %s\n", param.dev_, errbuf);
56         return -1;
57     }
58
59     int num = 0;
60     char src_ip[16];
61     char dst_ip[16];
62 }
```

```
36 int parse(Param* param, int argc, char* argv[]) {
37     if (argc != 2) {
38         usage();
39         return false;
40     }
41 }
```



Pcap-test Code Review

67,68줄 : 패킷캡처 정보 저장 변수 선언

70줄 : 패킷 수신 받기

71~75줄 : 에러 확인 및 에러 출력

```
65 while (true)
66 {
67     struct pcap_pkthdr* header;
68     const u_char* packet;
69
70     int res = pcap_next_ex(pcap, &header, &packet);
71     if (res == 0) continue;
72     if (res == PCAP_ERROR || res == PCAP_ERROR_BREAK) {
73         printf("pcap_next_ex return %d(%s)\n", res, pcap_geterr(pcap));
74         break;
75     }
76 }
```



Pcap-test Code Review

77~79줄 : 각각의 구조체에 캡처한 패킷을 적용

81~85줄 : `inet_ntoa()` 로 네트워크 바이트 순서의
32비트 값을 문자열로 변환

87,88줄 : `ip_offset` 을 받아와 페이로드 구간 지정

```
77     ep = (struct ether_header *)packet;
78     iph = (struct ip *)packet;
79     tcph = (struct tcphdr *)packet;
80
81     char *tmp = inet_ntoa(iph->ip_src);
82     strcpy(src_ip,tmp);
83
84     tmp = inet_ntoa(iph->ip_dst);
85     strcpy(dst_ip,tmp);
86
87     u_int8_t ip_offset = iph->ip_hl;
88     const u_char* packet_payload = packet + sizeof(ep) + ip_offset*4;
89
90
```



Pcap-test Code Review

91줄 : pcap 캡처의 총길이, n번째 패킷 출력

93,94줄 : 이더넷 헤더에서 얻은 정보(Mac_add) 출력

96줄 : IP 헤더에서 얻은 정보(ip_add) 출력

97줄 : TCP 헤더에서 얻은 정보(port_num) 출력

98~104줄 : 페이로드 데이터 출력

109줄 : pcap 캡처 종료

```
91     printf("%u bytes captured, len : %u, %dth packet \n", header->caplen, header->len,num);
92
93     printf("dst : %02x:%02x:%02x:%02x:%02x:%02x\n",ep->ether_dhost[0],ep->ether_dhost[1],ep->ether_dhost[2],ep->ether_dhost[3],ep->ether_dhost[4],ep->ether_dhost[5]);
94     printf("src : %02x:%02x:%02x:%02x:%02x:%02x\n",ep->ether_shost[0],ep->ether_shost[1],ep->ether_shost[2],ep->ether_shost[3],ep->ether_shost[4],ep->ether_shost[5]);
95
96     printf("src : %s, dst : %s\n",src_ip,dst_ip);
97     printf("sport : %d, dport : %d\n",tcph->th_sport,tcph->th_dport);
98     printf("payload : ");
99     for(int i = 0; i < 16; i++)
100     {
101         printf("%02x ",*(packet_payload+i));
102     }
103     printf("\n\n");
104
105     num++;
106 }
107
108 pcap_close(pcap);
109
110 }
111
```

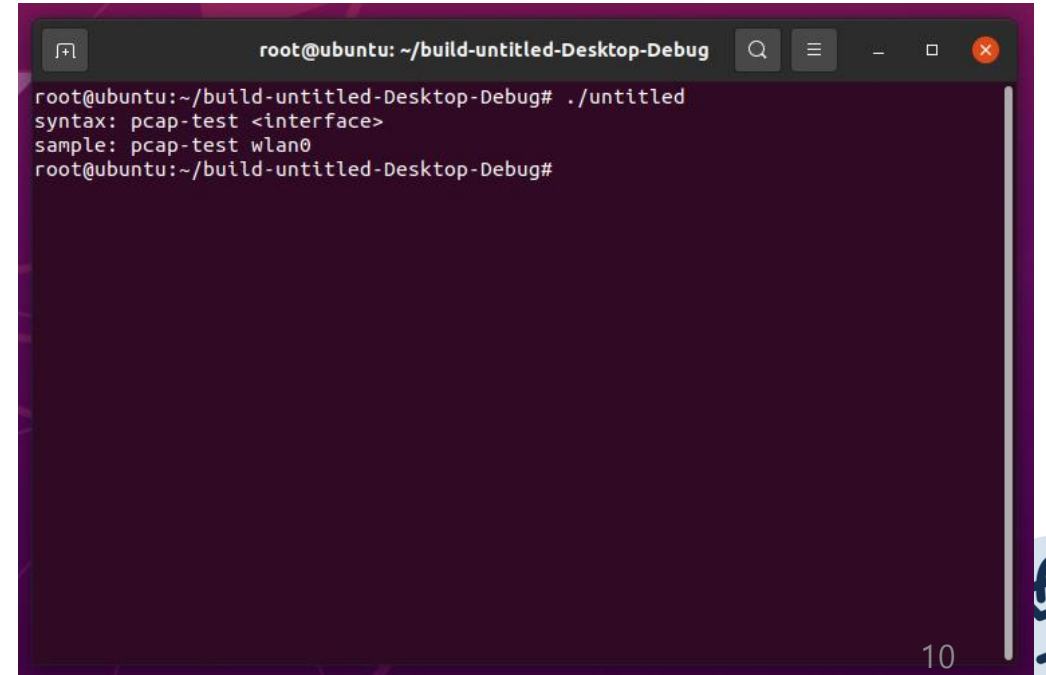


Pcap-test Code Review

```
90 bytes captured, len : 90, 81th packet
dst : 33:33:00:00:00:fb
src : 88:78:73:30:c9:83
src : 134.221.96.1, dst : 133.241.0.36
sport : 13107, dport : 0
payload : 11 01 fe 80 00 00 00 00 00 00 59 a4 53 0b 19 b2

70 bytes captured, len : 70, 82th packet
dst : 01:00:5e:00:00:fb
src : 88:78:73:30:c9:83
src : 8.0.69.0, dst : 0.56.77.117
sport : 1, dport : 94
payload : 08 00 45 00 00 38 4d 75 00 00 01 11 ca 97 c0 a8
```

ens33

A terminal window titled 'root@ubuntu: ~/build-untitled-Desktop-Debug' with search, menu, and window control icons. It shows the execution of a script named 'untitled' which displays the syntax and a sample command for the 'pcap-test' tool.

```
root@ubuntu: ~/build-untitled-Desktop-Debug# ./untitled
syntax: pcap-test <interface>
sample: pcap-test wlan0
root@ubuntu: ~/build-untitled-Desktop-Debug#
```





Beacon flooding 환경 구축

```
sudo apt install cmake libssl-dev
git clone https://github.com/mfontanini/libtins.git
cd libtins
git tag
git checkout <latest tag>
mkdir build
cd build
cmake ..
make -j4
sudo make install
sudo cp /usr/local/lib/libtins* /usr/lib
```

```
-- Installing: /usr/local/include/tins/dot11/dot11_base.h
-- Installing: /usr/local/include/tins/dot11/dot11_data.h
-- Installing: /usr/local/include/tins/dot11/dot11_mgmt.h
-- Installing: /usr/local/include/tins/dot11/dot11_beacon.h
-- Installing: /usr/local/include/tins/dot11/dot11_assoc.h
-- Installing: /usr/local/include/tins/dot11/dot11_auth.h
-- Installing: /usr/local/include/tins/dot11/dot11_probe.h
-- Installing: /usr/local/include/tins/dot11/dot11_control.h
-- Installing: /usr/local/include/tins/offline_packet_filter.h
-- Installing: /usr/local/include/tins/packet_writer.h
-- Installing: /usr/local/include/tins/pktap.h
-- Installing: /usr/local/include/tins/ppi.h
-- Installing: /usr/local/include/tins/sniffer.h
-- Installing: /usr/local/include/tins/tcp_stream.h
jh@ubuntu:~/libtins/build$ ls
CMakeCache.txt      cxx_declval          lib
CMakeFiles          cxx_functional       libtinsConfig.cmake
cmake_install.cmake cxx_initializer_list  libtinsConfigVersion.cmake
cmake_uninstall.cmake cxx_noexcept         libtins.pc
cxx_builtin-swap    cxx_rvalue-references libtinsTargets.cmake
cxx_chrono          examples             Makefile
cxx_decltype        install_manifest.txt src
jh@ubuntu:~/libtins/build$ sudo cp /usr/local/lib/libtins* /usr/lib
jh@ubuntu:~/libtins/build$
```



Beacon flooding 예제

.pro 파일에 LIBS += -ltins 필수!

```
1  #include <unistd.h>
2  #include <list>
3  #include <tins/tins.h>
4  using namespace std;
5  using namespace Tins;
6  int main()
7  {
8      list<string> ssidList {
9          "1.",
10         "2.",
11         "3.",
12         "4.",
13         "5.",
14         "6."
15     };
16     list<string>::iterator it = ssidList.begin();
17     while (true) {
18         RadioTap tap;
19
20         Dot11::address_type ap = "00:11:22:33:44:55";
21         Dot11::address_type broadcast = "ff:ff:ff:ff:ff:ff";
22         Dot11Beacon beacon(broadcast, ap);
23         beacon.addr4(ap);
24         beacon.ssid(*it);
25         beacon.ds_parameter_set(10);
26         beacon.supported_rates({ 1.0f, 5.5f, 11.0f });
27         tap.inner_pdu(beacon);
28
29         PacketSender sender("wlan0");
30         sender.send(tap);
31         usleep(10000);
32
33         if (++it == ssidList.end())
34             it = ssidList.begin();
35     }
36 }
37
```





Beacon flooding 예제

를! 하려고 했으나..

랜카드 인식이 안되는 상황이 발생하여 해결 후 다음주에 진행 예정..





Q & A