

BOB 3차 교육

무선 네트워크 Airodump 과제

2024.01.19



BOB 13th 컨설팅트랙

이진

ari.seebeyond@gmail.com

목차

1 // 사용 툴

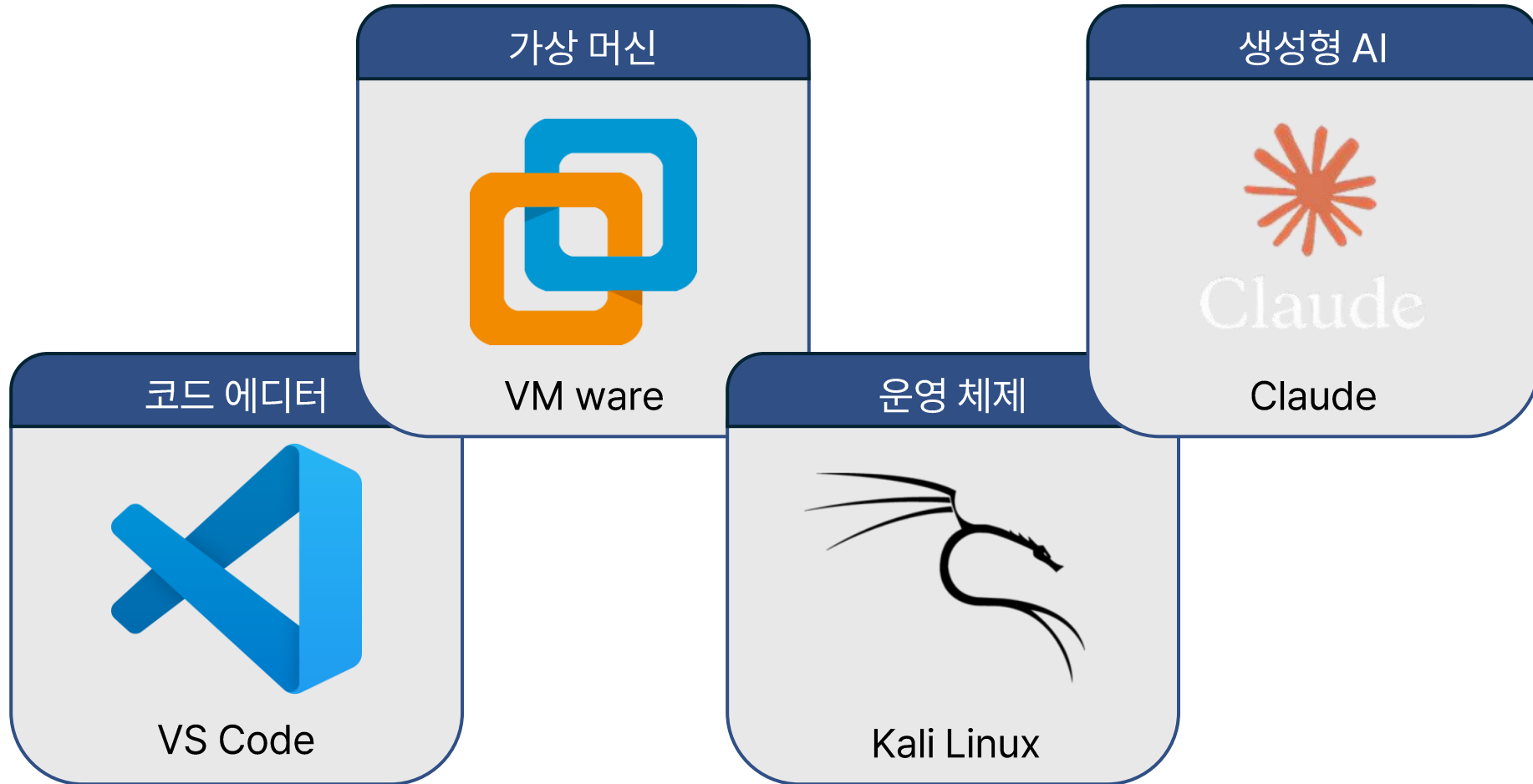
2 // 결과물

3 // 제작과정

4 // 코드리뷰

5 // 참고 문헌

1. 사용 툴



2. 결과물

airodump



BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
DC:F7:19:6B:07:03	-86	0	0 0	11	130	WPA2	CCMP	PSK	KITRI_Mentee
DC:F7:19:6B:07:00	-87	0	0 0	11	130	WPA2	CCMP	PSK	KITRI_BoB
DC:F7:19:6A:82:A1	-66	3	0 0	11	130	WPA2	CCMP	PSK	KITRI_Mentee2
DC:F7:19:6A:82:A0	-66	2	0 0	11	130	WPA2	CCMP	PSK	KITRI_BoB
B0:A7:B9:36:C3:0F	-78	2	0 0	8	260	WPA2	CCMP	PSK	sybon
56:45:73:D6:5B:63	-58	1	0 0	6	360	WPA2	CCMP	PSK	AndroidHotspot9798
DC:F7:19:6B:04:E1	-86	2	0 0	6	130	WPA2	CCMP	PSK	KITRI_Mentee2
DC:F7:19:6B:04:E0	-66	2	0 0	6	130	WPA2	CCMP	PSK	KITRI_BoB
DC:F7:19:6B:06:22	-70	1	0 0	1	130	WPA2	CCMP	PSK	KITRI_Guest
DC:F7:19:6B:06:21	-67	2	0 0	1	130	WPA2	CCMP	PSK	KITRI_Mentee2
D8:BC:38:F8:FE:FD	-78	2	0 0	1	135	OPN			SmallTV PRO
00:00:00:00:00:00	-1	0	2 0	1	-1	OPN			<length: 0>
58:86:94:7F:94:4C	-63	4	0 0	2	270	WPA2	CCMP	PSK	together2G

jindump



BSSID	PWR	Beacons	#Data	CH	MB	ENC	CIPHER	AUTH	ESSID
04:D9:F5:FA:86:28	-84	1	0	9	54	WPA2	CCMP	PSK	ASUS_28_2G
56:45:73:D6:5B:63	-64	30	0	6	54	WPA2	CCMP	PSK	AndroidHotspot9798
58:86:94:AD:7F:BA	-68	2	0	11	54	WPA2	CCMP	PSK	KITRI_Guest
58:86:94:C8:8E:48	-84	2	0	7	54	WPA2	CCMP	PSK	cafe24h
B0:A7:B9:36:C3:0F	-80	1	0	7	54	WPA2	CCMP	PSK	sybon
DC:F7:19:6B:04:E0	-76	24	0	6	54	WPA2	CCMP	PSK	KITRI_BoB
DC:F7:19:6B:04:E1	-76	12	0	6	54	WPA2	CCMP	PSK	KITRI_Mentee2
DC:F7:19:6B:04:E2	-76	22	0	6	54	WPA2	CCMP	PSK	KITRI_Guest
DC:F7:19:6B:04:E3	-72	1	0	6	54	WPA2	CCMP	PSK	KITRI_Mentee

3. 제작과정 | (1) pcap 라이브러리를 통한 바이너리 수집 및 분석

[prompt]
C++ pcap 라이브러리를 사용해서 패킷을 캡처 해 바이트를 출력하는 코드를 짜줘



```

6 void packet_handler(u_char *user, const struct pcap_pkthdr *pkthdr, const u_char *packet) {
7     // 패킷 길이 출력
8     std::cout << "Packet captured. Length: " << pkthdr->len << std::endl;
9
10    // 패킷 데이터를 16진수로 출력
11    for(unsigned int i = 0; i < pkthdr->len; i++) {
12        printf("%02x ", packet[i]);
13        if((i + 1) % 16 == 0) printf("\n");
14    }
15    printf("\n\n");
16 }
17
18 int main(int argc, char *argv[]) {
19
20
21     char errbuf[PCAP_ERRBUF_SIZE];
22     pcap_t *handle;
23     std::string interface = argv[1];
24
25     // 지정된 인터페이스 열기
26     handle = pcap_open_live(interface.c_str(), BUFSIZ, 1, 1000, errbuf);
27     if (handle == NULL) {
28         std::cerr << "Could not open device " << interface << ": " << errbuf << std::endl;
29         return 1;
30     }
31
32     std::cout << "Starting packet capture on interface: " << interface << std::endl;
33
34     // 패킷 캡처 시작
35     pcap_loop(handle, -1, packet_handler, NULL);
36
37     pcap_close(handle);
38     return 0;
39 }

```

Pcap 라이브러리를 사용하여
패킷을 캡처하는 소스코드

Packet captured. Length: 261

```

00 00 1a 00 2f 48 00 00 22 2b ca 07 00 00 00 00
00 02 85 09 a0 00 ac 00 00 00 80 00 00 00 ff ff
ff ff ff ff 18 c5 01 a8 66 3a 18 c5 01 a8 66 3a
90 7a 0a e8 95 23 11 00 00 00 64 00 31 04 00 09
55 2b 4e 65 74 36 36 33 38 01 08 82 84 8b 96 0c
12 18 24 03 01 07 05 04 00 05 00 00 07 06 4b 33
20 01 0d 1e 2a 01 00 32 04 30 48 60 6c 2d 1a ad
09 03 ff ff 00 00 00 00 00 00 00 00 00 01 00
00 00 00 00 00 00 00 00 3d 16 07 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 7f 08 04 00 0f 02 00 00 00 40 bf 0c b2 51 80
33 fa ff 00 00 fa ff 00 20 c0 05 00 00 00 fc ff
dd 18 00 50 f2 02 01 01 00 00 03 a4 00 00 27 a4
00 00 42 43 5e 00 62 32 2f 00 dd 09 00 03 7f 01
01 00 00 ff 7f dd 08 8c fd f0 01 01 02 01 00 30
14 01 00 00 0f ac 04 01 00 00 0f ac 04 01 00 00
0f ac 02 00 00

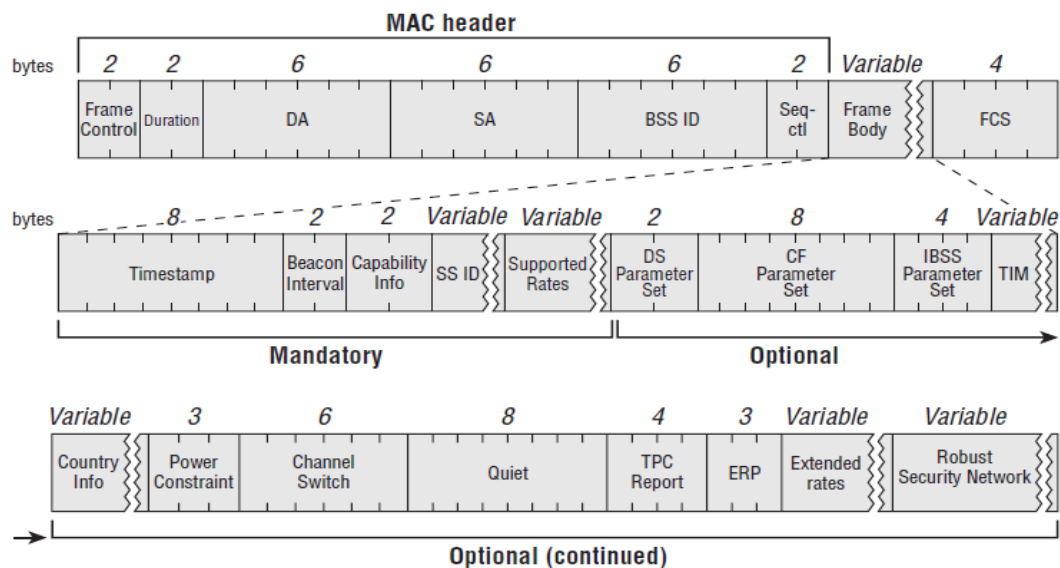
```

캡처 된 Beacon Frame 패킷

3. 제작과정 | (1) pcap 라이브러리를 통한 바이너리 수집 및 분석

[prompt]
(패킷 캡처 결과물 첨부) 네트워크 패킷 캡처 결과인데 분석해줘 자세하게

FIGURE 4.5 Beacon frame structure



IE의 Element ID :

- 0 (0x00): SSID Parameter Set
- 1 (0x01): Supported Rates
- 3 (0x03): DS Parameter Set
- 48 (0x30): RSN (Robust Security Network)
- 221 (0xDD): Vendor Specific

Packet captured. Length: 261

```

00 00 1a 00 2f 48 0 Radiotap 헤더 b ca 07 00 00 00 00
00 02 85 09 a0 00 ac 00 00 00 8 FCF 0 00 00 ff ff
ff ff ff ff 18 c5 BSSID 66 3a 18 c5 01 a8 66 3a
90 7a 0a e8 95 23 11 00 00 00 64 00 31 04 00 09
55 2b 4e 65 SSID 36 36 33 38 01 0 Supported Rates 5 0c
12 18 24 0 channel 7 05 04 00 05 00 00 07 06 4b 33
20 01 0d 1e 2a 01 00 32 04 30 48 60 6c 2d 1a ad
09 03 ff ff 00 00 00 00 00 00 00 00 00 00 01 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 7f 08 04 00 0f 02 00 00 40 bf 0c b2 51 80
33 fa ff 00 00 fa ff 00 20 c0 05 00 00 00 fc ff
dd 18 00 50 f2 02 01 01 00 00 03 a4 00 00 27 a4
00 00 42 43 5e 00 62 32 2f 00 dd 09 00 03 7f 01
01 00 00 ff 7f dd 08 8c fd f0 01 01 02 01 00 30
14 01 00 00 0f ac 04 RSN IE 00 0f ac 04 01 00 00
0f ac 02 00 00
  
```

3. 제작과정 | (2) 패킷 분석 – Radiotap 헤더

[prompt]
Radiotap 에 대해서 설명해줘



Packet captured. Length: 261

```
00 00 1a 00 2f 48 00 00 22 2b ca 07 00 00 00 00
00 02 85 09 a0 00 ac 00 00 00 80 00 00 00 ff ff
ff ff ff ff 18 c5 01 a8 66 3a 18 c5 01 a8 66 3a
90 7a 0a e8 95 23 11 00 00 00 64 00 31 04 00 09
55 2b 4e 65 74 36 36 33 38 01 08 82 84 8b 96 0c
12 18 24 03 01 07 05 04 00 05 00 00 07 06 4b 33
20 01 0d 1e 2a 01 00 32 04 30 48 60 6c 2d 1a ad
09 03 ff ff 00 00 00 00 00 00 00 00 00 01 00
00 00 00 00 00 00 00 00 00 3d 16 07 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 7f 08 04 00 0f 02 00 00 00 40 bf 0c b2 51 80
33 fa ff 00 00 fa ff 00 20 c0 05 00 00 00 fc ff
dd 18 00 50 f2 02 01 01 00 00 03 a4 00 00 27 a4
00 00 42 43 5e 00 62 32 2f 00 dd 09 00 03 7f 01
01 00 00 ff 7f dd 08 8c fd f0 01 01 02 01 00 30
14 01 00 00 0f ac 04 01 00 00 0f ac 04 01 00 00
0f ac 02 00 00
```

Radiotap 헤더 정의 및 역할

802.11 무선네트워크 패킷을 캡처할 때 사용되는 헤더 포맷.
무선 네트워크 인터페이스 카드(NIC)가 패킷을 캡처할 때
추가하는 메타데이터

Radiotap 헤더 분석

- 리틀 엔디안 방식 사용 (낮은 주소에 데이터의 하위 바이트부터 저장하는 방식, 필드 단위)
- 헤더의 길이는 0x1A 로 26바이트
- [Version: 0x0000] [Header Length: 0x001a] [Present flags: 0x0000482f]
[MAC timestamp: 0x00000007ca2b22]
[Flags: 0x00] [Rate: 1Mbps] [Channel: 2437MHz] [Ch flags: 0x00a0] **[Antenna signal: -84dBm]** [Padding: 0x000000]

3. 제작과정 | (2) 패킷 분석 – Frame Control Field

[prompt]

Frame Control Field 에 대해서 설명해줘



Packet captured. Length: 261

```
00 00 1a 00 2f 48 00 00 22 2b ca 07 00 00 00 00
00 02 85 09 a0 00 ac 00 00 00 80 00 00 00 ff ff
ff ff ff ff 18 c5 01 a8 66 3a 18 c5 01 a8 66 3a
90 7a 0a e8 95 23 11 00 00 00 64 00 31 04 00 09
55 2b 4e 65 74 36 36 33 38 01 08 82 84 8b 96 0c
12 18 24 03 01 07 05 04 00 05 00 00 07 06 4b 33
20 01 0d 1e 2a 01 00 32 04 30 48 60 6c 2d 1a ad
09 03 ff ff 00 00 00 00 00 00 00 00 00 01 00
00 00 00 00 00 00 00 00 00 3d 16 07 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 7f 08 04 00 0f 02 00 00 00 40 bf 0c b2 51 80
33 fa ff 00 00 fa ff 00 20 c0 05 00 00 00 fc ff
dd 18 00 50 f2 02 01 01 00 00 03 a4 00 00 27 a4
00 00 42 43 5e 00 62 32 2f 00 dd 09 00 03 7f 01
01 00 00 ff 7f dd 08 8c fd f0 01 01 02 01 00 30
14 01 00 00 0f ac 04 01 00 00 0f ac 04 01 00 00
0f ac 02 00 00
```

Frame Control Field 정의 및 역할

해당 프레임이 어떤 종류인지를 나타내는 데 사용.

해당 패킷은 Beacon 프레임임을 나타냄

=> 따라서 Beacon 프레임 패킷이 몇 번 오는지 카운트
함으로 써 Beacon 수 출력 가능

Frame Control Field 분석

- [Frame Control Field: 0x0080] (Version: 0x0, Type: Management, Subtype: Beacon, Flags: ToDS=0, FromDS=0, MoreFrag=0, Retry=0, PwrMgmt=0, MoreData=0, Protected=0, Order=0)
- 첫 2비트 : 프레임 타입 (Management frame)
- 4비트 : 서브타입 (8인 경우 - Beacon : 802.11 무선네트워크에서 AP가 주기적으로 브로드 캐스트하는 관리 프레임)

3. 제작과정 | (2) 패킷 분석 – BSSID, SSID

[prompt]
BSSID, SSID 에 대해서 설명해줘



```
Packet captured. Length: 261
00 00 1a 00 2f 48 00 00 22 2b ca 07 00 00 00 00
00 02 85 09 a0 00 ac 00 00 00 80 00 00 00 ff ff
ff ff ff ff 18 c5 01 a8 66 3a 18 c5 01 a8 66 3a
90 7a 0a e8 95 23 11 00 00 00 64 00 31 04 00 09
55 2b 4e 65 74 36 36 33 38 01 08 82 84 8b 96 0c
12 18 24 03 01 07 05 04 00 05 00 00 07 06 4b 33
20 01 0d 1e 2a 01 00 32 04 30 48 60 6c 2d 1a ad
09 03 ff ff 00 00 00 00 00 00 00 00 00 01 00
00 00 00 00 00 00 00 00 00 00 3d 16 07 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 7f 08 04 00 0f 02 00 00 00 40 bf 0c b2 51 80
33 fa ff 00 00 fa ff 00 20 c0 05 00 00 00 fc ff
dd 18 00 50 f2 02 01 01 00 00 03 a4 00 00 27 a4
00 00 42 43 5e 00 62 32 2f 00 dd 09 00 03 7f 01
01 00 00 ff 7f dd 08 8c fd f0 01 01 02 01 00 30
14 01 00 00 0f ac 04 01 00 00 0f ac 04 01 00 00
0f ac 02 00 00
```

BSSID 정의 및 역할

Basic Service Set 을 고유하게 식별하는 6바이트 식별자.
BSS 모드에 따라 설정방식에 차이를 보임.
Infrastructure Mode 의 경우 AP 의 무선 인터페이스
MAC 주소가 BSSID 로 사용됨

SSID 정의 및 역할

무선 네트워크의 이름을 나타내는 식별자.
최대 32 바이트 길이로 ASCII 문자로 구성.
Element ID 는 0x00

3. 제작과정 | (2) 패킷 분석 – Supported Rates

[prompt]
Supported Rates 에 대해서 설명해줘



```
Packet captured. Length: 261
00 00 1a 00 2f 48 00 00 22 2b ca 07 00 00 00 00
00 02 85 09 a0 00 ac 00 00 00 80 00 00 00 ff ff
ff ff ff ff 18 c5 01 a8 66 3a 18 c5 01 a8 66 3a
90 7a 0a e8 95 23 11 00 00 00 64 00 31 04 00 09
55 2b 4e 65 74 36 36 33 38 01 08 82 84 8b 96 0c
12 18 24 03 01 07 05 04 00 05 00 00 07 06 4b 33
20 01 0d 1e 2a 01 00 32 04 30 48 60 6c 2d 1a ad
09 03 ff ff 00 00 00 00 00 00 00 00 00 01 00
00 00 00 00 00 00 00 00 00 00 3d 16 07 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 7f 08 04 00 0f 02 00 00 00 40 bf 0c b2 51 80
33 fa ff 00 00 fa ff 00 20 c0 05 00 00 00 fc ff
dd 18 00 50 f2 02 01 01 00 00 03 a4 00 00 27 a4
00 00 42 43 5e 00 62 32 2f 00 dd 09 00 03 7f 01
01 00 00 ff 7f dd 08 8c fd f0 01 01 02 01 00 30
14 01 00 00 0f ac 04 01 00 00 0f ac 04 01 00 00
0f ac 02 00 00
```

Supported Rates 정의 및 역할

AP가 지원하는 데이터 전송 속도를 알림, AP는 최소 하나의 mandatory rate(basic rate)를 설정해야 함

Supported Rates 분석

- 각 옥텟은 하나의 지원되는 전송 속도를 나타냄
- 각 옥텟의 7번째 비트:
 - 1 = basic rate (필수 속도)
 - 0 = supported rate (지원 속도)
 - 0-6 비트: 500kbps 단위로 데이터 속도 값을 지정

3. 제작과정 | (2) 패킷 분석 – Channel

[prompt]
Channel 에 대해서 설명해줘



```
Packet captured. Length: 261
00 00 1a 00 2f 48 00 00 22 2b ca 07 00 00 00 00
00 02 85 09 a0 00 ac 00 00 00 80 00 00 00 ff ff
ff ff ff ff 18 c5 01 a8 66 3a 18 c5 01 a8 66 3a
90 7a 0a e8 95 23 11 00 00 00 64 00 31 04 00 09
55 2b 4e 65 74 36 36 33 38 01 08 82 84 8b 96 0c
12 18 24 03 01 07 05 04 00 05 00 00 07 06 4b 33
20 01 0d 1e 2a 01 00 32 04 30 48 60 6c 2d 1a ad
09 03 ff ff 00 00 00 00 00 00 00 00 00 01 00
00 00 00 00 00 00 00 00 00 3d 16 07 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 7f 08 04 00 0f 02 00 00 00 40 bf 0c b2 51 80
33 fa ff 00 00 fa ff 00 20 c0 05 00 00 00 fc ff
dd 18 00 50 f2 02 01 01 00 00 03 a4 00 00 27 a4
00 00 42 43 5e 00 62 32 2f 00 dd 09 00 03 7f 01
01 00 00 ff 7f dd 08 8c fd f0 01 01 02 01 00 30
14 01 00 00 0f ac 04 01 00 00 0f ac 04 01 00 00
0f ac 02 00 00
```

Channel 정의 및 역할

통신 과정에서 간섭을 방지하기 위한 특정 주파수 대역.
채널1(2.412GHz) 를 시작으로 5MHZ 간격으로
나뉘어짐

Channel

- AP가 실제로
운영되고 있는 채널
정보
- 03: IE ID
- 01: 길이(1 byte)
- 07: 채널 번호(7)



Radiotap channel

- 패킷을 캡처한 무선
인터페이스가 현재
모니터링하고 있는
채널 정보

3. 제작과정 | (2) 패킷 분석 – RSN IE

[prompt]
RSN IE 에 대해서 설명해줘



Packet captured. Length: 261

```
00 00 1a 00 2f 48 00 00 22 2b ca 07 00 00 00 00
00 02 85 09 a0 00 ac 00 00 00 80 00 00 00 ff ff
ff ff ff ff 18 c5 01 a8 66 3a 18 c5 01 a8 66 3a
90 7a 0a e8 95 23 11 00 00 00 64 00 31 04 00 09
55 2b 4e 65 74 36 36 33 38 01 08 82 84 8b 96 0c
12 18 24 03 01 07 05 04 00 05 00 00 07 06 4b 33
20 01 0d 1e 2a 01 00 32 04 30 48 60 6c 2d 1a ad
09 03 ff ff 00 00 00 00 00 00 00 00 00 01 00
00 00 00 00 00 00 00 00 00 00 3d 16 07 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 7f 08 04 00 0f 02 00 00 00 40 bf 0c b2 51 80
33 fa ff 00 00 fa ff 00 20 c0 05 00 00 00 fc ff
dd 18 00 50 f2 02 01 01 00 00 03 a4 00 00 27 a4
00 00 42 43 5e 00 62 32 2f 00 dd 09 00 03 7f 01
01 00 00 ff 7f dd 08 8c fd f0 01 01 02 01 00 30
14 01 00 00 0f ac 04 01 00 00 0f ac 04 01 00 00
0f ac 02 00 00
```

RSN IE 정의 및 역할

802.11i 보안 표준에서 정의된 정보 요소.
WPA2 네트워크의 보안 설정과 기능을 클라이언트에게 알리는 역할

RSN IE 분석

- [Element ID: 0x30] [Length: 0x14]
[Version: 0x01] [Group Cipher Suite: 0x000f ac04] [Pairwise Cipher Suite Count: 0x01] [Pairwise Cipher Suite: 0x000f ac04] [AKM Suite Count: 0x01] [AKM Suite: 0x000f ac02]
- 암호화 방식의 경우 04 로 CCMP (AES) 사용
- 02 : PSK 로 개인용

3. 제작과정 | (3) 과제 코드 작성

[prompt]
(report airodump 과제 전체 복사 후 붙여넣기)



[pwr, essid] 부분 오류 제외 정상 출력



[prompt]
아 이상하잖아 단디 똑디 제대로 렛츠고



아하, 네. 더 정확한 패킷 분석을 위해 코드를 전면 수정하겠습니다:

```
cpp Copy

#include <pcap.h>
#include <iostream>
#include <cstring>
#include <iomanip>
#include <vector>
#include <map>
#include <unistd.h>
#include <ieee80211.h>

#pragma pack(1)
struct radiotap_header {
    uint8_t version;
    uint8_t pad;
```

코드 정상 출력



4. 코드리뷰 | (1) 개요 (25.01.14 수업 참조)

1

IE 의 경우 Tag 를 기준으로 구분이 가능함 , Tag 1바이트와 길이 1바이트 이후 파싱 , 상속 사용

2

BSSID 를 기준으로 MAP 자료구조를 사용하여 구조체 정의

3

패킷 캡처 과정 스레드로 빼서 구현

4

메모리 할당 시, 메모리 해제 필수

5

구조체 이름 최대한 간결하게

4. 코드리뷰 | (2) pcap 라이브러리 이해

[prompt]

pcap 라이브러리에 대해 설명해주고 활용방법에 대해서 알려줘



[주요 기능]

- 네트워크 인터페이스에서 실시간으로 패킷 캡처
- 패킷 필터링 (BPF - Berkeley Packet Filter 사용)
- 캡처된 패킷 저장 및 읽기
- 네트워크 통계 수집



Pcap	설명
pcap_t *pcap_open_live(const char *device, int snaplen, int promisc, int to_ms, char *errbuf)	실시간 캡처 세션 시작 함수
const u_char *pcap_next(pcap_t *p, struct pcap_pkthdr *h)	다음 패킷 가져오는 함수
PCAP_ERRBUF_SIZE	libpcap에서 정의된 상수로, 일반적으로 256바이트
pcap_t* handle	libpcap의 패킷 캡처 세션을 관리하는 핸들러(handler)

4. 코드리뷰 | (3) 코드 전체 구성도

[prompt]

(Claude 가 짠 코드 전문 복붙) 이거 전체 코드 구성도 그려줘



[Main Program]

[packet_parser] (Abstract Base Class)

virtual parse()

virtual show()

[dot11_parser] (Derived from packet_parser)

parse()

show()

[std::map<string, ap_info>]

[IE System]

[IE] (Abstract Base Class)

virtual parse()

[TaggedParameter] (Derived from IE)

[SSID_IE]

[DS_Parameter_IE]

[Supported_Rates_IE]

[Extended_Supported_Rates_IE]

[RSN_IE]

[ap_info] (Structure)

bssid

ssid

channel

enc

beacons

pwr

cipher

auth

last_time

data

data_rate

updated

4. 코드리뷰 | (4) 코드리뷰 - AP 구조체 및 IE 파싱

AP 구조체

```
struct ap_info {
    std::string bssid;
    std::string ssid;
    int channel;
    std::string enc;
    unsigned int beacons;
    int pwr;
    std::string cipher;
    std::string auth;
    time_t last_time;
    int data;
    int data_rate;
    bool updated;

    ap_info() : channel(0), beacons(0), pwr(0),
last_time(0), data(0), data_rate(0), updated(false) {}
};
```

IE에 따른 Tag 파싱

```
std::unique_ptr<IE> IE::create(uint8_t tag, uint8_t len,
const uint8_t* data) {
    switch(tag) {
        case 0: // SSID
            return std::unique_ptr<IE>(new SSID_IE(tag, len,
data));
        case 1: // Supported Rates
            return std::unique_ptr<IE>(new
Supported_Rates_IE(tag, len, data));
        case 3: // DS Parameter Set (Channel)
            return std::unique_ptr<IE>(new
DS_Parameter_IE(tag, len, data));
        case 48: // RSN
            return std::unique_ptr<IE>(new RSN_IE(tag, len,
data));
        case 50: // Extended Supported Rates
            return std::unique_ptr<IE>(new
Extended_Supported_Rates_IE(tag, len, data));
        default:
            return nullptr; }};
```

4. 코드리뷰 | (4) 코드리뷰 - BSSID

BSSID

PWR

Beacons

#Data

CH

MB

ENC

CIPHER

AUTH

ESSID

```
char bssid[18];
```

```
const uint8_t* bssid_ptr = frame + 16;
```

```
std::snprintf(bssid, sizeof(bssid),
"%02X:%02X:%02X:%02X:%02X:%02X",
bssid_ptr[0], bssid_ptr[1], bssid_ptr[2],
bssid_ptr[3], bssid_ptr[4], bssid_ptr[5]);
ap.bssid = bssid;
```

```
void parse(const uint8_t* pkt, int len) override {
    uint16_t rtap_len = *(uint16_t*)(pkt + 2);
    const uint8_t* frame = pkt + rtap_len;
```

설명

- rtap_len 의 경우 radio tap 헤더의 길이
- Frame 에서 16 바이트 뒤에 bssid 위치
- BSSID 로 MAC 주소가 저장되므로, MAC 주소형식으로 출력

4. 코드리뷰 | (4) 코드리뷰 – PWR, Beacons

BSSID

PWR

Beacons

#Data

CH

MB

ENC

CIPHER

AUTH

ESSID

```
class dot11_parser : public packet_parser {
    ( ... )
    ap.pwr = -(int8_t)pkt[22];
    ( ... )
}
```

설명

- PWR 은 Linux 환경, airmmon-ng으로 설정된 모니터 모드, 일반적인 WiFi 카드인 경우 일반적으로 Radiotap 헤더안 22 바이트 부분에 위치함

```
auto& stored = aps[ap.bssid];
    ( ... )
    stored.beacons++;
```

설명

- 해당 코드의 경우 Beacon 프레임일 경우에만, 코드를 진행함
- 캡처될때마다 각각의 BSSID 를 기준으로 Beacons 수를 하나씩 증가시킴

4. 코드리뷰 | (4) 코드리뷰 – #Data , CH

BSSID

PWR

Beacons

#Data

CH

MB

ENC

CIPHER

AUTH

ESSID

```
if (to_ds == 0 && from_ds == 1)
    bssid_ptr = frame + 16;
else if (to_ds == 1 && from_ds == 0)
    bssid_ptr = frame + 4;
```

```
else if (type == 0x0020 || type == 0x0028) {
    auto it = aps.find(bssid);
    if (it != aps.end()) {
        it->second.data++;
        it->second.updated = true;
        need_header = true;}}}
```

설명

- #Data 의 경우 Data 프레임들의 총 개수를 카운트 함. 클라이언트와 AP 사이에서 실제로 교환되는 데이터 패킷들.Type 값이 0x20(데이터) 또는 0x28(QoS 데이터)인 프레임들
- 프레임 타입에 따라 BSSID 위치가 다름
- 데이터 프레임인 경우 카운트

```
class DS_Parameter_IE : public TaggedParameter {
public:
    DS_Parameter_IE(uint8_t tag, uint8_t len, const uint8_t* data)
        : TaggedParameter(tag, len, data) {}

    void parse(ap_info& ap) override {
        if (length > 0) {
            ap.channel = data[0]; } };
```

설명

- Beacon frame 의 IE 태그가 3이면 data[0] 의 값을 채널번호로 해석하여 ap.channel 에 저장

4. 코드리뷰 | (4) 코드리뷰 – MB, ENC

BSSID

PWR

Beacons

#Data

CH

MB

ENC

CIPHER

AUTH

ESSID

```
class Supported_Rates_IE : public TaggedParameter {
public:
    Supported_Rates_IE(uint8_t tag, uint8_t len, const uint8_t* data)
        : TaggedParameter(tag, len, data) {}

    void parse(ap_info& ap) override {
        if (length > 0) {
            for (size_t i = 0; i < length; i++) {
                int rate = (data[i] & 0x7F) * 500;
                ap.data_rate = std::max(ap.data_rate, rate);
            }
        }
    }
};
```

설명

- data[i] & 0x7F로 실제 속도 값을 추출한 후 500을 곱해 Kbps 단위로 변환
- 여러 개의 속도 값들 중 가장 높은 값을 선택하기 위해 std::max를 사용하여 ap.data_rate를 업데이트하며, 이 값이 나중에 출력될 때는 MB 단위로 표시

```
class RSN_IE : public TaggedParameter {
public:
    RSN_IE(uint8_t tag, uint8_t len, const uint8_t* data)
        : TaggedParameter(tag, len, data) {}

    void parse(ap_info& ap) override {
        if (length < 2) return;

        ap.enc = "WPA2";
        ( ... )
    }
};
```

설명

- rsn_ie 가 있는 경우 enc 가 wpa2로 정의되어 출력

4. 코드리뷰 | (4) 코드리뷰 – Cipher, Auth

BSSID

PWR

Beacons

#Data

CH

MB

ENC

CIPHER

AUTH

ESSID

```

if (ptr + 4 * pair_count > data + length) return;
for (int i = 0; i < pair_count; i++) {
    if (memcmp(ptr, "\x00\x0F\xAC", 3) == 0) {
        switch(ptr[3]) {
            case 4: ap.cipher = "CCMP"; break;
            case 2: ap.cipher = "TKIP"; break;
        }
    }
    ptr += 4;
}

```

설명

- RSN IE에서 암호화 방식을 파싱하며, 처음 3바이트가 IEEE 802.11 표준 OUI(00:0F:AC)인지 확인
- 4번째 바이트 값에 따라 암호화 방식을 결정하는데, 값이 4면 CCMP(AES), 2면 TKIP로 ap.cipher를 설정

```

if (ptr + 2 > data + length) return;
uint16_t akm_count = *(uint16_t*)ptr;
ptr += 2;

if (ptr + 4 * akm_count > data + length) return;
for (int i = 0; i < akm_count; i++) {
    if (memcmp(ptr, "\x00\x0F\xAC", 3) == 0)
    {switch(ptr[3]) {
        case 2: ap.auth = "PSK"; break;
        case 1: ap.auth = "MGT"; break;
    }} ptr += 4;}

```

설명

- 4번째 바이트 값이 2면 PSK(개인용), 1이면 MGT(기업용) 인증 방식으로 ap.auth를 설정

4. 코드리뷰 | (4) 코드리뷰 - ESSID

BSSID PWR Beacons #Data CH MB ENC CIPHER AUTH **ESSID**

설명

- SSID_IE는 태그 번호 0번으로, AP의 네트워크 이름(SSID)을 파싱하는 클래스
- data 포인터가 가리키는 바이트 배열을 문자열(SSID)로 변환하여 ap.ssid에 저장하며, length만큼의 길이를 사용해 문자열을 생성

```
class SSID_IE : public TaggedParameter {
public:
    SSID_IE(uint8_t tag, uint8_t len, const uint8_t*
data)
        : TaggedParameter(tag, len, data) {}

    void parse(ap_info& ap) override {
        if (length > 0) {
            ap.ssid.assign(reinterpret_cast<const
char*>(data), length);
        }
    }
};
```

6. 참고 문헌

- 차재복. (2021, October 7). 802.11 Frame Control Field (802.11 프레임 제어 필드). 정보통신기술용어해설. <http://www.ktword.co.kr/test/view/view.php?no=2320>
- Beacon frame. (2024, February 26). In Wikipedia. https://en.wikipedia.org/wiki/Beacon_frame
- Nayanajith, R. (2014, October 8). "802.11 Mgmt : Beacon Frame". *mrn-cciew*. Retrieved January 19, 2025, from <https://mrncciew.com/2014/10/08/802-11-mgmt-beacon-frame/>