WI-DP 침입탐지 및 방지시스템 구현

(Wireless Intrusion Detection & Prevention)

팀 명:	패킷사냥꾼
지도 교수:	유승재 교수님
팀 장:	신현석
팀 원:	전경준
	김경수
	박민석
	유원상
	윤서완

2018. 11.

중부대학교 정보보호학과

목 차

1. 서론

2. 관련연구

3. 본론

3.1 구동

3.2 관리자 페이지

3.3 웹 페이지

4. 결론

5. 첨부

5.1 소스코드

5.2 발표 PPT

5.3 참고 문헌

1. 서론

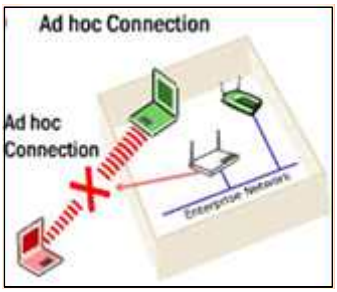
그 동안 기존의 유선랜이 제공할 수 없는 무선랜 만의 다양한 장점과 편리함에 힘입어 무선 랜 사용 범위가 작게는 가정집에서 크게는 거대한 대기업까지 점차 이용자들이 빠른 속도로 증가 되고 있다. 국내 대기업이나 금융권 유선네트워크의 경우 방화벽, 유선IPS 등 각종 보안솔루션을 다중으로 설치하여 각종 네트워크 해킹 위협에 대비하고 있다고 한다. 하지만 문제는 이들 네트워크에 "무선"이라는 단어가 추가 되면서 시작된다.
기존에는 반드시 유선 방화벽 등 다양한 겹겹의 보안 감시를 무사히 통과해야만 데이터가 정상적으로 이동을 할 수 있었지만 이제는 데이터가 무선을 통해 관리자가 모르게 어디론가 유출될 수 있다는 것이다.
무선랜의 급속한 확산으로 인해 비즈니스 상의 이점과 사용자의 편의성이 증가하면서 다양한 무선랜 장비가 많은 사람들에게 깊숙이 사용되고 있다. 이로 인해 많은 사용자들이 인식하지 못한 상태로 손쉽게 무선 보안의 위협 위에 노출되는 경우가 번번하게 발생하고 있다. 다양한 무선랜의 위협들은 무선랜 혹은 장비들을 악용하여 기밀 자료 유출 혹은 개인정보 유출 및 서비스 가용성 문제로 연결 될 수 있다.
이러한 이유로 무선 공격에 대한 탐지 및 분석을 통해 실시간으로 감지하는 WIPS 시스템을 개발하여 안전한 무선 네트워크 환경을 구축하는 것을 주제로 선정하였다.

2. 관련연구

2.1 WIPS(Wireless Intrusion Prevention System)

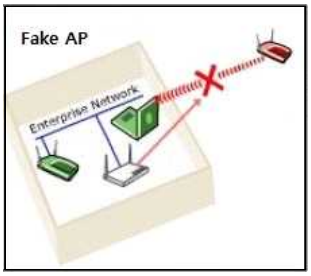
WIPS는 인가되지 않은 무선단말기의 접속을 차단하고 보안에 취약한 AP를 탐지하는 솔루션이다. 즉 유선 방화벽과 유사하게 외부 공격으로부터 내부 시스템을 보호하기 위해 무선랜 환경에서의 보안 위협을 탐지하고 대응하는 시스템이라고 할 수 있다.

2.2 Ad-Hoc 네트워크



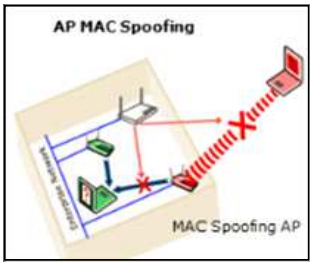
Ad-Hoc 네트워크는 P2P(Peer To Peer)라고도 불리며 고정된 유선망을 가지지 않고 이동호스트로만 이루어져 통신되는 네트워크 망이다.
따라서 유선망을 구성하기 어렵거나 망을 구성한 후 단기간 사용되는 경우에 적합하며 호스트의 이동에 제약이 없고 유선망과 기지국(Base Station)이 필요 없으므로 빠른 망 구성과 저렴한 비용의 장점이 있다.
AP와 단말이 통신하는 형태가 아니라, 각 단말끼리 연결이 되는 형태로 멀리 떨어져 있는 단말끼리는 중간에 있는 단말들이 중계기 역할을 해 주어서 중앙 시스템의 도움 없이 언제, 어디서나 기기간 통신을 가능하게 해준다.

2.3 Fake AP



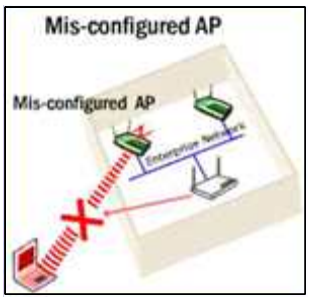
공격자가 기존에 사용자가 사용하던 AP의 wifi 이름과 동일하게 AP를 만들어 사용자가 공격자의 AP인지 모르고 공격자의 AP를 사용할 때 정보를 탈취하는 공격기법이다.

2.3-1 AP MAC Spoofing



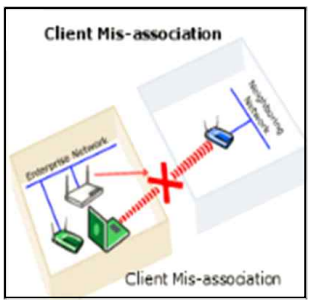
공격자가 기존에 사용자가 사용하던 AP의 Wi-fi 이름과 MAC주소를 동일하게 해서 사용자가 이용하던 AP의 연결을 끊는 신호를 보내고 공격자의 AP를 사용하게 하여 정보를 탈취하는 공격기법이다.

2.4 Mis-Configured AP



무선랜 AP는 다양한 종류의 기본적인 보안기능 및 구성 방법을 제공하고 있으나 암호화 미 적용, WEP과 같이 암호화 기능이 취약한 보안 설정, 보안 관리자의 실수, 무선 공격자의 해킹 등 으로 인해서 보안구성이 잘못된 AP를 말하며, 보안 정책에 위반되는 보안설정의 경우 외부의 비인가 무선 단말이 임의로 접속하여 내부 망에 침입할 수 있는 통로를 제공하고 기업 전체의네트워크가 무선 망으로부터 심각한 위협을 받게 된다.

2.5 Client Mis-Association



Client Mis-Association은 내부 사용자와 외부 AP 사이에 통신하는 것이다. 인가된 클라이언트가 허락없이 외부의 비인가 AP에 연결하는 경우 회사 주위에서 탐지되는 AP들은 보안설정이 없는 open인 경우가 대부분이다. 사내의 인가된 클라이언트들이 무선상으로 이들 외부 AP에 연결된 경우, 회사 내부의 중요한 자료유출이 가능하고 또한 공식적으로 무선랜을 전혀 사용하지 않는 환경에서도 대부분의 노트북에 무선랜이 장착되어 있으며 사내 AP가 없을 경우 자동으로 외부 AP에 접속하는 등 심각한 보안 위협이 발생한다.

2.6 Django

Django는 파이썬으로 만들어진 오픈소스 웹 어플리케이션 프레임워크이며, 웹 개발에서 번거로운 요소들을 새로 개발할 필요 없이 내장된 기능을 이용해 빠른 개발을 할 수 있다는 것이 장점이다. Django는 파이썬으로 코딩한 모델을 관계형 데이터베이스로 구축해주는

Model, HTTP 요청을 처리하는 웹 템플릿 시스템인 View, URL의 라우팅을 처리하는 URL 컨트롤러로 구성된 MVC 디자인 패턴을 따른다. 장고는 파이썬 프레임워크의 일반적인 URL 방식을 채택하여 다른 프레임워크에서도 사용할 수 있다. 또한 URL형태를 개발자가 직접 결정할 수 있고, 각 URL형태를 파이썬 함수에 직접 연결하도록 되어 있어서 개발이 편리하고 이해하기 쉽다. 또한, 사용자 관리와, 각각의 모델 객체에 대한 기능이 관리자 인터페이스에서 모두 제공된다. 따라서 데이터베이스 모델링만으로 웹 어플리케이션의 작동을 실험해 볼 수 있다.

2.7 **WEP** (Wired Equivalent Privacy)

유선 동등 프라이버시(영어: Wired Equivalent Privacy, WEP)는 무선 랜 표준을 정의하는 IEEE 802.11 규약의 일부분으로 무선 LAN 운용간의 보안을 위해 사용되는 알고리즘이다. 2001년 초, 암호학자들이 몇 가지 치명적인 취약점을 발견하였으며, 이를 이용하면 누구나 구할 수 있는 소프트웨어를 사용해 몇 십 분만에 WEP 연결을 크랙할 수 있다. 2004년 발표된 802.11i 표준에서 IEEE는 WEP-40 및 WEP-104을 모두 사용중단(deprecated) 선언했다.

2.8 **WPA, WPA2** (Wi-Fi Protected Access)

와이파이 보호 접속(Wi-Fi Protected Access, WPA , WPA2)는 와이파이 얼라이언스의 감시 하에 수행하는 인증 프로그램으로, 와이파이 얼라이언스가 책정한 보안 프로토콜 네트워크 장비가 준수하고 있음을 나타내는 보안 프로토콜이다. WPA 프로토콜은 이전의 유선 동등 프라이버시(WEP)의 취약점 때문에 그 대안으로 나온 것이다.

2.9 **TKIP** (Temporal Key Integrity Protocol, TKIP)

IEEE 802.11의 무선 네트워킹 표준으로 사용되는 보안 프로토콜이다. TKIP는 IEEE 802.11i의 작업 그룹과 와이파이 얼라이언스에서 WEP를 하드웨어의 교체 없이 대체하기 위해 고안되었다. 이것은 WEP의 취약점으로 인해서 와이파이 네트워크간의 LAN과 같은 보안을 제공하지 못한 채 방치된 것과, 이미 많이 사용되고 있는 하드웨어에 대한 대안으로서 필수적이었다.

2.10 **CCMP**(CTR with CBC-MAC Protocol)

WPA 방식도 TKIP 알고리즘을 이용하면 여전히 취약한데, 그래서 생겨난 방식이 AES 알고리즘과 기존 WPA 방식을 강화한 WPA2 암호화 방식이다. WPA2는 TKIP를 버리고 AES 기반 CCMP를 기본으로 사용한다.

2.11 IEEE 802.1X

IEEE 802.1X는 포트 기반 네트워크 접근 제어(PNAC)에 대한 IEEE 표준이다. 이것은 네트워크 프로토콜에 대한 그룹인 IEEE 802.1의 일부이다. 이 표준은 근거리 통신망과 무선랜을 연결하기 위한 장치의 인증 매커니즘을 제공한다.

2.12 PSK(Pre-shared Key)

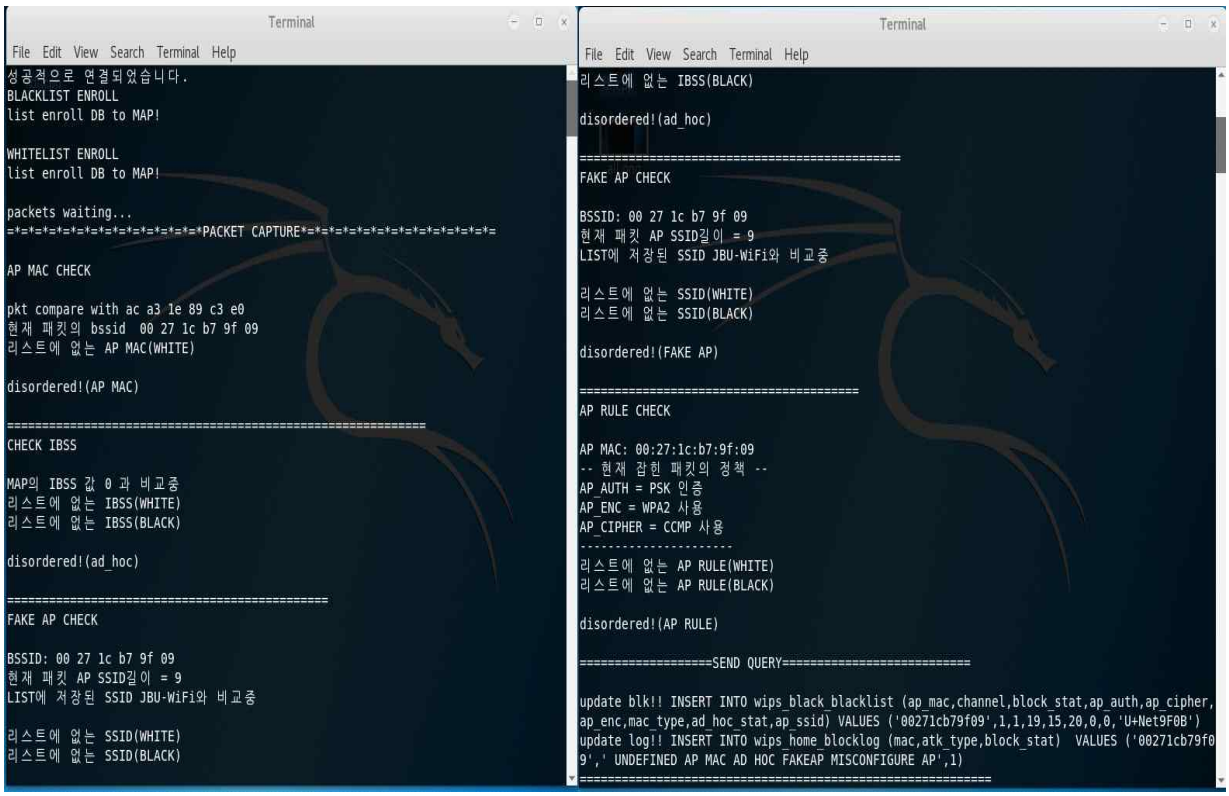
PSK 인증방식은 인증 서버가 설치되지 않은 소규모 망에서 사용되는 방식으로, 무선 AP는 무선 단말기가 자신과 동일한 비밀키(PSK)를 가지고 있는지802.1x에 규정된 EAPoL-Key 프레임을 활용하여 4웨이 핸드셰이킹 절차를 통해 확인하여 인증을 수행한다.

2.13 Deauthentication 패킷

무선연결을 차단하는 패킷으로 허가 받지 않는 Station에 특정 값을 보내어 접속을 차단.

3. 본론

3.1 구동



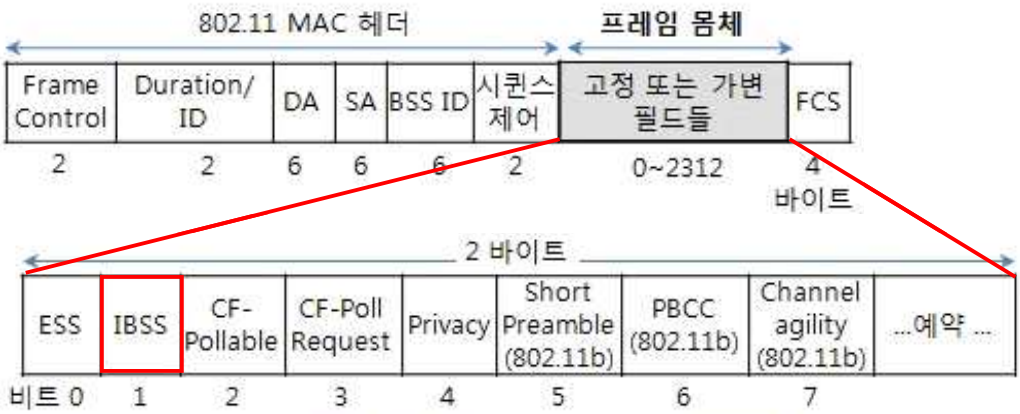
[그림1. 전체실행 화면]

Ad-hoc 네트워크

```
CHECK IBSS

cmp ibss with 0
LISTED IBSS!(WHITE)
BSSID: ac a3 1e 8a 38 41
cur leng is 8
```

[그림2 Ad-hoc 판별]

```
void usrfunc::adhocFunc(listload& listMan2)
{
        WHTFlag = false; //true in whitelisted packet
        BLKFlag = false; //true in blacklisted packetd
//wht list!
        printf("CHECK IBSS ₩n ₩n");
        for(it = listMan2.WhiteList.begin();it !=listMan2.WhiteList.end();it++) {
                bwDatas = (listload::bwList)it->second;
                 macCmpFlag = memcmp(&(bwDatas.apMac),&(MN->BSS),6);
                printf("MAP의 IBSS 값 %d 과 비교중₩n",bwDatas.adHocStat);
                If(macCmpFlag == 0) {
                        cmpFlag = memcmp(&(bwDatas.adHocStat),&(IBSS_Status),1);
                if(cmpFlag == 0) {
                        printf("이미 저장된 IBSS!(WHITE) ₩n");
                        WHTFlag = true;
                        break;
                                                        ~ 이하 생략 ~
```



[그림2.1 Ad-hoc 판별]

프레임 몸체의 IBSS 값이 1일 경우 Ad-Hoc 네트워크로 판별

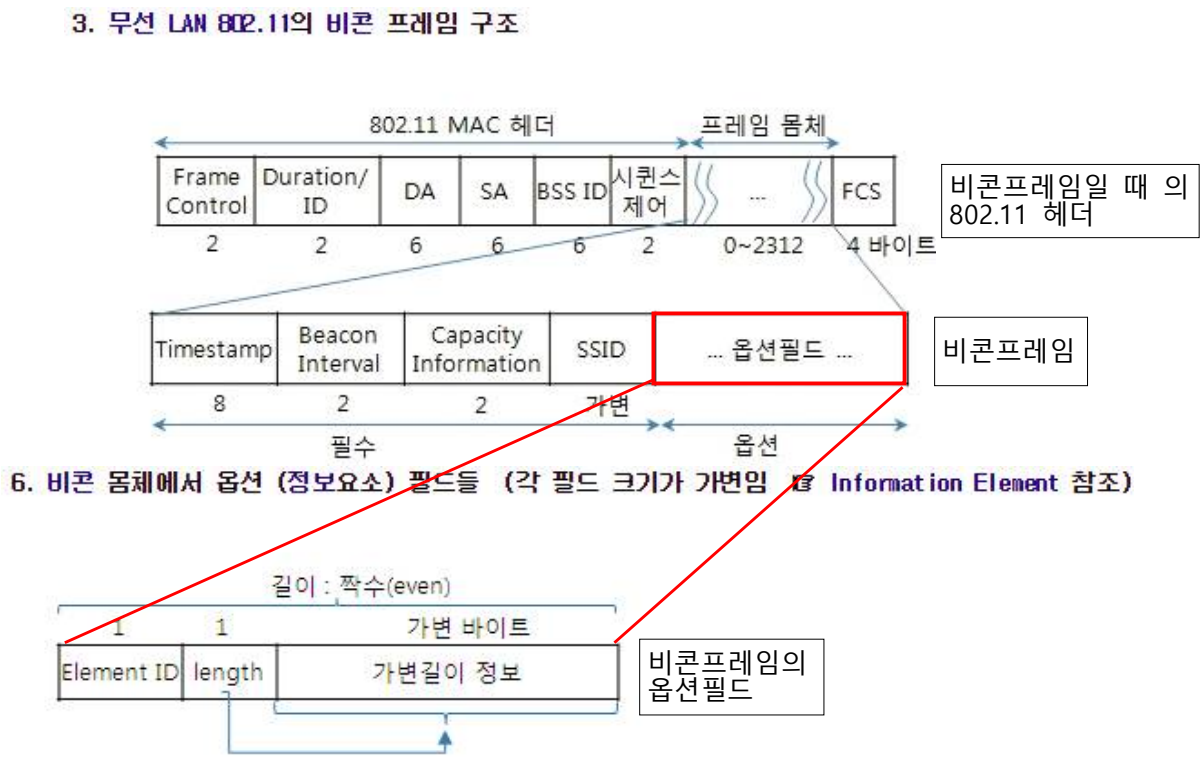
Misconfigure AP(보안정책 위반)

[그림3 AP 암호 정책]

```
int usrfunc::misconfigureAP(listload& listMan2)
{
                    ~ 중략 ~
 //********************Flag set******************
        if(sM.oUI[0] == 0x00 && sM.oUI[1] == 0x50 && sM.oUI[2] == 0xf2 &&
        sM.gCSS[0] == 0x00 && sM.gCSS[1] == 0x0f && sM.gCSS[2] == 0xac)//OUI
00-50-f2 &&     OUI 00-0f-ac -> WPA2
        {
                //a = 20; //WPA-2 flsg:20
                sF.enc = WPA2;
                sF.groupCipher = Cipher(sM.gCSS[3]);
                sF.pairwiseCipher = Cipher(sM.pCSS[3]);
                sF.auth = Auth(sM.aSS[3]);
        }
        else if(sM.oUI[0] == 0x00 && sM.oUI[1] == 0x50 && sM.oUI[2] ==
0xf2)//OUI 00-50-f2       -> WPA-1
                                                        ~ 이하 생략 ~
```

프레임구조

IEEE 802.11 -> 관리프레임 -> 비콘프레임 -> 옵션필드 -> RSN, Vendor
-> 프레임 콘트롤 -> protected frame



[그림4 프레임 구조]

RSN(Robust Security Network)

AP의 보안 능력(Security Capability)을 무선단말에게 알려주기 위함
(순서 21) RSN IE ID : 48
Element ID (1 바이트) : 48 => (RSN IE를 말함)
Length (1 바이트) : 바로 다음 Version 필드부터 끝까지의 길이
Version (2 바이트) : 현재 2
802.11i 규정에 따른 802.1X 키 관리 기능을 수행하고, CCMP 암호화도 가능함을 의미
Group Cipher Suite Selector (4 바이트)
- AP와 결합된 모든 무선단말에게 보내는 브로드캐스트,멀티캐스트 프레임을 암호화하기 위한 그룹 키(Group Key)에 대한 OUI(벤더 ID) 및 Cipher(암호 ID) 선택자로 구성됨
- WEP, TKIP, CCMP 등의 선택
Authentication and Key Management Protocol (AKMP)
- 인증 및 키관리 프로토콜

※ Cipher Suite Selector 형식

3 바이트	1 바이트
OUI	Suite type

Suite 종류	OUI	Suite type	의미
Cipher Suite	00:0F:AC (IEEE 802.11)	0	Group Cipher Suite 사용
		1	WEP-40
		2	TKIP
		3	(예약)
		4	CCMP
		5	WEP-104
	기타 Vendor OUI	x	Vendor specific
AKM Suite	00:0F:AC (IEEE 802.11)	0	(예약)
		1	802.1X 인증 (대규모)
		2	PSK 인증 (개인,SOHO)
		3~255	· · ·
	기타 Vendor OUI	x	Vendor specific

Element ID = 48 (RSN IE)	Length	Ver.	Group Cipher Suite Selector	Pairwise Cipher Suite Count	Pairwise Cipher Suite Selector	AKM Suite Count	AKM Suite Selector	RSN Capabilities	PMK Count	PMK list
1	1	2	4	2	4 x 가변	2	4 x 가변	2	2	16 x 가변

[그림5 RSN 구조 및 정보]

구분		인증 방식	암호화 방식	비고
개방 인증	SSID 숨김	-	-	보안은 아니나, 현실적으로 사용
	MAC 인증	-	-	보안은 아니나, 현실적으로 많이 사용
공유키	WEP	-	RCA	보안 취약으로 비추천
인증,암호 모두 사용	WPA	802.1x/다양한 EAP	TKIP	RSN IE (ID 48) Vendor-specific element (ID 221)
	802.11i	802.1x/다양한 EAP		
	WPA2	802.1x/다양한 EAP	AES	

[그림6 주요 암호방식 비교]

Vendor specific

필드는 IEEE 802.11 표준에서 정의하지 않는 판매자-특정 정보를 위해서 사용될 수 있다.
Element ID (1 바이트) : 221 => (Vendor specific)
length (1 바이트) : 바로 다음필드부터 끝까지의 길이
OUI (3 바이트) :
Vendor specific type(1 바이트) :
DATA(가변)

Element ID	Length	OUI 3byt	Vender Specific OUI Type	WPA Version	Multicast Cipher Suite OUI
Multicast Cipher Suite Type			Unicast Cipher Suite Count	Unicast Cipher Suite OUI(3) / Type(1)	
Auth Key Management Suite Count		Auth Key Management OUI(3) / Type(1)			

[그림7 Vendor specific 구조]

WPA 정보 요소는 다음과 같습니다
WPA 요소 ID는 0x221로 설정됩니다. WPA 요소 ID는 공급 업체 특정 요소 ID와 동일합니다. 따라서 벤더 고유의 요소 ID가 수신 될 때마다 Station / AP가 정보 요소가 WPA인지 확인하기 위해 OUI를 검사해야합니다.
WPA OUI (Organizationally Unique Identifier)는 00-50-f2로 설정됩니다.

OUI	스위트 룸 유형	의미
00-50-f2	0	그룹 암호 모음 사용
00-50-f2	1	WEP-40
00-50-f2	2	TKIP
00-50-f2	3	예약섭
00-50-f2	4	CCMP(예약섭??)
00-50-f2	5	WEP-104

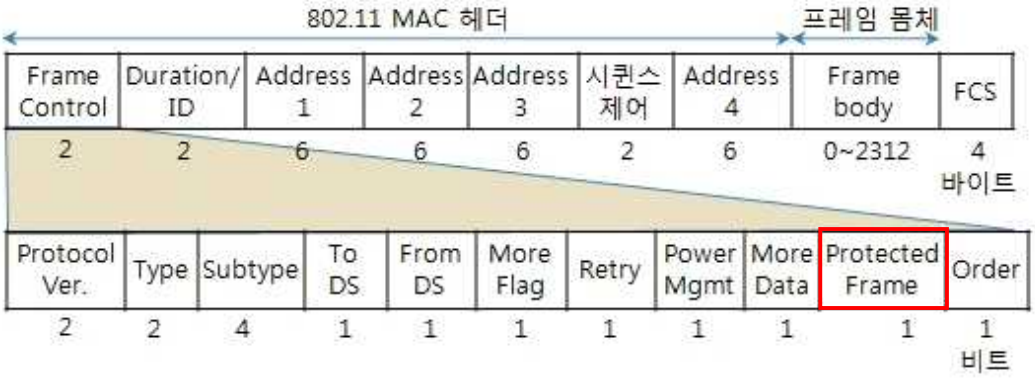
[그림8 Suite OUI 및 Type]

TKIP는 WPA의 기본 암호 모음입니다.
참고 : WEP-40 및 WEP-104는 TSN (Transition Station Network)의 그룹 암호 그룹으로 만 사용할 수 있습니다.
AKM 카운트 - 인증 키 관리 카운트는 지원되는 인증 키 관리 스위트의 수를 제공합니다.
AKM 목록 - 지원되는 다양한 인증 메커니즘의 수입니다. 표의 열은 암호 모음 및 유형 조합 중 일부를 나타냅니다.

OUI	스위트 룸 유형	의미
00-50-f2	0	예약섭
00-50-f2	1	802.1X
00-50-f2	2	PSK

[그림9 Auth OUI 및 Type]

Protected Frame



[그림10 Protected Frame 구조]

Protected Frame 필드가 0 이면 OPEN 1이면 WEP 설정으로 탐지합니다.

FLAG 세팅

-	ENC	CIPHER		AUTH	
0	OPN (OPEN)	-		-	
1	WEP	-		-	
10	WPA	11	그룹암호 모음사용(0)	17	예약석(0)
		12	WEP-40(1)		
		13	TKIP (기본)(2)	18	802.1X(1)
20	WPA2	14	예약석(3)		
		15	CCMP(4)	19	PSK(2)
		16	WEP-104(5)		

[그림11 설정한 FLAG 값]

위의 RSN, Vendor specific의 정보요소 암호 프로토콜 인증방식, 암호알고리즘 FLAG값으로 세팅하여 탐지합니다.

Client Mis-Association(비인가 AP 접속)

```
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*PACKET CAPTURE*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=

AP MAC CHECK

pkt compare with ac a3 1e 89 c3 e0
현재 패킷의 bssid  70 25 59 16 9e 20
리스트에 없는 AP MAC(WHITE)
cmp with 00 27 1c b7 9f 09
cmp with 70 25 59 16 9e 20
이미 저장된 AP MAC!(BLACK)
==============================================================
```

[그림12 AP MAC 체크]

```
void usrfunc::macCmp(listload& listMan2)
{
        WHTFlag = false; //true in whitelisted packet
        BLKFlag = false; //true in blacklisted packet
        RadiotapHeader *RH = (RadiotapHeader*)(pktPoint);
        int length = RH->length;
        ManagementFrame *MF = (ManagementFrame*)(pktPoint+length);
        int cmpFlag = 0;
        listload::bw_list::iterator it;
         printf("AP MAC CHECK ₩n ₩n");
//wht list!
        for(it = listMan2.WhiteList.begin();it !=listMan2.WhiteList.end();it++) {
                bwDatas = (listload::bwList)it->second;
                printf("pkt compare with %02x %02x %02x %02x %02x %02x
₩n",bwDatas.a          pMac[0],b
wDatas.apMac[1],bwDatas.apMac[2],bwDatas.apMac[3],bwDat
as.apMac[4],bwDatas.     apMac[5]);
                                        ~ 이하 생략  ~
```

[그림13 802.11 MAC 프레임]

Cient Mis-Association의 탐지는 Data Packet의 주소1, 주소2, 주소3을 분석하여 외부 AP와 인가된 사용자의 MAC이 사용되는 것을 탐지합니다.

Fake AP Check

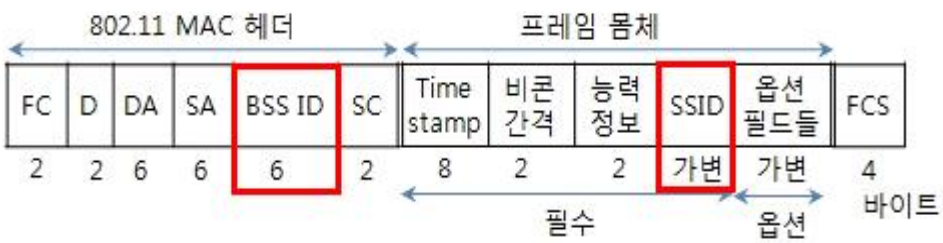
```
==============================================
FAKE AP CHECK

BSSID: 18 c5 01 10 d3 86
현재 패킷 AP SSID길이 = 9
LIST에 저장된 SSID JBU-WiFi와 비교중

리스트에 없는 SSID(WHITE)
cmp ssid U+NetD387
이미 저장된 SSID(BLACK)
```

[그림14 FAKE AP 체크]

```
void usrfunc::fakeAp(listload& listMan2)
{
        WHTFlag = false; //true in whitelisted packet
        BLKFlag = false; //true in blacklisted packet
        uint8_t* pktPoint2 = this->pktPoint
        printf("==============================================
₩n");
        printf("FAKE AP CHECK ₩n ₩n");
        struct packframes::WifiName *wifiName
//BSSID*************************************************************************//
        printf("BSSID: ");
        for(int i=0; i<6; i++)
        {
                printf("%02x ", mgmtFrame->addr3[i]);
        }
        printf(" ₩n");
//SSID**************************************************************************//
        pktPoint2 += (RTHeader->rth_length + sizeof(struct
packframes::ManagementFrame)                                  ~ 이하 생략 ~
```

o 비콘 프레임 (subtype : 1000)

- 시간 동기 및 네트워크 존재를 알리는 목적의 프레임



o (순서 4) SSID (Service Set Identifier) (ID : 0)

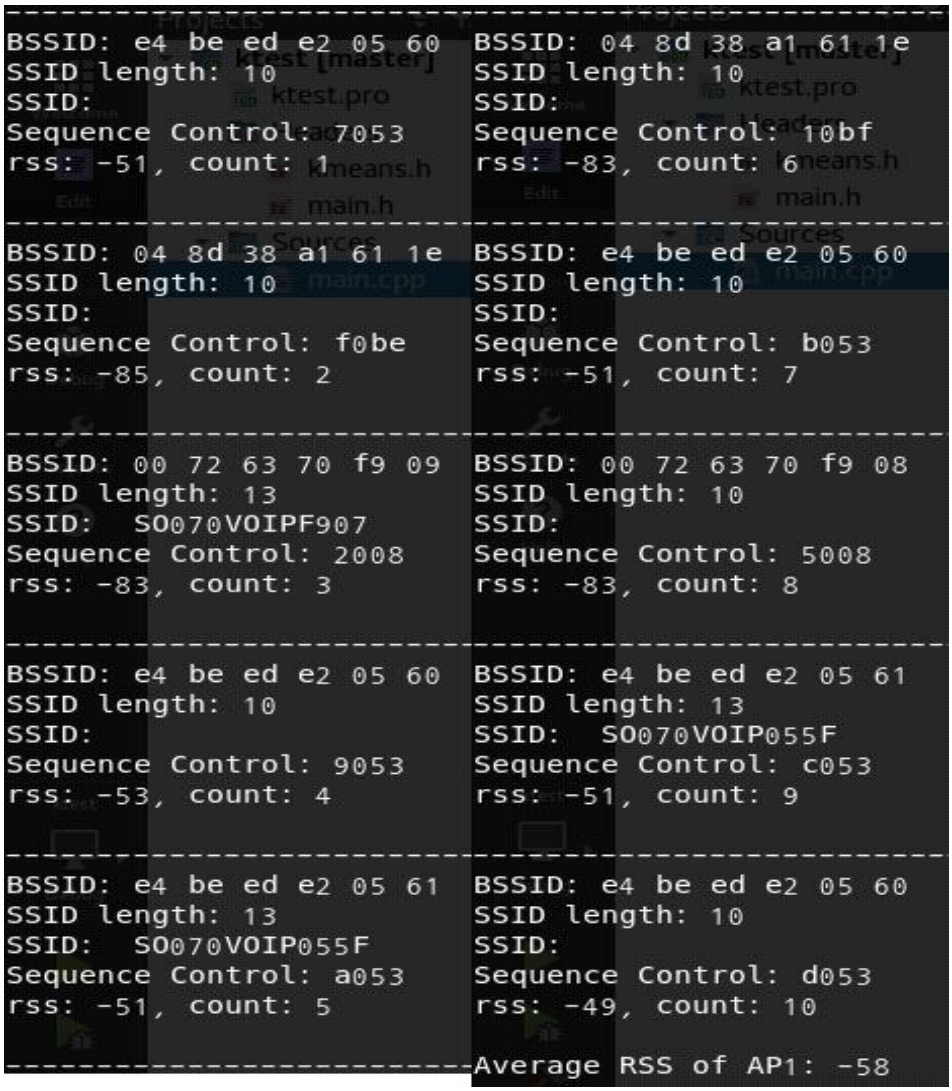
- 여러 AP들을 그룹화시킨 단일 관할영역(ESS,확장서비스셋)의 서비스 제공자 명칭

1	1	0 ~ 32 바이트
Element ID	length	SSID 문자열

[그림15 BSS ID 및 SSID]

Fake AP 공격은 캡처한 패킷의 SSID와 일치하는 화이트리스트에 있는 AP의 SSID가 있는지 확인한다.
동일한 SSID가 존재하는 경우, 캡처한 패킷의 BSS ID와 해당 AP의 MAC 주소를 비교한다.
캡처한 패킷의 BSSID와 AP의 MAC 주소가 일치하지 않을 때 Fake AP 공격이 발생하였다고 판단한다.

AP MAC Spoofing (모듈)



[그림16 AP MAC Spoofing 공격 탐지를 위해 필요한 모듈]

```
#include "main.h"
#include "kmeans.h"
            ~ 중략 ~
//BSSID*********************************************************************************//
        printf("-------------------------------------------------------------₩n");
        printf("BSSID: ");
        for(int i=0; i<6; i++)
        {
            printf("%02x ", mgmtFrame->addr3[i]);
        }
        printf("₩n");
                                        ~ 이하 생략
```

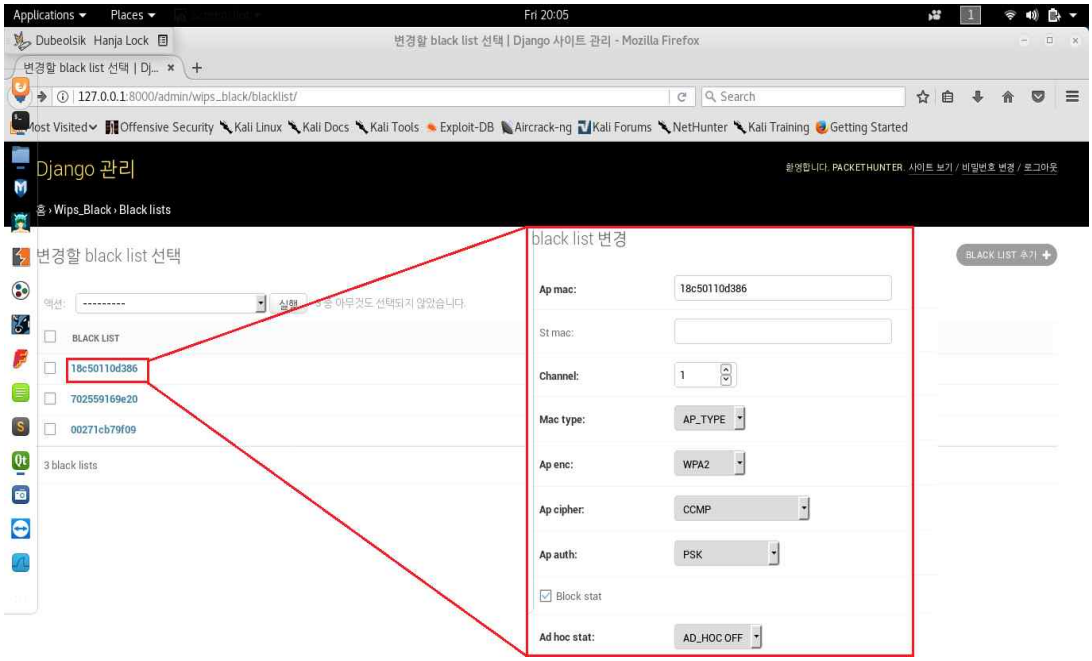
AP MAC 위장(AP MAC Spoofing) 공격은 캡처한 패킷의 SSID와 일치하는 화이트리스트에 있는 AP의 SSID가 있는지 확인한다.
동일한 SSID가 존재하는 경우, 캡처된 패킷의 BSS ID와 해당 AP의 MAC 주소를 비교한다.
위의 두 항목이 모두 일치한다면 화이트리스트에 저장되어있는 AP의 순차 번호(Sequence number)와 캡처한 패킷의 순차 번호(Sequencenumber)를 비교한다.
순차 번호(Sequence number)가 순차적으로 증가하는 경우 화이트리스트의
순차번호(Sequence number)를 업데이트 한다.
순차적이지 않을 경우에는 신호세기(rss)를 조사하여 화이트리스트에 있는 신호세기의
허용된 오차범위에 포함되는지 비교한다.
포함되지 않는 경우에는 AP MAC 위장(AP MAC Spoofing) 공격이 발생하였다고 판단한다.

STL::MAP 메모리 관리

```
                                ~ 중략 ~
        rowNum = (int)strtol(row[0],NULL,10);

        convMac(0,row[1]);
        bwStruct.channel = (int)strtol(row[2],NULL,10);
        bwStruct.blockStat = (int)strtol(row[3],NULL,10);
        convMac(1,row[4]);
        bwStruct.apAuth = (int)strtol(row[5],NULL,10);
        bwStruct.apCipher = (int)strtol(row[6],NULL,10);
        bwStruct.apEnc = (int)strtol(row[7],NULL,10);
        bwStruct.macType = (int)strtol(row[8],NULL,10);
        bwStruct.adHocStat = (uint8_t)strtol(row[9],NULL,10);

        sprintf(bwStruct.ssid,"%s",row[10]);
        printf("ssid is %s₩n",row[10]);
        if(bwStruct.macType == 0){
            memset(&(bwStruct.stMac[0]),0,6);
                                                ~ 이하 생략 ~
```

데이터 값과 정렬 키를 갖고 있는 Pair을 이용하여 저장, 검색하는 MAP입니다.
고유 키를 이용해 메모리 상에 정리하게 되는데, DB데이터 및 패킷을 캡처해 메모리를 갱신하며 해당 데이터는 등록한 정책과 비교하여 차단/허용 하는데 사용합니다.

3.2 관리자 페이지



[그림17 실행 이후 관리자 페이지]

[그림17.1 실행 이후 관리자 페이지]

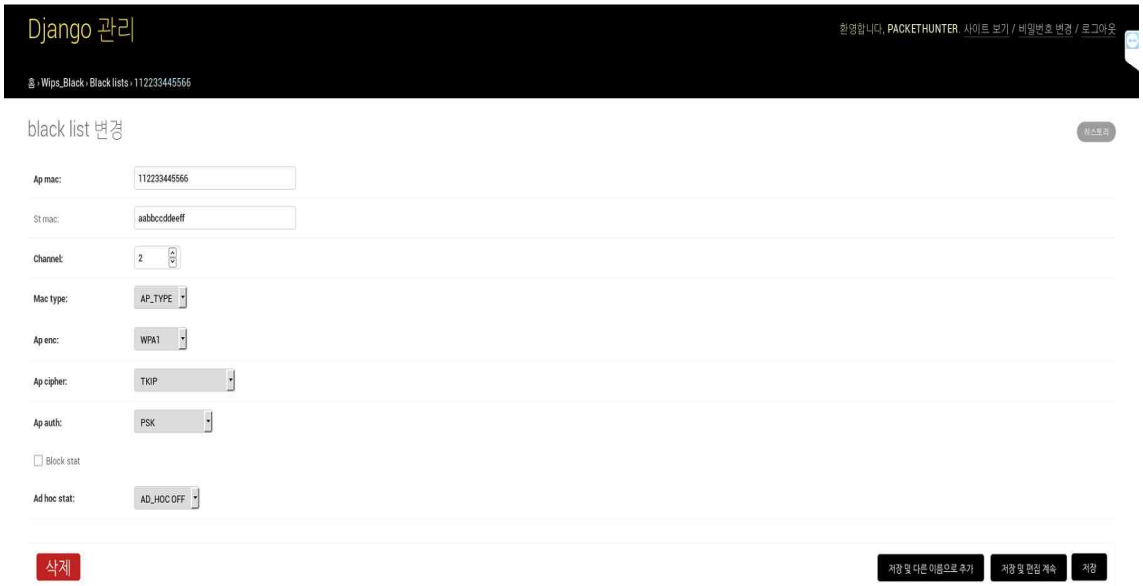
관리자 페이지로 접속하면 설정한 데이터베이스의 값을 조회하고 수정할 수 있는 화면을 볼 수 있다.

base.html

```
{% load i18n static %}<!DOCTYPE html>
{% get_current_language as LANGUAGE_CODE %}{% get_current_language_bidi as
LANGUAGE_BIDI %}
<html lang="{{ LANGUAGE_CODE|default:"en-us" }}" {% if LANGUAGE_BIDI
%}dir="rtl"{% endif %}>
<head>
<title>{% block title %}{% endblock %}</title>
<link rel="stylesheet" type="text/css" href="{% block stylesheet %}{% static
"admin/css/base.css" %}{% endblock %}">
{% block extrastyle %}{% endblock %}
{% if LANGUAGE_BIDI %}<link rel="stylesheet" type="text/css" href="{% block
stylesheet_rtl %}{% static "admin/css/rtl.css" %}{% endblock %}">{% endif %}
                                        ~ 이하 생략 ~
```

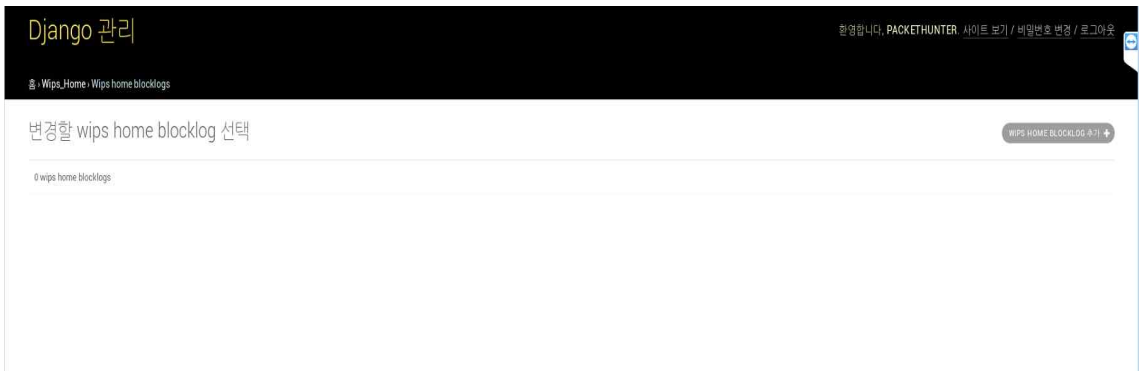
base.css

```
/*
    DJANGO Admin styles
*/
@import url(fonts.css);
body {
    margin: 0;
    padding: 0;
    font-size: 23px;
    font-family: "Roboto","Lucida Grande","DejaVu Sans","Bitstream Vera
Sans",Verdana,Arial,sans-serif;
    color: #333;
    background: #fff;
}/* LINKS */
                                        ~ 이하 생략 ~
```



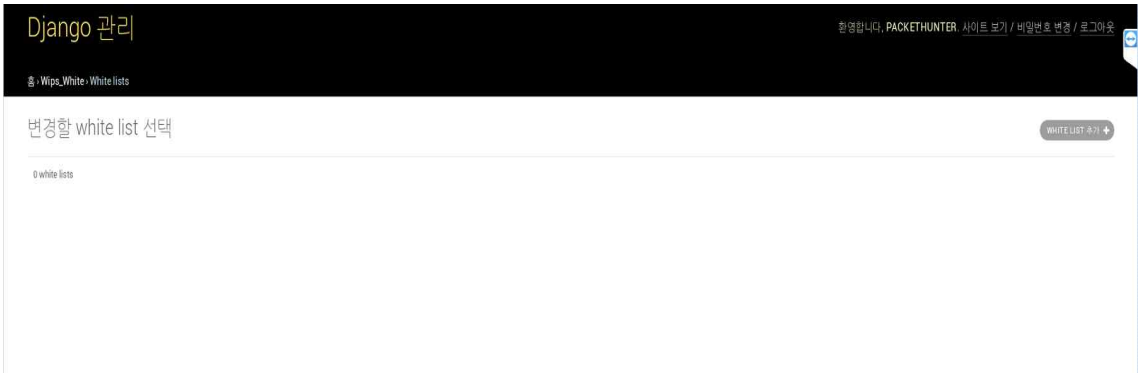
[그림18 Black list 관리자 페이지 화면]

Black list 데이터베이스로 접속하면 black list값을 변경할 수 있는 페이지로 넘어갈 수 있다.



[그림19 Block log 관리자 페이지 화면]

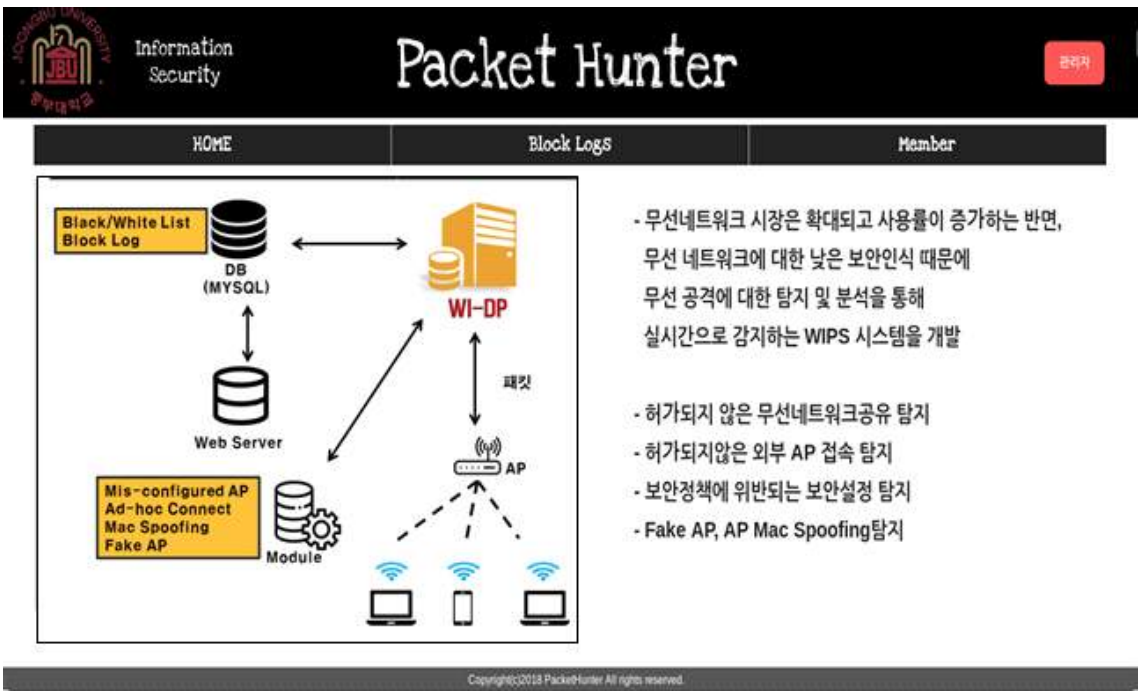
Block log 데이터베이스로 접속하면 Block log를 조회, 수정할 수 있는 페이지로 이동할 수 있다.



[그림20 White list 관리자 페이지 화면]

white list 데이터베이스로 접속하면 whitelist를 조회, 수정할 수 있는 페이지로 이동할 수 있다.

3.3 웹 페이지



[그림21 메인 웹페이지]

웹페이지에 접속하면 시스템 구상도와 함께 주제선정, 개발한 시스템에 대한 설명이 있다.

index.html

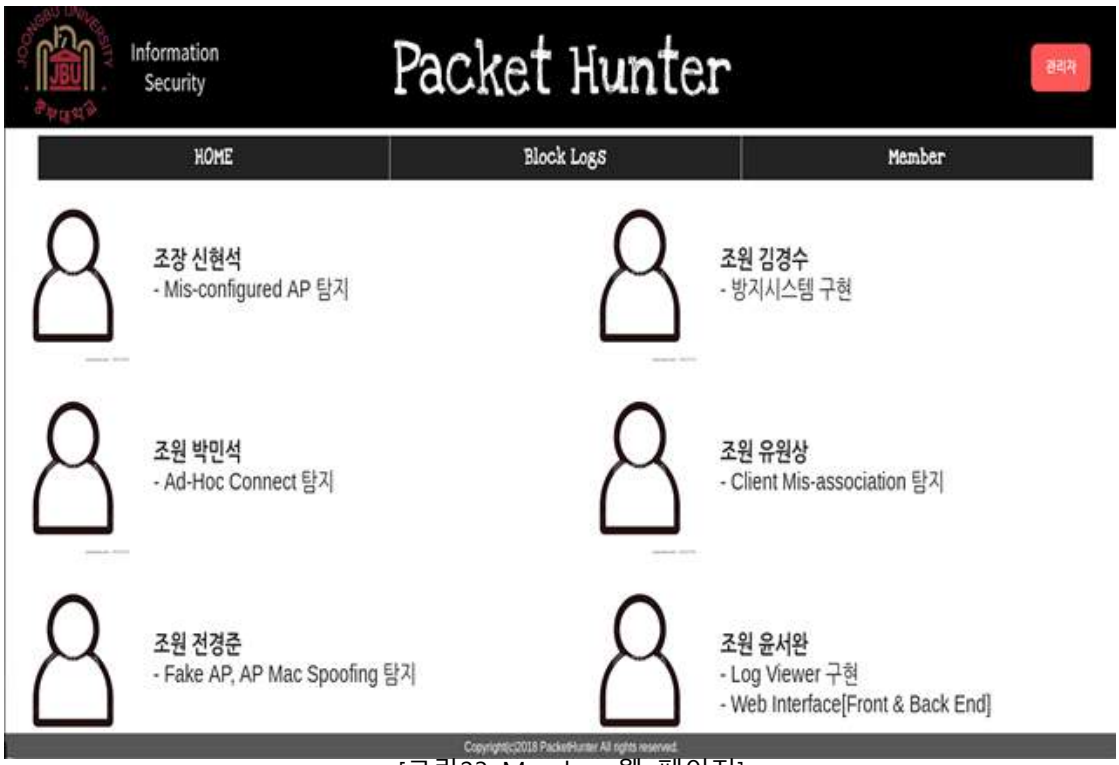
```
{% extends 'base.html' %}
{% load static %}
{% block content %}
<style type="text/css">
h3 {
    font-size: 80px;
    font-family: "Courier New";
    color: white;
    background-color: black;
    position: absolute;
    height: 150px;
    width:100%;
    z-index: 1;
    text-align: center;
    margin-bottom: 10px;
    padding: 30px;
    left: -15px;
    top: 1px;
                    ~ 이하 생략 ~
```

[그림22 Block logs 웹페이지]

Block Logs탭에서는 관리자페이지에서 사용자의 설정에 의해서 데이터베이스에 저장된 무선공격에 대한 차단된 패킷들의 정보들을 확인할 수 있다.

block_log.html

```
{% extends 'base.html' %}
{% load static %}
{% block content %}
<style type="text/css">
h3 {
    font-size: 80px;
    font-family: "Courier New";
    color: white;
    background-color: black;
    position: absolute;
    height: 150px;
    width:100%;
    z-index: 1;
    text-align: center;
    margin-bottom: 10px;
    padding: 30px;
    left: -15px;
    top: 1px;
                    ~ 이하 생략 ~
```



[그림23 Member 웹 페이지]

member.html

```
{% extends 'base.html' %}
@import url(http://fonts.googleapis.com/earlyaccess/jejuhallasan.css)
{% load static %}
{% block content %}
<style type="text/css">
h3 {
    font-size: 80px;
    font-family: "Courier New";
    color: white;
    background-color: black;
    position: absolute;
    height: 150px;
    width: 100%;
    z-index: 1;
    text-align: center;
    margin-bottom: 10px;
    padding: 30px;
    left: -15px;
    top: 1px;
                    ~ 이하 생략 ~
```

4. 결론

포괄적인 방어를 수행하기 위해서는 다양하고 위협적인 무선랜위협을 완화시켜야 한다. 그러나 모든 보안 유지 수단은 위험이 발생하기 전에 즉, 위험을 관리할 수 있을 때 도입하는 것이 가장 효과적이다.
위장 액세스 포인트 공격은 간단한 MAC 변조를 통하여 손쉽게 설치가 가능하고 내부 네트워크에 바로 들려 있어서 공격자는 방화벽이나 IDS와 같은 보안 장비를 우회하여 내부 네트워크에 침투할 수 있게 된다. 그래서 우리 팀은 무선 구간의 실시간 모니터링을 통하여 정보를 수집하고, 수집한 정보를 분석하여 무선랜의 위협을 사전에 탐지를 차단하는 시스템을 개발 하였다.
무선 구간 모니터링 에이전트는 위장 액세스 포인트뿐만 아니라 무선 네트워크에 속해 있는 노드들의 정보를 파악할 수 있고, 그 상태를 분석하여 공격을 미연에 방지 할 수 있다.

5. 첨부

5.1 소스코드

[aaa.h] <list update with map,userfunc>

```
#include <stdio.h>
#include <stdlib.h>
#include <pcap.h> /* if this gives you an error try pcap/pcap.h */
#include <errno.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netinet/if_ether.h> /* includes net/ethernet.h */
#include <iostream>
#include <string.h>

//ENC
#define WEP 1
#define WPA1 10
#define WPA2 20
//Cipher
#define OPEN 0
#define GROUP_CIPHER_SUITE 11
#define WEP40 12
#define TKIP 13
#define C_RESERVATON 14//예약사용
#define CCMP 15
#define WEP104 16
//Auth
#define A_RESERVATON 17//예약사용
#define A_8021X 18//802.1X
#define PSK 19


using std::cout;
using std::endl;
#pragma pack(push, 1)



struct SecurityFlag
{
    int8_t enc;//Encrypt
    int8_t groupCipher; //GroupCypher
    int8_t pairwiseCipher; //PairwiseCipher
    int8_t auth; //Authentication
};
```

```
struct SecurityMethod
{
    uint8_t wep:1;//WEP
    uint8_t gCSS[4]; //Group Cipher Suite Selector gcss[3] = type; ID=48
    uint16_t pCSC; //Pairwise Cipher Suite Count
    uint8_t aSC[2]; //AKM Suite Count
    uint8_t pCSS[4]; //Pairwise Cipher Suite Selector 4x가변길이.. pcss[3] = type ID=48
    uint8_t aSS[4]; //AKM Suite Selector 4x가변길이.. akmss[3] = type ID=48
    uint8_t oUI[3]; //Organizationally unique identifier ID=221
    uint8_t mCSS[4]; //Group Cipher Suite Selector gcss[3] = type; ID=221 //필요없을듯?
    uint16_t uCSC; //Pairwise Cipher Suite Count
    uint8_t uCSS[4]; //unicast Cipher Suite Select OUI 4x가변길이 uCSS[3] = type ID=221
    uint8_t aKMC[2]; //AKM Suite Count
    uint8_t aKMS[4]; //AKM Suite Selector1 4x가변길이.. aSS[3] = type ID=221
    uint8_t vST; //Vendor Specific (OUI) Type 필요없음??
};

struct FrameCtrl //Management Frame Control
{
    uint8_t    protocolVer      : 2;
    uint8_t    type             : 2;
    uint8_t    subType          : 4;
    uint8_t    toDs             : 1;
    uint8_t    fromDs           : 1;
    uint8_t    moreFlag         : 1;
    uint8_t    retry            : 1;
    uint8_t    powerMgmt        : 1;
    uint8_t    moreData         : 1;
    uint8_t    protectedFrame : 1;
    uint8_t    order            : 1;
};

struct ManagementFrame
{
    struct FrameCtrl   frameCtrl;   //2 bytes
            uint16_t    duration;    //2 bytes
            uint8_t     addr1[6];    //6 bytes
            uint8_t     addr2[6];    //6 bytes
            uint8_t     addr3[6];    //6 bytes BSS = AP MAC
            uint16_t    seq_ctrl;    //2 bytes
};

struct RadiotapHeaderFlag
{
    uint8_t cfp:1;
    uint8_t preamble:1;
    uint8_t wep:1;
    uint8_t fragmentation:1;
    uint8_t fcs:1;
    uint8_t data_pad:1;
```

```
    uint8_t bad_fcs:1;
    uint8_t short_gi:1;
};

struct RadiotapHeader
{
        uint8_t     version;
        uint8_t     pad;
        uint16_t    length;
        uint64_t    presentFlags;
    struct RadiotapHeaderFlag     flags;
        uint8_t     dataRate;
        uint16_t    channelFrequency;
        uint16_t    channelFlags;
        uint8_t     ssiSignal_1;
        uint8_t     wtfTrash;    //strange stuff
        uint16_t    rxFlags;
        uint8_t     ssiSignal_2;
        uint8_t     antenna;
};

struct PairwiseCipherSuiteSelector //Pairwise Cipher Suite Selector 4x가변길이 <-pCSS[4];
{
    uint8_t pOUI[3]; // pOUI[0] == Rsn.pCSS[0], pOUI[1] == Rsn.pCSS[1], pOUI[2] ==
Rsn.pCSS[2]
    uint8_t pTYPE; // pTYPE == Rsn.pCSS[3]
};

struct Rsn
{
    uint8_t elementId; //Element_ID
    uint8_t length; //Length
    uint16_t version;
    uint8_t gCSS[4]; //Group Cipher Suite Selector gcss[3] = type
    uint16_t pCSC; //Pairwise Cipher Suite Count
    PairwiseCipherSuiteSelector pCSS;
    //uint8_t pCSS[4]; //Pairwise Cipher Suite Selector 4x가변길이.. pcss[3] = type
    uint8_t aSC[2]; //AKM Suite Count
    uint8_t aSS[4]; //AKM Suite Selector 4x가변길이.. aSS[3] = type
    uint16_t rsnC; //RSN Capabilities
    uint16_t pmkC; //PMK Count
    uint16_t pmkL; //PMK List
};

struct VendorSpecific
{
    uint8_t elementId; //Element_ID 221
    uint8_t length; //Length
    uint8_t oUI[3]; //Organizationally unique identifier
    uint8_t vST; //Vendor Specific (OUI) Type
```

```
    uint16_t wpaVersion; //WPA
    uint8_t mCSS[4]; //Multicast Cipher Suite Select OUI mCSS[3] = type
    uint16_t uCSC; //unicast Cipher Suite Count
    uint8_t uCSS[4]; //unicast Cipher Suite Select OUI 4x가변길이 uCSS[3] = type
    uint8_t aSC[2]; //AKM Suite Count
    uint8_t aSS[4]; //AKM Suite Selector 4x가변길이.. aSS[3] = type
};

struct OptionField
{
    uint8_t elementId; //Element_ID
    uint8_t length; //Length
    //Rsn rsn;
    //Vendor_specific vendor_specific;
    //Variable Length Optionflied 가변길이의 옵션필드
};

struct BeaconFrameBody
{
    uint64_t   timestamp;
    uint16_t   beaconInterval;
    uint16_t   capacityInformation;
};

struct AkmSuiteSelector//??
{
    //uint16_t aSC; //AKM Suite Count
    uint8_t aSS[4]; //AKM Suite Selector 4x가변길이.. aSS[3] = type
};

#pragma pack(pop)
//void misconfigureAP (const uint8_t *);
//int Cipher(uint8_t *);
//int Auth(uint8_t *);
```

[bbb.h] <combine userFunc>

```
#ifndef BBB_H
#define BBB_H
#include <sys/time.h>
#include <netinet/in.h>
#include <pcap/pcap.h>
#include <signal.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <stdint.h>
```

```
#pragma pack(push,1)

typedef struct Radiotap_Header
{
	u_int8_t Header_Revision;
	u_int8_t Header_Pad;
	u_int16_t Header_Length;

}Radiotap_Header;

typedef struct FrameControl
{
	u_int8_t ProVer:2;
	u_int8_t Type:2;
	u_int8_t Subtype:4;
	u_int8_t ToDs:1;
	u_int8_t FromDs:1;
	u_int8_t MoreFlag:1;
	u_int8_t Retry:1;
	u_int8_t PowerMgmt:1;
	u_int8_t MoreData:1;
	u_int8_t Wep:1;
	u_int8_t Rsvd:1;
 } FrameControl;

typedef struct Manage
{
	FrameControl Frame_control;
	u_short Duration;
	u_int8_t des[6];
	u_int8_t src[6];
	u_int8_t BSS[6];
	u_short Sequence;


	struct Wireless_LAN{
		u_int8_t Timestamp[8];
		u_short Beacon;
		struct Capavility{
			u_int8_t ESS:1;
			u_int8_t IBSS:1;
			u_int8_t CFP:2;
			u_int8_t Privacy:1;
			u_int8_t Short_Preamble:1;
			u_int8_t PBCC:1;
			u_int8_t Channel_Agility:1;
			u_int8_t Spectrum_Agility:1;
			u_int8_t Short_Slot:1;
			u_int8_t APSD:1;
			u_int8_t Raido_measurement:1;
```

```
                u_int8_t DSSS_OFDM:1;
                u_int8_t DBA:1;
            }Capavility;

        }Wireless_LAN;

}Manage;

#pragma pack(pop)
#endif // BBB_H
```

[clisocket.cpp] <plus ssid cmp>

```
#include "clisocket.h"

clisocket::clisocket()
{
 /*    clientSocket = socket(PF_INET,SOCKET_STREAM,0);
      if(clientSocket == -1){
            printf("server socket create fail");
      }
      serverAdr.sin_family = AF_INET;
      serverAdr.sin_port = htons(portNo);
      serverAdr.sin_addr.s_addr = htonl(INADDR_ANY);
      if(connect(clientSocket,(struct sockaddr *)&clientSocket,sizeof(clientSocket)) == -1){
            printf("can't connect₩n");
            close(clientSocket);
            return -1;
      }
      while(true){
            read(clientSocket,recvBuf,1024)
            //write
      }
*/
}
```

[clisocket.h] <plus ssid cmp>

```
#ifndef CLISOCKET_H
#define CLISOCKET_H

#include <sys/types.h>
#include <sys/socket.h>
#include <unistd.h>
#include <netinet/in.h>

class clisocket
{
private:
```

```
	int clientSocket;

	//uint8_t recvBuf[];
	char * inetAdr;
	char * portNo;
	struct sockaddr_in serverAdr;

public:
	clisocket();
};

#endif // CLISOCKET_H
```

[dbmanage.cpp] <recv string to hex>

```
#include "dbmanage.h"

dbmanage::dbmanage()
{

	mysql_init(&this->mysql);


if(!mysql_real_connect(&mysql,DBInfo.dbHost,DBInfo.dbUser,DBInfo.dbPass,DBInfo.dbName,3
306,(char *)NULL, 0))
	{
		printf("%s₩n",mysql_error(&mysql));
		exit(1);
	}
	printf("성공적으로 연결되었습니다.₩n") ;
}

dbmanage::~dbmanage()
{
	mysql_close(&mysql);
}


MYSQL_RES* dbmanage:: dbQuery(char * qStr)
{
	//qStr = "SHOW TABLES"
	if(mysql_query(&mysql, qStr))
	{
		printf("%s₩n", mysql_error(&mysql));
		exit(1);
	}
	this->res = mysql_store_result(&mysql);
	return this->res;

}
```

```
void dbmanage::dbFree(void)
{
	mysql_free_result(this->res);
}
```

[dbmanage.h] <user func update!>

```
#include <iostream>
#include <cstdio>
#include <mysql/mysql.h>

#define WHITEAP 1
#define WHITESTATION 2
#define BLACKAP 3
#define BLACKSTATION 4
#pragma once

class dbmanage
{
private:
	struct{
		char const *dbHost ="localhost";
		char const *dbUser ="packethunter";
		char const *dbPass ="packethunter";
		char const *dbName ="PacketHunter";
	}DBInfo;
	MYSQL mysql;
	MYSQL_RES *res;


public:
	dbmanage();
	~dbmanage();
	MYSQL_RES* dbQuery(char * qStr);
	void dbFree(void);

};
```

[deauth.cpp] <deauth_mutex>

```
#include "deauth.h"

deauth::deauth()
{

}

void deauth::testFunc11(listload& listMan,dbmanage& wipsDB,std::mutex& mtx_lock1)
{
	printf("deauth testFunc₩n");
```

```
    char tempbuf[250] = {0,};
    char chanbuf[250] = {0,};
    //std::mutex mtx_lock;

    listload::bw_list::iterator it;
    printf("CHECK fakeAp₩n₩n");
    while(1){
        mtx_lock1.lock();
        for(it = listMan.BlackList.begin();it !=listMan.BlackList.end();it++){
            //printf("view func first %d₩n",it->first);
            bwDatas = (listload::bwList)it->second;
            deauthInfo.channel = bwDatas.channel;
            deauthInfo.blockStat = bwDatas.blockStat;
            if(deauthInfo.blockStat == 1 ){
                memcpy(&(deauthInfo.stMac),&(bwDatas.stMac),6);
                memcpy(&(deauthInfo.apMac),&(bwDatas.apMac),6);
            }else if(deauthInfo.blockStat == 0){
                memcpy(&(deauthInfo.apMac),&(bwDatas.apMac),6);
                memset(&(deauthInfo.stMac),0,6);

            }
            sprintf(chanbuf,"sudo iwconfig wlan3 channel %d",deauthInfo.channel);
            int ret1 = system(chanbuf);
            sprintf(tempbuf,"aireplay-ng -0 20 -a %02X:%02X:%02X:%02X:%02X:%02X
wlan3",deauthInfo.apMac[0],deauthInfo.apMac[1],deauthInfo.apMac[2],deauthInfo.apMac[3],d
eauthInfo.apMac[4],deauthInfo.apMac[5]);
            int ret2 = system(tempbuf);
            //--ignore-negative-one

        memset(&(tempbuf[0]),0,250);
        memset(&(chanbuf[0]),0,250);
        }
         mtx_lock1.unlock();
         sleep(0.01);
    }

}
```

[deauth.h] <deauth_mutex>

```
#ifndef DEAUTH_H
#define DEAUTH_H

#include <cstdio>
#include <iostream>
#include "listload.h"
#include "dbmanage.h"
#include <stdlib.h>
#include <map>
#include <cstring>
```

```
#include <unistd.h>
#include <mutex>


class deauth
{
public:
	int numTest;
	listload::bwList bwDatas;
	typedef struct deauthSet{
		uint8_t apMac[6];
		int channel;
		int blockStat;
		uint8_t stMac[6];
	}deauthSet;
	deauthSet deauthInfo;
	deauth();
	void testFunc11(listload& listMan,dbmanage& wipsDB,std::mutex& mtx_lock1);
};

#endif // DEAUTH_H
```

[listload.cpp] <check error>

```
#include "listload.h"
#include <typeinfo>
#include <cstring>



listload::listload()
{
//make map for dev list
//accessList
//getDevList();
//blackList
//whitelist


}

//basically macField = 0
int listload:: initlist(MYSQL_RES* lRes,int lFlag)
{

	bwStruct={0};
	int fCnt = mysql_num_fields(lRes);
	int rowNum = 0;
	while((row = mysql_fetch_row(lRes)))
	{
```

```
            rowNum = (int)strtol(row[0],NULL,10);
           // debug  printf("num check1 is %d₩n",rowNum);
            convMac(0,row[1]);
            bwStruct.channel = (int)strtol(row[2],NULL,10);
            bwStruct.blockStat = (int)strtol(row[3],NULL,10);
            convMac(1,row[4]);
            bwStruct.apAuth = (int)strtol(row[5],NULL,10);
            bwStruct.apCipher = (int)strtol(row[6],NULL,10);
            bwStruct.apEnc = (int)strtol(row[7],NULL,10);
            bwStruct.macType = (int)strtol(row[8],NULL,10);
            bwStruct.adHocStat = (uint8_t)strtol(row[9],NULL,10);
            //memcpy(&(bwStruct.ssid[0]),&row[10],32);
            // debug printf("st test %02x %02x %02x %02x %02x
%02x₩n",bwStruct.apMac[0],bwStruct.apMac[1],bwStruct.apMac[2],bwStruct.apMac[3],bwStruc
t.apMac[4],bwStruct.apMac[5]);
           // debug  printf("₩n");
            //
            sprintf(bwStruct.ssid,"%s",row[10]);
            //printf("ssid is %s₩n",row[10]);
            if(bwStruct.macType == 0){
                memset(&(bwStruct.stMac[0]),0,6);

            }
            if(lFlag == 10){
                BlackList.insert(std::make_pair(rowNum,bwStruct));
            }else if(lFlag == 11){
                WhiteList.insert(std::make_pair(rowNum,bwStruct));
            }

        }
        blkCnt = rowNum;
        /* debug code
        bw_list::iterator it;
        for(it = BlackList.begin();it !=BlackList.end();it++){
            printf("first %d₩n",it->first);
            if((it->first) == 5){
                printf("struct %02h₩n",(listload::bwList)it->second);
            }
        }*/
        printf("list enroll DB to MAP!₩n₩n");
        return 0;
}

int listload:: initwht(MYSQL_RES* lRes)
{

        bwStruct={0};
        int fCnt = mysql_num_fields(lRes);
        int rowNum = 0;
        while((row = mysql_fetch_row(lRes)))
```

```
	{
		rowNum = (int)strtol(row[0],NULL,10);
		// debug printf("num check1 is %d₩n",rowNum);
		convMac(0,row[1]);
		bwStruct.channel = (int)strtol(row[2],NULL,10);
		bwStruct.blockStat = (int)strtol(row[3],NULL,10);
		convMac(1,row[4]);
		bwStruct.apAuth = (int)strtol(row[5],NULL,10);
		bwStruct.apCipher = (int)strtol(row[6],NULL,10);
		bwStruct.apEnc = (int)strtol(row[7],NULL,10);
		bwStruct.macType = (int)strtol(row[8],NULL,10);
		bwStruct.adHocStat = (uint8_t)strtol(row[9],NULL,10);
		// debug  printf("st test %02x %02x %02x %02x %02x
%02x₩n",bwStruct.apMac[0],bwStruct.apMac[1],bwStruct.apMac[2],bwStruct.apMac[3],bwStruc
t.apMac[4],bwStruct.apMac[5]);
		// debug  printf("₩n");

		BlackList.insert(std::make_pair(rowNum,bwStruct));


	}
	bw_list::iterator it;
	for(it = BlackList.begin();it !=BlackList.end();it++){
		printf("first %d₩n",it->first);


	}
	//printf("thie end₩n₩n");
	return 0;
}

void listload:: convMac(int macFlag,std::string recvMac){
	//std::vector<char> cMac(recvMac.c_str(), recvMac.c_str() + recvMac.size() + 1);
	//char cMac[12] = (char)recvMac[];
	//printf("%s₩n",cMac);
	//char const *test1 = recvMac.c_str();
	//char* test1;
	//std::strcpy(test1, recvMac.c_str());
	//char hexbyte[3] = {0,};
	//int octets[sizeof(recvMac) / 2];
	int arrCnt = 0;
	if(recvMac.length() == 12)
	{
		if(macFlag == 0)//apMac
		{
			for(int ex = 0;ex<12;ex+=2 ){
				std::string tempstr = recvMac.substr(ex,2);
				uint8_t num1 = (uint8_t)strtol(tempstr.c_str(),NULL,16);
				arrCnt = ex/2;
				bwStruct.apMac[arrCnt] = num1;
```

```
                    //printf("ok %02x₩n",bwStruct.apMac[arrCnt]);
                    }
                //char *te1 = "90";
               // printf("test %d₩n",bwStruct.apMac[0]);
                //int test3 = std::atoi("");

                //printf("test2sdf %02x₩n₩n",num1);

            }else if(macFlag == 1)//stationMac
            {
                for(int ex = 0;ex<12;ex+=2 ){
                    std::string tempstr = recvMac.substr(ex,2);
                    uint8_t num1 = (uint8_t)strtol(tempstr.c_str(),NULL,16);
                    arrCnt = ex/2;
                    bwStruct.stMac[arrCnt] = num1;
                    //printf("ok %02x₩n",bwStruct.stMac[arrCnt]);
                    }

            }
        }//else{//not 12 length mac then
            //uint8_t num1 = (uint8_t)strtol(recvMac.c_str(),NULL,16);
    // }

}

void listload:: getPktInfo(uint8_t* pktData){
    //uint8_t* rth_data = pktData+RTHLENGTH;
    uint8_t* rssAdr = pktData+RTHLENGTH;
    uint8_t* channelAdr = pktData+RTHLENGTH;

    rthFrame =  (packframes::rth *)pktData;
    int d_form=4;
    int padding=0;
    //D printf("₩nchannel cur adr = %p₩n",channelAdr);
    for(int temp=0;temp<29;temp++){
        if(rthFrame->rth_present_flg.pflg1[temp]== false){
            continue;
        }
        else if(rthFrame->rth_present_flg.pflg1[temp]==true){
            while((d_form % PFrame.pflg_allign[temp])!=0){

                d_form+=1;
                padding+=1;
            }
            if(temp<3){
                //D printf("₩nchan_tmp = %d, pading = %d
size=%d₩n",temp,padding,PFrame.pflg_size[temp]);
                channelAdr=channelAdr+padding+PFrame.pflg_size[temp];
                //D printf("₩nchannel cur adr = %p₩n",channelAdr);
            }
```

```
                d_form= d_form+PFrame.pflg_size[temp];
                padding=0;
                continue;
            }

        }
        rssAdr += (d_form-4);
        infoForm.channel = ntohs((uint16_t)*channelAdr);
        infoForm.rss = (char)*rssAdr;



        //D printf("₩n₩n₩n rth_data = %p ₩n rthdata is %02x₩n",channelAdr,channelAdr[0]);
        //D printf("₩ncur rss_position = %p₩n data is %02x
%02x₩n",rssAdr,rssAdr[0],rssAdr[1]);

//-------------------rth_data_abstract_end--------------------

        //printf("₩nsize of struct = %d₩n",sizeof();

        //while(pflg_index){
            //for(int temp;temp<4;temp++){
                //&cmp_v
            //}
         //printf("₩ntt₩n");

       // }
        /* uint8_t test1 = (rthFrame->rth_present_flg.pflg1.TSFT);
        uint8_t test2 = (rthFrame->rth_present_flg.pflg1.Flags);
        uint8_t test3 = (rthFrame->rth_present_flg.pflg1.Rate);
        uint8_t test4 = (rthFrame->rth_present_flg.pflg1.Channel);
        uint8_t test5 = (rthFrame->rth_present_flg.pflg1.dbm_antenna_signal);
        uint8_t test6 = (rthFrame->rth_present_flg.pflg1.RX_flags);
        uint8_t test7 = (rthFrame->rth_present_flg.pflg1.radiotap_NS_next);
        uint8_t test8 = (rthFrame->rth_present_flg.pflg1.Ext);
        uint8_t test9 = (rthFrame->rth_present_flg.pflg2.dbm_antenna_signal);
        printf("₩ntest pflg1 = %02x ₩n fhss = %02x %02x %02x %02x %02x %02x %02x
%02x ₩n",test1,test2,test3,test4,test5,test6,test7,test8,test9);
        */
        //enum packframes::RTH_pflg presentFlag;
        // =

        /*while(presentFlag != 31){
            if(rthFrame->rth_present_flg.pflg1[presentFlag]|rthFrame->rth_present_flg.pflg2)
            switch(presentFlag)
            {
                case(packframes::RTH_TSFT):
                    printf("1");
                    continue;
```

```
                default:
                    printf("end");
            }
        }*/
        /*
        presentFlag = packframes::RTH_TSFT;
        printf("₩n enum data = %d₩n",presentFlag);
        presentFlag = packframes::RTH_FLAGS;
        printf("₩n enum data2 = %d₩n",presentFlag);
        */
         //while()
        //switch
       // rthFrame->rth_length
         //switch()


}
listload::bw_list& listload::rtnBlkMap(){

        return this->BlackList;
}
```

[listload.h] <check error>

```
#pragma once
#include <stdio.h>
#include <iostream>
#include <list>
#include <vector>
#include <map>
#include <pcap/pcap.h>
#include <mysql/mysql.h>
#include "dbmanage.h"
#include "mac.h"
#include "packframes.h"
#include <arpa/inet.h>
#include <string>
#include <cstring>
#include <stdlib.h>



#define WHITEAP 1
#define WHITESTATION 2
#define BLACKAP 3
#define BLACKSTATION 4

#define RTHLENGTH     12
```

```
class listload
{

    public:
    uint8_t cmp_v = 1;
    MYSQL_ROW row;
    mac tempMac;

 //--------------------------get pkt info(rth+mac)

    //packframes pktForm;
    packframes PFrame;
    packframes::rth* rthFrame;
    #pragma pack(push,1)
    typedef struct recv_info{
        mac apMac;
        char rss;
        uint16_t channel;
        mac stationMac;
    }info;
    info infoForm;
 //-------------blk----------------

    int blkCnt = 0;

    typedef struct blk_wht_list{
        uint8_t apMac[6];
        int channel;
        int blockStat;
        uint8_t stMac[6];
        int apAuth;
        int apCipher;
        int apEnc;
        int macType;
        uint8_t adHocStat;
        char ssid[32];



    }bwList;
    bwList bwStruct;
    #pragma pack(pop)
    typedef std::map<int,bwList>bw_list;
    bw_list BlackList;
    bw_list WhiteList;
    //typedef std::map<int,bwList>::iterator blk_list_iter;
    //blk_list_iter BlackIter;
    typedef std::map<int,info>pkt_info;
    typedef std::map<int,info>::iterator pkt_info_iter;
```

```
//----------------------white----------------------

//-----------------

      //MYSQL_RES  *listRes;
     /*  typedef  std::map<int,std::string>  wApMap;
      wApMap  CipwAp;//int,string
      typedef  std::map<int,std::string>::iterator  wApIter;
      typedef  std::map<int,std::string>  wStMap;
      wStMap  CipwSp;//int,string
      typedef  std::map<int,std::string>::iterator  wStIter;
      typedef  std::map<int,std::string>  bStMap;
      bStMap  CipbAp;//int,string
      typedef  std::map<int,std::string>::iterator  bAPIter;
*/
            listload();
            void  convMac(int  macFlag,std::string  recvMac);
            int  initlist(MYSQL_RES*  lRes,int  lFlag);
            int  initwht(MYSQL_RES*  lRes);
            void  getPktInfo(uint8_t*  pktData);
            bw_list&  rtnBlkMap();

};
```

[mac.cpp] <recv string to hex>

```
#include  "mac.h"

mac::mac()
{

}
```

[mac.h] <protect my data>

```
#ifndef  MAC_H
#define  MAC_H
#include  <cstdint>
#include  <string>
#include  <cstdlib>
#include  <iostream>
#include  <cstring>


#include  <map>
#include  <tuple>
```

```
class mac
{
public:
	mac();
	uint8_t macAddr[6];
	mac& operator =(char *strAddr)
	{
		std::string baseMac = "000000000000";
		baseMac.replace(12-(std::strlen(strAddr)),std::strlen(strAddr),strAddr);
		std::string te2;
		//printf("macLen = %d₩n",strlen(strAddr));

		for(int cnt = 5;cnt>-1;cnt--)
		{
		  // std::cout<<"string="<<baseMac<<'₩n';
		   int num=cnt*2;
		   te2 = baseMac.substr(num,2);
		   //std::cout<<"te2="<<te2<<'₩n';
		   macAddr[cnt] = (uint8_t)(std::stoi(te2,0,16));
		   printf("%d₩n",macAddr[cnt]);

		}

		//if()
	return *this;
	}
	bool operator<(const mac sMac)	const{
			return
std::tie(macAddr[0],macAddr[1],macAddr[2],macAddr[3],macAddr[4],macAddr[5])<std::tie(sMac
.macAddr[0],sMac.macAddr[1],sMac.macAddr[2],sMac.macAddr[3],sMac.macAddr[4],sMac.mac
Addr[5]);
	}

};

#endif // MAC_H
```

[main.cpp] <deauth_mutex>

```
#include <iostream>
#include "listload.h"
#include "dbmanage.h"
#include "pktpassway.h"
#include "deauth.h"
#include <thread>
using namespace std;

#define BLACKLISTFLAG 10
#define WHITELISTFLAG 11
```

```
int main(int argc, char *argv[])
{
	char a = 0xba;
	dbmanage wipsDB;
	listload listMan;
	pktPassWay test;
	std::mutex mtx_lock;
	deauth dd;
	printf("BLACKLIST ENROLL₩n");


	int stat =  listMan.initlist(wipsDB.dbQuery("SELECT * FROM
wips_black_blacklist"),BLACKLISTFLAG);
	printf("WHITELIST ENROLL₩n");
	int stat1 =  listMan.initlist(wipsDB.dbQuery("SELECT * FROM
wips_white_whitelist"),WHITELISTFLAG);
	//thread t1{&pktPassWay::main,&test,listMan,wipsDB};
	thread t1([&](){test.main(listMan,wipsDB,mtx_lock);});
	thread t2([&](){dd.testFunc11(listMan,wipsDB,mtx_lock);});
	//t1.join();
	t2.join();
	return 0;
}
```

[packframes.h] <input db okay>

```
#ifndef PACKFRAMES_H
#define PACKFRAMES_H
#include "cstdint"
#include <bitset>


class packframes
{
public:
#pragma pack(push,1)
/*
	typedef struct radiotap_hdr_present_flag_word{
		uint8_t TSFT                    : 1;
		uint8_t Flags                   : 1;
		uint8_t Rate                    : 1;
		uint8_t Channel                 : 1;
		uint8_t FHSS                    : 1;
		uint8_t dbm_antenna_signal      : 1;
		uint8_t dbm_antenna_noise       : 1;
		uint8_t Lock_quality            : 1;
		uint8_t TX_attenuation          : 1;
		uint8_t db_TX_attenuation       : 1;
		uint8_t dbm_TX_power            : 1;
		uint8_t Antenna                 : 1;
```

```
        uint8_t db_Antenna_signal     : 1;
        uint8_t db_Antenna_noise      : 1;
        uint8_t RX_flags              : 1;
        uint8_t non_data              : 3;
        uint8_t Channel_plus          : 1;
        uint8_t mcs_information       : 1;
        uint8_t a_mpdu_status         : 1;
        uint8_t none_data1            :1;
        uint8_t none_data2            :1;
        uint8_t reserved1             :6;
        uint8_t radiotap_NS_next      :1;
        uint8_t vendor_NS_next        :1;
        uint8_t Ext                   :1;
    }pflg_word;*/
 //-------------------------------radiotap header setting
    enum RTH_pflg {
        RTH_TSFT = 0,
        RTH_FLAGS = 1,
        RTH_RATE = 2,
        RTH_CHANNEL = 3,
        RTH_FHSS = 4,
        RTH_DBM_ANTSIGNAL = 5,
        RTH_DBM_ANTNOISE = 6,
        RTH_LOCK_QUALITY = 7,
        RTH_TX_ATTENUATION = 8,
        RTH_DB_TX_ATTENUATION = 9,
        RTH_DBM_TX_POWER = 10,
        RTH_ANTENNA = 11,
        RTH_DB_ANTSIGNAL = 12,
        RTH_DB_ANTNOISE = 13,
        RTH_RX_FLAGS = 14,
        RTH_TX_FLAGS = 15,
        RTH_RTS_RETRIES = 16,
        RTH_DATA_RETRIES = 17,
        RTH_XChannel = 18,
        RTH_MCS = 19,
        RTH_AMPDU_STATUS = 20,
        RTH_VHT = 21,
        RTH_TIMESTAMP = 22,
        RTH_RADIOTAP_NAMESPACE = 29,
        RTH_VENDOR_NAMESPACE = 30,
        RTH_EXT = 31
    };

/*flag_to_size_rth
    {
        TSFT            0     size= 8 allign=8
        flags           1     size= 1 allign=1;
        rate            2     size= 1 allign=1
        channel         3     size= 4 allign=2
```

```
        fhss                    4      size= 2 allign=???(2?)
        antena signal           5      size= 1 allign=1;dbm
        antena noise            6      size= 1 allign=1;
        lock quality            7      size= 2 allign=2
        tx attenuation          8      size= 2 allign=2
        db_tx_attenuation       9      size= 2 allign=2
        dbm_tx_power            10     size= 1 allign=1
        antena                  11     size= 1 allign=1
        db_antena_signal        12     size= 1 allign=1
        db_antena_noise         13     size= 1 allign=1
        rx_flags                14     size= 2 allign=2
                                15
                                16
                                17
                                18
        mcs                     19     size= 3 allign=1
        mpdu                    20     size= 6 allign=4
        vht                     21     size=12 allign=2
        timestamp               22     size=12 allign=8
        vendor_namespace        30     size=6  allign=2
        //uint8_t radiotap_ns_next;*
        //uint8_t Vendor_NS_next ;
        //uint8_t ext;*
        // is need but don't need size?
    }rth_flag_size;*/
        int pflg_size[32]={8,1,1,4,2,1,1,2,2,2,1,1,1,1,2,0,0,0,0,3,6,12,12,0,0,0,0,0,0,0,6,0};
                                //      4       8       12      16      20      24
28      32
        int pflg_align[32]={12,1,1,2,2,1,1,2,2,2,1,1,1,1,2,0,0,0,0,1,4,2,8,0,0,0,0,0,0,0,2,0};
                                //      4       8       12      16      20      24
28      32
    typedef struct radiotap_hdr_present_flags{
        std::bitset<32> pflg1;
        std::bitset<32> pflg2;
    }pflgs;

    typedef struct radiotap_80211_hdr{
        uint8_t rth_version;
        uint8_t rth_pad;
        uint16_t rth_length;
        pflgs rth_present_flg;
    }rth;
//-------------------abstract datas(rth)
    typedef struct radiotap_80211_hdr_data{
        uint8_t* rss;
        uint16_t* channel;

    }rth_data;
    //void
    //struct management frame
```

```
//----------------80211 frame----------------------------


typedef struct FrameCtrl
{
    uint8_t    protocolVer      : 2;
    uint8_t    type             : 2;
    uint8_t    subType          : 4;
    uint8_t    toDs             : 1;
    uint8_t    fromDs           : 1;
    uint8_t    moreFlag         : 1;
    uint8_t    retry            : 1;
    uint8_t    powerMgmt        : 1;
    uint8_t    moreData         : 1;
    uint8_t    protectedFrame : 1;
    uint8_t    order            : 1;
}FC;

struct ManagementFrame
{
            FC   frameCtrl;   //2 bytes
            uint16_t    duration;    //2 bytes
            uint8_t     addr1[6];    //6 bytes
            uint8_t     addr2[6];    //6 bytes
            uint8_t     addr3[6];    //6 bytes
            uint16_t    seq_ctrl;    //2 bytes
};

struct BeaconFrameBody
{
    uint64_t   timestamp;
    uint16_t   beaconInterval;
    uint16_t   capacityInformation;
};

struct TagBody
{
    uint8_t    elementID;
    uint8_t    tagLength;
};

struct WifiName
{
    uint8_t ssid[32];
};
//void fakeAp(const uint8_t *);


#pragma pack(pop)
```

```
	packframes();
	//void get_fucking_addr();
};

#endif // PACKFRAMES_H
```

[pktpassway.cpp] <deauth_mutex>

```
#include "pktpassway.h"
#define BLACKLISTFLAG 10
#define WHITELISTFLAG 11


using namespace std;

//char pktpassway::*correct_dev(int argu_count,char *argu_vector);


int pktPassWay::main(listload& listMan,dbmanage& wipsDB,std::mutex& mtx_lock)
{
	//database connect
	//dbmanage wipsDB;
	//listload listMan;

	//query(dbMacField,query)

	//printf("BLACKLIST ENROLL₩n");
	//int stat = listMan.initlist(wipsDB.dbQuery("SELECT * FROM
wips_black_blacklist"),BLACKLISTFLAG);
	//printf("WHITELIST ENROLL₩n");
	//int stat1 = listMan.initlist(wipsDB.dbQuery("SELECT * FROM
wips_white_whitelist"),WHITELISTFLAG);
	//initTbl send wipsdb. query table
	// debug printf("thus₩n");

	//devCheck
	//char* dev = correct_dev(argc,argv[1]);
	char *dev = "wlan1";
	char errbuf[PCAP_ERRBUF_SIZE];
	pcap_t * pktDescrpt = pcap_open_live(dev, BUFSIZ, PROMISCUOUS, 0, errbuf);
	if(pktDescrpt == NULL) {
		printf("%s₩n",errbuf);
		exit(1);
	}

	//pkt recv
	int loopStat;
	struct pcap_pkthdr *pktHdr;
	const u_char *pktData;
	printf("packets waiting...₩n");
```

```
    //for(int i = 0;i<100000;i++){
      //   printf("pkt cap test %d₩n",i);

   // }
    while(true)
    {

        loopStat = pcap_next_ex(pktDescrpt, &pktHdr, &pktData);
        //(void)pktHdr;//useless
        //pcap_stats(packeDescrpt,&stat);//useless

        switch(loopStat)
        {
        case 1:
        {
                pktFilter((uint8_t*)pktData,listMan, wipsDB,mtx_lock);
                break;
        }
            case 0:
                continue;//timeout check
            case -1:
                pcap_perror(pktDescrpt,"Packet data read error");
                exit(1);
            case -2:
                pcap_perror(pktDescrpt,"Packet data read error");
                exit(1);
        }
    }
    printf("end while₩n");
    return 0;
}

char* pktPassWay::correct_dev(int argCnt,char *argVector)
{
    if (argCnt != 2){
        printf("use this form to use program₩nProgramName DeviceName₩n");
        printf("available dev Lists₩n");
        devsearch t1;
        t1.GetDevList();

        exit(1);
    }
    printf("Device : %s₩n", argVector);
    return argVector;
}



void pktPassWay::pktFilter(uint8_t *pktData,listload& listMan1,dbmanage&
wipsDB1,std::mutex& mtx_lock1)
```

```
{
    int debugCnt = 0;
    //uint8_t*
    usrfunc usrFunc(pktData);
    //printf("============================PACKET
CAPTURE================================₩n₩n");
    memset(&(usrFunc.exPkt),0,sizeof(usrFunc.exPkt));
    //usrFunc.test_viewFunc(listMan1);
    //make reference
    //printf("")

    switch(usrFunc.frameCtrl->type){
        case(0):{
            //D memList.getPktInfo(pktData);

            switch(usrFunc.frameCtrl->subType){

                case(8):
                {
                    mtx_lock1.lock();
                    usrFunc.getCurPktData(listMan1);
                    if(usrFunc.doFlag == true){
                        break;
                    }
                    printf("============================PACKET
CAPTURE================================₩n₩n");
                    int aa;
                    usrFunc.macCmp(listMan1);
                    usrFunc.adhocFunc(listMan1);


                    usrFunc.fakeAp(listMan1);
                    aa = usrFunc.misconfigureAP(listMan1);
                    //usrFunc.test_viewFunc(listMan1);

                    //printf("misconf₩n");




                    //memList.getPktInfo(pktData);
                    if(usrFunc.storFlag == true){
                        usrFunc.inputCurPkt(listMan1,wipsDB1);
                        usrFunc.storFlag = false;
                        memset(usrFunc.atkType,0,sizeof(usrFunc.atkType));

                        }
                    debugCnt+=1;
                    if(debugCnt == 10)
                        system("PAUSE");
```

```
                        mtx_lock1.unlock();
                        break;
                        }
                    case(10):{
                       //D  memList.getPktInfo(pktData);
                        break;

                    }
                    default:
                        break;
                }
            break;
            }
            case(1):
            //type  is  1
                printf("₩n");
                break;
            case(2):
                printf("₩n");
                break;
            case(3):
                printf("₩n");
                break;
        }



}
```

[pktpassway.h] <deauth_mutex>

```
#pragma  once
#include  <iostream>
//#include  <ctypes>
//#include  "pktpassway.h"
#include  "listload.h"
#include  "dbmanage.h"
#include  "devsearch.h"
#include  "packframes.h"
#include  "usrfunc.h"
#include  <mutex>

#define  PROMISCUOUS  1
#define  NONPROMISCUOUS  0

class  pktPassWay
{
    //char  *correct_dev(int  argu_count,char  *argu_vector);
private:
public:
```

```
	//std::mutex mtx_lock;
	int main(listload& listMan,dbmanage& wipsDB,std::mutex& mtx_lock);
	char *correct_dev(int argCnt,char *argVector);
	void pktFilter(uint8_t *pktData,listload& listMan1,dbmanage& wipsDB1,std::mutex&
mtx_lock1);
	listload memList;
};
```

[usrfunc.cpp] <deauth_mutex>

```
#include "usrfunc.h"

usrfunc::usrfunc(uint8_t *packet)
{
	this->pktPoint = packet;
	this->RTHeader = (packframes::rth*)((uint8_t*)packet);
	this->frameCtrl = (packframes::FC*)((uint8_t*)(packet+RTHeader->rth_length));
	this->mgmtFrame =
(packframes::ManagementFrame*)((uint8_t*)(packet+RTHeader->rth_length));
	memset(this->APMac,0,6);
	memset(this->cliMac,0,6);
	//when mgmt then set ap,cli
	//need set um....
}
void usrfunc::fakeAp(listload& listMan2)
{
	WHTFlag = false; //true in whitelisted packet
	BLKFlag = false; //true in blacklisted packet
	uint8_t* pktPoint2 = this->pktPoint;

	struct packframes::WifiName *wifiName;

//BSSID*****************************************************************************//
		printf("BSSID: ");
		for(int i=0; i<6; i++)
		{
			printf("%02x ", mgmtFrame->addr3[i]);
		}
		printf("₩n");


//SSID******************************************************************************//
		pktPoint2 += (RTHeader->rth_length + sizeof(struct
packframes::ManagementFrame) + sizeof(struct packframes::BeaconFrameBody) +
sizeof(struct packframes::TagBody));
		wifiName = (struct packframes::WifiName *)pktPoint2;
		uint8_t* lengthPnt = ((uint8_t*)pktPoint2) - 1;
		printf("cur leng is %d₩n",*lengthPnt);
		if(*pktPoint2 == 0){
			memset(&(exPkt.ssid),0,32);
```

```
                return;
            }
            /*printf("SSID:   ");
            for(int i=0; i<32; i++)
            {
                if(pktPoint2[i] == 1)
                    break;
                printf("%c", wifiName->ssid[i]);
            }
    printf("₩n₩n");*/
    int cmpFlag = 0;
    int macCmpFlag = 0;
    //uint8_t cmpArray[6];
    listload::bw_list::iterator it;

    printf("-----------------------------------------------₩n");
      //wht list!
    printf("CHECK fakeAp₩n₩n");
     for(it = listMan2.WhiteList.begin();it !=listMan2.WhiteList.end();it++){
        //printf("view func first %d₩n",it->first);
        bwDatas = (listload::bwList)it->second;
        macCmpFlag = memcmp(&(bwDatas.apMac),&(mgmtFrame->addr3),6);
        if(macCmpFlag == 0){
            printf("cmp ssid %s₩n",bwDatas.ssid);

            cmpFlag = memcmp(&(bwDatas.ssid),&(wifiName->ssid),*lengthPnt);
            if(cmpFlag == 0){
                printf("LISTED SSID!(WHITE)₩n");
                WHTFlag = true;
                break;
                //alreay have white list OK.
            }
        }
    }

    cmpFlag = 0;
    macCmpFlag = 0;
    if(WHTFlag == false){
        printf("NOT LISTED SSID(WHITE) ₩n");
        cmpFlag = 0;
        //uint8_t cmpArray[6];
        listload::bw_list::iterator it1;


                //blk list!
        for(it1 = listMan2.BlackList.begin();it1 !=listMan2.BlackList.end();it1++){
            //printf("view func first %d₩n",it->first);
            bwDatas = (listload::bwList)it1->second;
            //printf("cmp with %02x %02x %02x %02x %02x
%02x₩n",bwDatas.apMac[0],bwDatas.apMac[1],bwDatas.apMac[2],bwDatas.apMac[3],bwDatas.
```

```
apMac[4],bwDatas.apMac[5]);
                //printf("check blk SSID₩n");
                macCmpFlag = memcmp(&(bwDatas.apMac),&(mgmtFrame->addr3),6);
                if(macCmpFlag == 0){
                    printf("cmp ssid %s₩n",bwDatas.ssid);

                    cmpFlag = memcmp(&(bwDatas.ssid),&(wifiName->ssid),*lengthPnt);
                    if(cmpFlag == 0){
                        printf("LISTED SSID(BLACK)₩n");
                        BLKFlag = true;
                        break;
                        //alreay have black list OK.
                    }
                }

            }

        }
        memset(&(exPkt.ssid),0,32);
        memcpy(&(exPkt.ssid),&(wifiName->ssid),*lengthPnt);
        printf("exPkt ssid %s₩n",exPkt.ssid);

        if(WHTFlag == false && BLKFlag == false){
            printf("NOT LISTED SSID(BLACK) ₩n");
            printf("₩ndisordered!(FAKE AP)₩n₩n");
            storFlag = true;
            char tempbuf[250] = {0,};

            sprintf(tempbuf,"%s FAKEAP",atkType);
            sprintf(atkType,"%s",tempbuf);
            //start function(structure write and throw map,db)

        }


}




int usrfunc::Cipher(uint8_t cipher)
{
    switch (cipher)//Group Cipher Suite(broad,multicast) Type
                    {

                    case 0:
                        printf("Group Cipher Suite 사용₩n");
                        cipher = GROUP_CIPHER_SUITE; //Group Cipher Suite flsg:21
                        break;
```

```
                    case 1:
                        printf("WEP-40 사용₩n");
                        cipher = WEP40; //WEP-40 flag:12
                        break;

                    case 2:
                        printf("TKIP 사용₩n");
                        cipher = TKIP; //TKIP flag:13
                        break;

                    case 3:
                        printf("예약 사용₩n");
                        cipher = C_RESERVATON; //예약 flag:14
                        break;

                    case 4:
                        printf("CCMP 사용₩n");
                        cipher = CCMP; //CCMP flag:15
                        break;

                    case 5:
                        printf("WEP-104 사용₩n");
                        cipher = WEP104; //WEP-104 flag:16
                        break;

                printf("Cipher fun:%d₩n", cipher);
                    }
    return(cipher);
}

int usrfunc::Auth(uint8_t auth)
{
    switch (auth)
    {
    case 0:
        printf("예약 사용₩n");
        auth = A_RESERVATON;//flag:17
        break;

    case 1:
        printf("802.1X 인증₩n");
        auth = A_8021X;//flag:18
        break;

    case 2:
        printf("PSK 인증₩n");
        auth = PSK;//flag:19
        break;

    default:
```

```
            printf("...₩n");
            break;
        }
        return(auth);

}


int usrfunc::misconfigureAP(listload& listMan2)
{

        WHTFlag = false; //true in whitelisted packet
        BLKFlag = false; //true in blacklisted packet
        uint8_t *data = this->pktPoint;
        struct RadiotapHeader *rH;//
        struct RadiotapHeaderFlag *rF;//
        struct ManagementFrame *mF;//
        struct FrameCtrl *fC;//
        struct OptionField *oF;//
        struct Rsn *rsn;//
        struct VendorSpecific *vS;//
        struct SecurityMethod sM;
        struct SecurityFlag sF;
        struct SecurityFlag listSF;
        //struct AkmSuiteSelector *aS;
       // struct PairwiseCipherSuiteSelector *pS;


        rH = (struct RadiotapHeader *)data;
        rF = (struct RadiotapHeaderFlag *)data;
        mF = (struct ManagementFrame *)(data + rH->length);
        fC = (struct FrameCtrl *)(data + rH->length);
        oF = (struct OptionField *)(data + rH->length + sizeof(struct ManagementFrame) +
sizeof(struct BeaconFrameBody));
        rsn = (struct Rsn *)data;
        vS = (struct VendorSpecific *)data;


        printf("--------------------------------₩n");
        printf("AP RULE CHECK₩n₩n");

        uint8_t apMac[6]; //BSS = AP MAC
        //printf("ok₩n₩n");
        sM.wep = fC->protectedFrame;
        //int16_t tpcss;

/*비콘 아닐때도 48, 221있는듯? 221은 확실히있음
그럼 비콘일때 ap mac 잡아버리면 ap mac 없는 패킷도 나옴 암호는 나오지만
그냥 비콘일때로 통일?*/
        apMac[0] = mF->addr3[0];
```

```
    apMac[1] = mF->addr3[1];
    apMac[2] = mF->addr3[2];
    apMac[3] = mF->addr3[3];
    apMac[4] = mF->addr3[4];
    apMac[5] = mF->addr3[5];
    printf("AP MAC:
%02x:%02x:%02x:%02x:%02x:%02x₩n",apMac[0],apMac[1],apMac[2],apMac[3],apMac[4],apMa
c[5]);

            if(fC->subType == 8 && fC->type == 0)//BeaconFrame
            {
                apMac[0] = mF->addr3[0];
                apMac[1] = mF->addr3[1];
                apMac[2] = mF->addr3[2];
                apMac[3] = mF->addr3[3];
                apMac[4] = mF->addr3[4];
                apMac[5] = mF->addr3[5];


                while((oF->elementId | oF->length )!= 0)//OptionFrame
                {
                    if(oF->elementId == 48)//RSN ID:48
                    {
                        rsn = (Rsn *)(uint8_t*)oF;

                        sM.gCSS[0] = rsn->gCSS[0];//00-0f-ac
                        sM.gCSS[1] = rsn->gCSS[1];
                        sM.gCSS[2] = rsn->gCSS[2];
                        sM.gCSS[3] = rsn->gCSS[3];//Type


                        {



                            switch (rsn->pCSC)
                            {
                            case 1:
                                sM.pCSS[3] = rsn->pCSS.pOUI[3];
                                sM.aSS[3] = rsn->aSS[3];
                                sM.aSC[0] = rsn->aSC[0];
                                sM.aSC[1] = rsn->aSC[1];
                                break;

                            case 2:
                                sM.pCSS[3] = rsn->pCSS.pOUI[7];
                                sM.aSS[3] = rsn->aSS[7];
                                sM.aSC[0] = rsn->aSC[2];
                                sM.aSC[1] = rsn->aSC[3];
                                break;
```

```
                    case 3:
                        sM.pCSS[3] = rsn->pCSS.pOUI[11];
                        sM.aSS[3] = rsn->aSS[7];
                        sM.aSC[0] = rsn->aSC[4];
                        sM.aSC[1] = rsn->aSC[5];
                        break;

                    case 4:
                        sM.pCSS[3] = rsn->pCSS.pOUI[15];
                        sM.aSS[3] = rsn->aSS[15];
                        sM.aSC[0] = rsn->aSC[6];
                        sM.aSC[1] = rsn->aSC[7];
                        break;

                    default:
                        printf("not found/₩n");
                        break;
                    }


                if(sM.aSC[0] != 0x01)
                {
                   switch (sM.aSC[0])
                  {

                  case 2:
                        printf("AKM OUI : %02x-%02x-%02x₩n", rsn->aSS[4],
rsn->aSS[5], rsn->aSS[6]);//AKM OUI ID=48 -> WPA-2
                        printf("AKM TYPE : %02x₩n",rsn->aSS[7]);//AKM TYPE
ID=48 -> WPA-2
                        sM.aSS[3] = rsn->aSS[7];
                        break;

                  case 3:
                        printf("AKM OUI : %02x-%02x-%02x₩n", rsn->aSS[8],
rsn->aSS[9], rsn->aSS[10]);//AKM OUI ID=48 -> WPA-2
                        printf("AKM TYPE : %02x₩n",rsn->aSS[11]);//AKM TYPE
ID=48 -> WPA-2
                        sM.aSS[3] = rsn->aSS[11];
                        break;

                  case 4:
                        printf("AKM OUI : %02x-%02x-%02x₩n", rsn->aSS[12],
rsn->aSS[13], rsn->aSS[14]);//AKM OUI ID=48 -> WPA-2
                        printf("AKM TYPE : %02x₩n",rsn->aSS[15]);//AKM TYPE
ID=48 -> WPA-2
                        sM.aSS[3] = rsn->aSS[15];
                        break;
```

```
                    default:
                        //printf("not found/₩n");
                        break;
                    }
                  }
                }

                sM.aSS[0] = rsn->aSS[0];//00-0f-ac
                sM.aSS[1] = rsn->aSS[1];
                sM.aSS[2] = rsn->aSS[2];

            }

            else if(oF->elementId == 221)//Vendor specific ID:221
            {
                vS = (VendorSpecific *)(uint8_t*)oF;


                if(vS->oUI[0] == 0x00 && vS->oUI[1] == 0x50 && vS->oUI[2]
== 0xf2)//OUI 00-50-f2
                {
                    if(vS->vST == 0x01)//1: WPA Information Type-> WPA-1 &
2: WMM/WME -> CCMP 일때 2임...?
                    {

                    sM.oUI[0] = vS->oUI[0];//00-50-f2
                    sM.oUI[1] = vS->oUI[1];
                    sM.oUI[2] = vS->oUI[2];
                    sM.mCSS[3] = vS->mCSS[3]; //Type =1 WPA Information

                    switch (vS->uCSC)
                    {
                    case 1:
                        printf("Unicast Cipher Suite OUI: %02x-%02x-%02x₩n",
vS->uCSS[0], vS->uCSS[1], vS->uCSS[2]);//Unicast Cipher Suite OUI ID:221
                        printf("Unicast Cipher Suite TYPE: %02x₩n",
vS->uCSS[3]);//Unicast Cipher Suite TYPE ID:221
                        sM.uCSS[3] = vS->uCSS[3];
                        printf("AKM OUI : %02x-%02x-%02x₩n", vS->aSS[0],
vS->aSS[1], vS->aSS[2]);//AKM OUI ID=221 -> WPA-1
                        printf("AKM TYPE : %02x₩n",vS->aSS[3]);//AKM TYPE
ID=221 -> WPA-1
                        sM.aKMS[3] = vS->aSS[3];
                        sM.aKMC[0] = vS->aSC[0];
                        sM.aKMC[1] = vS->aSC[1];
                        break;

                    case 2:
                        printf("Unicast Cipher Suite OUI: %02x-%02x-%02x₩n",
vS->uCSS[4], vS->uCSS[5], vS->uCSS[6]);//Unicast Cipher Suite OUI ID:221
```

```
                                    printf("Unicast Cipher Suite TYPE: %02x₩n",
vS->uCSS[7]);//Unicast Cipher Suite TYPE ID:221
                                    sM.uCSS[3] = vS->uCSS[7];
                                    printf("AKM OUI : %02x-%02x-%02x₩n", vS->aSS[4],
vS->aSS[5], vS->aSS[6]);//AKM OUI ID=221 -> WPA-1
                                    printf("AKM TYPE : %02x₩n",vS->aSS[7]);//AKM TYPE
ID=221 -> WPA-1
                                    sM.aKMS[3] = vS->aSS[7];
                                    sM.aKMC[0] = vS->aSC[2];
                                    sM.aKMC[1] = vS->aSC[3];
                                    break;

                                case 3:
                                    printf("Unicast Cipher Suite OUI: %02x-%02x-%02x₩n",
vS->uCSS[8], vS->uCSS[9], vS->uCSS[10]);//Unicast Cipher Suite OUI ID:221
                                    printf("Unicast Cipher Suite TYPE: %02x₩n",
vS->uCSS[11]);//Unicast Cipher Suite TYPE ID:221
                                    sM.uCSS[3] = vS->uCSS[11];
                                    printf("AKM OUI : %02x-%02x-%02x₩n", vS->aSS[8],
vS->aSS[9], vS->aSS[10]);//AKM OUI ID=221 -> WPA-1
                                    printf("AKM TYPE : %02x₩n",vS->aSS[11]);//AKM TYPE
ID=221 -> WPA-1
                                    sM.aKMS[3] = vS->aSS[11];
                                    sM.aKMC[0] = vS->aSC[4];
                                    sM.aKMC[1] = vS->aSC[5];
                                    break;

                                case 4:
                                    printf("Unicast Cipher Suite OUI: %02x-%02x-%02x₩n",
vS->uCSS[12], vS->uCSS[13], vS->uCSS[14]);//Unicast Cipher Suite OUI ID:221
                                    printf("Unicast Cipher Suite TYPE: %02x₩n",
vS->uCSS[15]);//Unicast Cipher Suite TYPE ID:221
                                    sM.uCSS[3] = vS->uCSS[15];
                                    printf("AKM OUI : %02x-%02x-%02x₩n", vS->aSS[12],
vS->aSS[13], vS->aSS[14]);//AKM OUI ID=221 -> WPA-1
                                    printf("AKM TYPE : %02x₩n",vS->aSS[15]);//AKM TYPE
ID=221 -> WPA-1
                                    sM.aKMS[3] = vS->aSS[15];
                                    sM.aKMC[0] = vS->aSC[6];
                                    sM.aKMC[1] = vS->aSC[7];
                                    break;

                                default:
                                    printf("not found/₩n");
                                    break;
                                }


                                if(sM.aKMC[0] != 0x01)
                                {
```

```
                                    switch (sM.aKMC[0])
                                    {


                                    case 2:
                                        printf("AKM OUI : %02x-%02x-%02x₩n", vS->aSS[4],
vS->aSS[5], vS->aSS[6]);//AKM OUI ID=221 -> WPA-1
                                        printf("AKM TYPE : %02x₩n",vS->aSS[7]);//AKM TYPE
ID=221 -> WPA-1
                                        sM.aKMS[3] = vS->aSS[7];
                                        break;

                                    case 3:
                                        printf("AKM OUI : %02x-%02x-%02x₩n", vS->aSS[8],
vS->aSS[9], vS->aSS[10]);//AKM OUI ID=221 -> WPA-1
                                        printf("AKM TYPE : %02x₩n",vS->aSS[11]);//AKM
TYPE ID=221 -> WPA-1
                                        sM.aKMS[3] = vS->aSS[11];
                                        break;

                                    case 4:
                                        printf("AKM OUI : %02x-%02x-%02x₩n", vS->aSS[12],
vS->aSS[13], vS->aSS[14]);//AKM OUI ID=221 -> WPA-1
                                        printf("AKM TYPE : %02x₩n",vS->aSS[15]);//AKM
TYPE ID=221 -> WPA-1
                                        sM.aKMS[3] = vS->aSS[15];
                                        break;

                                    default:
                                        printf("not found/₩n");
                                        break;
                                    }
                                }


                            }

                            if(vS->vST == 0x02)//1: WPA Information Type-> WPA-1 &
2: WMM/WME
                            {

                            sM.oUI[0] = vS->oUI[0];//00-50-f2
                            sM.oUI[1] = vS->oUI[1];
                            sM.oUI[2] = vS->oUI[2];
                            }
                        }
                    }
                    oF = (OptionField *)((uint8_t*)oF+sizeof(OptionField)+oF->length);
                }
```

```
            }

            //*********************Flag set*******************
            if(sM.oUI[0] == 0x00 && sM.oUI[1] == 0x50 && sM.oUI[2] == 0xf2 &&
                sM.gCSS[0] == 0x00 && sM.gCSS[1] == 0x0f && sM.gCSS[2] ==
0xac)//OUI 00-50-f2 && OUI 00-0f-ac -> WPA2
            {
                sF.enc = WPA2;
                printf("WPA-2: %d₩n", sF.enc);
                sF.groupCipher = Cipher(sM.gCSS[3]);
                sF.pairwiseCipher = Cipher(sM.pCSS[3]);
                sF.auth = Auth(sM.aSS[3]);

            }

            else if(sM.oUI[0] == 0x00 && sM.oUI[1] == 0x50 && sM.oUI[2] ==
0xf2)//OUI 00-50-f2 -> WPA-1
            {

                sF.enc = WPA1; //WPA-1 flsg:10
                printf("WPA-1: %d₩n", sF.enc);

                printf("Group Cipher Suite Selector: %d₩n",sM.mCSS[3]);//Group Cipher
Suite Selector(multicast) Type
                sF.groupCipher = Cipher(sM.mCSS[3]);
                printf("Flag: %d₩n", sF.groupCipher);

                printf("Pairwise Cipher Suite Selector: %d₩n",sM.uCSS[3]);//Pairwise
Cipher Suite Selector(unicast) Type
                sF.pairwiseCipher = Cipher(sM.uCSS[3]);


                printf("AKM Suite Selector: %d₩n",sM.aKMS[3]);//Authentication and Key
Management Type
                sF.auth = Auth(sM.aKMS[3]);

            }

            if(sM.wep == 1)//WEP
            {
                sF.enc = WEP; //WEP flsg:1
                printf("WEP flag: %d₩n", sF.enc);
            }

            if(sM.wep == 0 && rsn->elementId != 48 && vS->elementId !=
221)//OPEN
            {
                sF.enc = OPEN; //OPEN flsg:0
                printf("OPEN flag: %d₩n", sF.enc);
            }
```

```
            exPkt.apCipher = sF.groupCipher;
            exPkt.apEnc = sF.enc;
            int cmpFlag = 0;
            int macCmpFlag = 0;
            listload::bw_list::iterator it;

             for(it = listMan2.WhiteList.begin();it !=listMan2.WhiteList.end();it++){
                //printf("view func first %d₩n",it->first);
                bwDatas = (listload::bwList)it->second;
                macCmpFlag = memcmp(&(bwDatas.apMac),&(mF->addr3),6);
                if(macCmpFlag == 0){

                //printf("view func datas %d %02x %02x %02x %02x %02x
%02x₩n",bwDatas.apCipher,bwDatas.apMac[0],bwDatas.apMac[1],bwDatas.apMac[2],bwDatas.
apMac[3],bwDatas.apMac[4],bwDatas.apMac[5]);
                    listSF.auth = bwDatas.apAuth;
                    listSF.enc = bwDatas.apEnc;
                    listSF.groupCipher = bwDatas.apCipher;
                    listSF.pairwiseCipher = bwDatas.apCipher;
                    cmpFlag = memcmp(&listSF,&sF,sizeof(SecurityFlag));
                    if(cmpFlag == 0){
                        printf("LISTED AP RULE!(WHITE)₩n");
                        WHTFlag = true;
                        break;
                        //passpkt
                    }
                }
            }

            cmpFlag = 0;
            macCmpFlag = 0;
            if(WHTFlag == false){
                printf("NOT LISTED AP RULE(WHITE)₩n");
                 //cmpFlag = 0;
                 //uint8_t cmpArray[6];
                 listload::bw_list::iterator it1;


                 //blk list!
                 for(it1 = listMan2.BlackList.begin();it1 !=listMan2.BlackList.end();it1++){
                     bwDatas = (listload::bwList)it1->second;
                     //printf("cmp with %02x %02x %02x %02x %02x
%02x₩n",bwDatas.apMac[0],bwDatas.apMac[1],bwDatas.apMac[2],bwDatas.apMac[3],bwDatas.
apMac[4],bwDatas.apMac[5]);
                     //printf("checking blk ap rule₩n");
                     macCmpFlag = memcmp(&(bwDatas.apMac),&(mF->addr3),6);
                     if(macCmpFlag == 0){

                         //printf("view func datas %d %02x %02x %02x %02x %02x
```

```
%02x₩n",bwDatas.apCipher,bwDatas.apMac[0],bwDatas.apMac[1],bwDatas.apMac[2],bwDatas.
apMac[3],bwDatas.apMac[4],bwDatas.apMac[5]);
                                listSF.auth = bwDatas.apAuth;
                                listSF.enc = bwDatas.apEnc;
                                listSF.groupCipher = bwDatas.apCipher;
                                listSF.pairwiseCipher = bwDatas.apCipher;
                                cmpFlag = memcmp(&listSF,&sF,sizeof(SecurityFlag));
                                if(cmpFlag == 0){
                                        printf("LISTED AP RULE!(BLACK)₩n");
                                        BLKFlag = true;
                                        break;
                                        //pass pkt
                                }
                        }


                }

            }

            if(WHTFlag == false & BLKFlag == false){
                        printf("NOT LISTED AP RULE(BLACK) ₩n");
                        storFlag = true;
                        char tempbuf[250] = {0,};
                        sprintf(tempbuf,"%s MISCONFIGURE AP",atkType);
                        sprintf(atkType,"%s",tempbuf);
                        //sprintf(blkBuff,"INSERT INTO TEST_BB (ap_mac,channel,block_stat)
VALUES
('%02x%02x%02x%02x%02x%02x',%d,%d)",exPkt.apMac[0],exPkt.apMac[1],exPkt.apMac[2],exP
kt.apMac[3],exPkt.apMac[4],exPkt.apMac[5],exPkt.channel,exPkt.blockStat);
                        printf("₩ndisordered!(AP RULE)₩n");
                        //start function(structure write and throw map,db)

            }

            //pkt data extract here
             //********************Initialization*******************
             memset(rH,0,sizeof(struct RadiotapHeader));
             memset(rF,0,sizeof(struct RadiotapHeaderFlag));
             memset(mF,0,sizeof(struct ManagementFrame));
             memset(fC,0,sizeof(struct FrameCtrl));
             memset(oF,0,sizeof(struct OptionField));
             memset(rsn,0,sizeof(struct Rsn));
             memset(vS,0,sizeof(struct VendorSpecific));
             return 0;

}


void usrfunc::test_viewFunc(listload& listMan2){
```

```
    // lLoad = listMan2;
     //listload::blk_list test = blkColl;
     listload::bw_list::iterator it;

     for(it = listMan2.BlackList.begin();it !=listMan2.BlackList.end();it++){
          //printf("view func first %d₩n",it->first);
          bwDatas = (listload::bwList)it->second;
          //printf("view func datas %d %02x %02x %02x %02x %02x
%02x₩n",bwDatas.apCipher,bwDatas.apMac[0],bwDatas.apMac[1],bwDatas.apMac[2],bwDatas.
apMac[3],bwDatas.apMac[4],bwDatas.apMac[5]);

     }


}


void usrfunc::macCmp(listload& listMan2){
        WHTFlag = false; //true in whitelisted packet
        BLKFlag = false; //true in blacklisted packet
        RadiotapHeader *RH = (RadiotapHeader*)(pktPoint);
        int length = RH->length;
        ManagementFrame *MF = (ManagementFrame*)(pktPoint+length);

       // int type = MF->frameCtrl.type;
       // int subtype = MF->frameCtrl.subType;

       // int ToDs = MF->frameCtrl.toDs;
        //int FromDs = MF->frameCtrl.fromDs;


        //if(type == 0 && subtype ==0) // Association Request Frame
        //if((ToDs == 0 && FromDs == 1) || (ToDs == 1 && FromDs == 0))
     //   {

//printf("=====================================================₩n")
;
            /* if(ToDs == 0 && FromDs == 1)
             {
                  printf("From AP₩n");
             }
             else
             {
                  printf("To AP₩n");
             }*/

             //printf("Sequnce   : %d₩n", (MF->seq));

             //printMac(1);      //DA
```

```
//printMac(2);      //SA
//printMac(3);      //BSS ID
int cmpFlag = 0;
//int macCmpFlag = 0;
//uint8_t cmpArray[6];
listload::bw_list::iterator it;
printf("-----------------------------------------------₩n");
printf("AP MAC CHECK₩n₩n");


//wht list!
for(it = listMan2.WhiteList.begin();it !=listMan2.WhiteList.end();it++){
    //printf("view func first %d₩n",it->first);
    bwDatas = (listload::bwList)it->second;
    printf("pkt compare with %02x %02x %02x %02x %02x
%02x₩n",bwDatas.apMac[0],bwDatas.apMac[1],bwDatas.apMac[2],bwDatas.apMac[3],bwDatas.
apMac[4],bwDatas.apMac[5]);
    printf("current pack bssid is %02x %02x %02x %02x %02x
%02x₩n",MF->addr3[0],MF->addr3[1],MF->addr3[2],MF->addr3[3],MF->addr3[4],MF->addr3
[5]);
    cmpFlag = memcmp(&(bwDatas.apMac),&(MF->addr3),6);
    if(cmpFlag == 0){
        printf("LISTED AP MAC!(WHITE)₩n");
        WHTFlag = true;
        break;
        //start function(structure write and throw map,db)
    }

}

if(WHTFlag == false){
    printf("NOT LISTED AP MAC(WHITE)₩n");
    cmpFlag = 0;
    //uint8_t cmpArray[6];
    listload::bw_list::iterator it1;


    //blk list!
    for(it1 = listMan2.BlackList.begin();it1 !=listMan2.BlackList.end();it1++){
        //printf("view func first %d₩n",it->first);
        bwDatas = (listload::bwList)it1->second;
        printf("cmp with %02x %02x %02x %02x %02x
%02x₩n",bwDatas.apMac[0],bwDatas.apMac[1],bwDatas.apMac[2],bwDatas.apMac[3],bwDatas.
apMac[4],bwDatas.apMac[5]);

        cmpFlag = memcmp(&(bwDatas.apMac),&(MF->addr3),6);
        if(cmpFlag == 0){
            printf("LISTED AP MAC!(BLACK)₩n");
            BLKFlag = true;
            break;
```

```
                        //already enrolled packet
                                //start function(structure write and throw map,db)
                    }

                }

            }
            if(WHTFlag == false && BLKFlag == false){
                    printf("₩ndisordered!(AP MAC)₩n₩n");
                    storFlag = true;
                    char tempbuf[250] = {0,};
                    sprintf(tempbuf,"%s UNDEFINED AP MAC",atkType);
                    sprintf(atkType,"%s",tempbuf);
                    //start function(structure write and throw map,db)

                }


            /*
            for(int i=0 ; i<6; i++)
            {
                int n = memcmp((char*)&MF->addr2[i], (char*)&MF->addr3[i], 6);
                if(n != 0)
                {
                    printf("different₩n");
                    break;
                }
                if(i==5)
                {
                    printf("same₩n");
                }
            }
           */
 }



//-----------------------------------------------------------------------------------------------------
------------------
void usrfunc::adhocFunc(listload& listMan2)
{
    WHTFlag = false; //true in whitelisted packet
    BLKFlag = false; //true in blacklisted packet
    //WHTFlag = false; //true in whitelisted packet
    //BLKFlag = false; //true in blacklisted packet
    Radiotap_Header *RD;
    RD = (Radiotap_Header *)pktPoint;
    Manage *MN;
    MN = (Manage *)((u_char *)pktPoint+(RD->Header_Length));
    //u_int16_t type1 = ntohs(MN->Duration);
```

```
    //u_int16_t type2 = ntohs(MN->Sequence);
    //u_int8_t frame_type = MN->Frame_control.Type;
    u_int8_t IBSS_Status = MN->Wireless_LAN.Capavility.IBSS;
    exPkt.adHocStat = IBSS_Status;
    //listMan2.bwStruct.adHocStat = IBSS_Status;

/*
    if (frame_type == 0){
        printf("-------------------------------------------------₩n");
        printf("Managemenet Frame₩n");
        printf("FramControl : %04x₩n", MN->Frame_control);
        printf("frame type : %02x₩n", frame_type);
        printf("Duration/ID Field : %04x₩n",type1);
        printf("DA : %02x-%02x-%02x-%02x-%02x-%02x₩n",
MN->des[0],MN->des[1],MN->des[2],MN->des[3],MN->des[4],MN->des[5]);
        printf("SA : %02x-%02x-%02x-%02x-%02x-%02x₩n",
MN->src[0],MN->src[1],MN->src[2],MN->src[3],MN->src[4],MN->src[5]);
        printf("BSS ID : %02x-%02x-%02x-%02x-%02x-%02x₩n",
MN->BSS[0],MN->BSS[1],MN->BSS[2],MN->BSS[3],MN->BSS[4],MN->BSS[5]);
        printf("Sequence : %04x₩n", type2);
        printf("Capability : %04x₩n", MN->Wireless_LAN.Capavility);
        printf("IBSS_satus : %01x₩n",IBSS_Status);
    }
*/
  //   if (IBSS_Status == 1)
    // printf("₩nAD-HOC Network₩n");
    int cmpFlag = 0;
    int macCmpFlag = 0;
    //uint8_t cmpArray[6];
    listload::bw_list::iterator it;

    printf("-------------------------------------------------₩n");
      //wht list!
    printf("CHECK IBSS₩n₩n");
     for(it = listMan2.WhiteList.begin();it !=listMan2.WhiteList.end();it++){
        //printf("view func first %d₩n",it->first);
        bwDatas = (listload::bwList)it->second;
        macCmpFlag = memcmp(&(bwDatas.apMac),&(MN->BSS),6);
        if(macCmpFlag == 0){
            printf("cmp ibss with %d₩n",bwDatas.adHocStat);

            cmpFlag = memcmp(&(bwDatas.adHocStat),&(IBSS_Status),1);
            if(cmpFlag == 0){
                printf("LISTED IBSS!(WHITE)₩n");
                WHTFlag = true;
                break;
                //alreay have white list OK.
            }
        }
    }
```

```
    cmpFlag = 0;
    macCmpFlag = 0;
    if(WHTFlag == false){
        printf("NOT LISTED IBSS(WHITE) ₩n");
        cmpFlag = 0;
        //uint8_t cmpArray[6];
        listload::bw_list::iterator it1;


                    //blk list!
        for(it1 = listMan2.BlackList.begin();it1 !=listMan2.BlackList.end();it1++){
            //printf("view func first %d₩n",it->first);
            bwDatas = (listload::bwList)it1->second;
            //printf("cmp with %02x %02x %02x %02x %02x
%02x₩n",bwDatas.apMac[0],bwDatas.apMac[1],bwDatas.apMac[2],bwDatas.apMac[3],bwDatas.
apMac[4],bwDatas.apMac[5]);
            //printf("check blk ibss₩n");
            macCmpFlag = memcmp(&(bwDatas.apMac),&(MN->BSS),6);
            if(macCmpFlag == 0){
                printf("cmp ibss with %d₩n",bwDatas.adHocStat);

                cmpFlag = memcmp(&(bwDatas.adHocStat),&(IBSS_Status),1);
                if(cmpFlag == 0){
                    printf("LISTED IBSS₩n");
                    BLKFlag = true;
                    break;
                    //alreay have black list OK.
                }
            }

        }

    }

    if(WHTFlag == false && BLKFlag == false){
        printf("NOT LISTED IBSS(BLACK) ₩n");
        printf("₩ndisordered!(ad_hoc)₩n₩n");
        storFlag = true;
        char tempbuf[250] = {0,};
        sprintf(tempbuf,"%s AD HOC",atkType);
        sprintf(atkType,"%s",tempbuf);
        //start function(structure write and throw map,db)

    }



}
```

```
void usrfunc::retMacAdr(void)
{
	switch(this->frameCtrl->toDs){
		case 0:
		{
			if(this->frameCtrl->fromDs == 0){//00
				memcpy(this->APMac,this->mgmtFrame->addr3,6);
				memcpy(this->cliMac,this->mgmtFrame->addr1,6);
			}else if(this->frameCtrl->fromDs == 1){
				memcpy(this->APMac,this->mgmtFrame->addr2,6);
				memcpy(this->cliMac,this->mgmtFrame->addr1,6);

			}

		}
		break;
		case 1:
		{
			if(this->frameCtrl->fromDs == 0){//00
				memcpy(this->APMac,this->mgmtFrame->addr1,6);
				memcpy(this->cliMac,this->mgmtFrame->addr3,6);
			}else if(this->frameCtrl->fromDs == 1){
				printf("11₩n");

			}
		}
		break;
	}
	}

void usrfunc::getCurPktData(listload& listMan2){
	radioGetData();//channel
	exPkt.blockStat = 1;
	exPkt.macType = 0;//beaconFrame


	memcpy(exPkt.apMac,mgmtFrame->addr3,6);
	memset(exPkt.stMac,0,6);//beaconFrame
	memset(atkType,0,sizeof(atkType));
	WHTFlag = false; //true in whitelisted packet
	BLKFlag = false;
	storFlag = false;

	/*
		int apAuth;
		int apCipher;
		int apEnc;
		uint8_t adHocStat;
```

```
            uint8_t ssid[50];
        */


}
void usrfunc::radioGetData(void){
        int pflgLth = 0;
        bool pflgChk = false;
        for(u_long i=0; i<3;i++){
            pflgChk = RTHeader->rth_present_flg.pflg1.test(i);
            //std::cout << pflgChk <<std::endl;
            if(pflgChk == true){
                pflgLth += PFrame.pflg_allign[i];
                //printf("current pFlg %d₩n",pflgLth);
            }

        }
        uint16_t HzChan = 0;
        memcpy(&HzChan,(pktPoint+RTHEADERSIZE+pflgLth),2);
        if(HzChan == 128){
            doFlag = true;
        }else{
            doFlag = false;
        }

        exPkt.channel = hzToCnl(HzChan);

}

void usrfunc::inputCurPkt(listload& listMan2,dbmanage& wipsDB2){
        printf("expkt channel %d₩n",exPkt.channel);
        char blkBuff[255];
        char logBuff[255];
        //exPkt;
        printf("INSERT INTO wips_black_blacklist (ap_mac,channel,blockstat) VALUES
('%02x%02x%02x%02x%02x%02x',%d,%d)₩n",exPkt.apMac[0],exPkt.apMac[1],exPkt.apMac[2],e
xPkt.apMac[3],exPkt.apMac[4],exPkt.apMac[5],exPkt.channel,exPkt.blockStat);
        sprintf(blkBuff,"INSERT INTO wips_black_blacklist
(ap_mac,channel,block_stat,ap_auth,ap_cipher,ap_enc,mac_type,ad_hoc_stat,ap_ssid) VALUES
('%02x%02x%02x%02x%02x%02x',%d,%d,%d,%d,%d,%d,%d,'%s')",exPkt.apMac[0],exPkt.apMac
[1],exPkt.apMac[2],exPkt.apMac[3],exPkt.apMac[4],exPkt.apMac[5],exPkt.channel,exPkt.blockSta
t,exPkt.apAuth,exPkt.apCipher,exPkt.apEnc,exPkt.macType,exPkt.adHocStat,exPkt.ssid);
        printf("send blk!! %s₩n",blkBuff);
        wipsDB2.dbQuery(blkBuff);
        sprintf(logBuff,"INSERT INTO wips_home_blocklog (mac,atk_type,block_stat) VALUES
('%02x%02x%02x%02x%02x%02x','%s',%d)",exPkt.apMac[0],exPkt.apMac[1],exPkt.apMac[2],exP
kt.apMac[3],exPkt.apMac[4],exPkt.apMac[5],atkType,exPkt.blockStat);
        printf("send log!! %s₩n",logBuff);
        wipsDB2.dbQuery(logBuff);
        //input data in map
```

```
	listMan2.blkCnt +=1;
	memset(&(listMan2.bwStruct),0,sizeof(listMan2.bwStruct));
	//memcpy(&(listMan2.bwStruct),&exPkt,sizeof(exPkt));
	listMan2.bwStruct.apAuth = exPkt.apAuth;
	listMan2.bwStruct.channel = exPkt.channel;
	listMan2.bwStruct.blockStat = exPkt.blockStat;
	listMan2.bwStruct.apCipher = exPkt.apCipher;
	listMan2.bwStruct.apEnc = exPkt.apEnc;
	listMan2.bwStruct.macType = exPkt.macType;
	listMan2.bwStruct.adHocStat = exPkt.adHocStat;
	memcpy(&(listMan2.bwStruct.apMac[0]),&(exPkt.apMac[0]),6);
	memcpy(&(listMan2.bwStruct.stMac[0]),&(exPkt.stMac[0]),6);
	memcpy(&(listMan2.bwStruct.ssid),&(exPkt.ssid),32);

	listMan2.BlackList.insert(std::make_pair(listMan2.blkCnt,listMan2.bwStruct));

	//change query
}
int usrfunc::hzToCnl(uint16_t recvhz){
	switch(recvhz) {
		case 2412:{
			return 1;
		}
		case 2417:{
			return 2;
		}
		case 2422:{
			return 3;
		}
		case 2427:{
			return 4;
		}
		case 2432:{
			return 5;
		}
		case 2437:{
			return 6;
		}
		case 2442:{
			return 7;
		}
		case 2447:{
			return 8;
		}
		case 2452:{
			return 9;
		}
		case 2457:{
			return 10;
		}
```

```
            case 2462:{
                return 11;
            }
            case 2467:{
                return 12;
            }
            case 2472:{
                return 13;
            }
            case 2482:{
                return 14;
            }
            default :
                break;


        }

}
```

[usrfunc.h] <check error>

```
#ifndef USRFUNC_H
#define USRFUNC_H
#pragma once
#include <arpa/inet.h>
#include <iostream>
#include <cstdint>

#include "packframes.h"
#include "aaa.h"
#include "listload.h"
#include "wonsang.h"
#include "bbb.h"
#include <map>

#define RTHEADERSIZE 12

class usrfunc
{
public:
    #pragma pack(push,1)
    packframes PFrame;
    uint8_t* pktPoint;
    packframes::rth* RTHeader;
    packframes::FC* frameCtrl;
    struct packframes::ManagementFrame *mgmtFrame;
    listload::bwList bwDatas;
    bool WHTFlag = false; //true in whitelisted packet
    bool BLKFlag = false; //true in blacklisted packet
```

```
bool storFlag = false;
uint8_t cliMac[6];
uint8_t APMac[6];

typedef struct currentPktData{

}curPktData;
#pragma pack(pop)
//listload* lLoad;
//listload t1t1;
bool doFlag = false;


typedef struct extPktInfo{
    uint8_t apMac[6];
    int channel;
    int blockStat;
    uint8_t stMac[6];
    int apAuth;
    int apCipher;
    int apEnc;
    int macType;
    uint8_t adHocStat;
    char ssid[32];


}extPktDatas;
extPktDatas exPkt;

char atkType[250] = {0,};


usrfunc(uint8_t *packet);
void fakeAp(listload& listMan2);

//---------------------------hyunseok
int Cipher(uint8_t cipher);
int Auth(uint8_t auth);
int misconfigureAP(listload& listMan2);
//----------------------------------------

void test_viewFunc(listload& listMan2);

//-----------------wonsang--------------
void macCmp(listload& listMan2);
//----------------------------------------

//-----------------minseok-------------
void adhocFunc(listload& listMan2);
// ---------------------------------------
```

```
    void retMacAdr(void);
    void getCurPktData(listload& listMan2);
    void radioGetData(void);
    void inputCurPkt(listload& listMan2,dbmanage& wipsDB2);
    int hzToCnl(uint16_t recvhz);

};

#endif // USRFUNC_H
```

[wonsang.h] <combine userFunc>

```
#ifndef WONSANG_H
#define WONSANG_H

#include <pcap.h>
#include <stdio.h>
#include <arpa/inet.h>
#include <string.h>

#pragma pack(push, 1)
/*struct RadiotapHeader
{
    uint8_t     reversion;
    uint8_t     pad;
    uint16_t    length;
};
typedef struct FrameCtrl
{
    uint8_t    protocolVer      : 2;
    uint8_t    type             : 2;
    uint8_t    subType          : 4;
    uint8_t    toDs             : 1;
    uint8_t    fromDs           : 1;
    uint8_t    moreFlag         : 1;
    uint8_t    retry            : 1;
    uint8_t    powerMgmt        : 1;
    uint8_t    moreData         : 1;
    uint8_t    protectedFrame : 1;
    uint8_t    order            : 1;
}FC;
struct ManagementFrame
{
            FC   frameCtrl;   //2 bytes
            uint16_t    duration;    //2 bytes
            uint8_t     addr1[6];    //6 bytes
            uint8_t     addr2[6];    //6 bytes
            uint8_t     addr3[6];    //6 bytes
            //uint16_t    seq_ctrl;    //2 bytes
```

```
                uint16_t     seq         : 12;     //12 bits
                uint8_t      fragment :   4;     // 4 bits
};*/
#pragma pack(pop)

void macCmp(const u_char *);
void printMac(const u_char *, int);

#endif // WONSANG_H
```

[index.html]

```
{% extends 'base.html' %}
{% load static %}
{% block content %}
<link
href='http://fonts.googleapis.com/css?family=Love+Ya+Like+A+Sister'rel='stylesheet'type='te
xt/css'>
<style type="text/css">
h3 {
      font-size: 87px;
      font-family: "Love Ya Like A Sister", cursive;
      color: white;
      background-color: black;
      position: absolute;
      height: 150px;
      width: 100%;
      z-index: 1;
      text-align: center;
      margin-bottom: 10px;
      padding: 30px;
      left: -15px;
      top: 1px;
}h5 {
      font-family: "Love Ya Like A Sister","Courier New";
      font-size: 30px;
      color: white;
      background-color: black;
      position: absolute;
      display: table-cell;
      text-align: center;
      padding-top: 40px;
      top: 1px;
      left: 10%;
      height: 150px;
      width: 12%;
      z-index: 2
}a {
      text-decoration: none;
      color: #333333;
```

```
}#topMenu {
	position: absolute;
	top: 160px;
	left: 15px;
	height: 50px;
	width: 100%;
}#topMenu ul li {
	list-style: none;
	color: white;
	background-color: #2d2d2d;
	float: left;
	width: 31.7%;
	line-height: 53px;
	vertical-align: middle;
	text-align: center;
	border-left: 1px solid white;
}#topMenu .menuLink {
	text-decoration: none;
	color: white;
	display: block;
	width: 100%;
	font-size: 26px;
	font-weight: bold;
	/*font-family: "Trebuchet MS", Dotum, Arial;*/
	font-family: "Love Ya Like A Sister","Courier New";
}#topMenu .menuLink:hover{
	color: red;
	background-color: #4d4d4d;
}#school {
	z-index: 3;
	position: absolute;
	top: 3px;
	left: 15px;
	width: 9%;
	height: 145px;
}footer {
	position: fixed;
	left: 0px;
	bottom: 0px;
	height: 30px;
	width: 100%;
	background: #5D5D5D;
	color: white;
	padding-top: 5px;
}footer p {
	text-align: center;
}#mainimg {
	top: 230px;
	left: 60px;
	position: absolute;
```

```
    width: 47%;
    height: 63%;
}button {
    position: absolute;
    top: 50px;
    float: right;
    right: 80px;
    z-index: 4;
    width: 100px;
    background-color: #f8585b;
    border: none;
    color: white;
    padding: 15px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 18px;
    font-weight: bold;
    border-radius: 10px;
    cursor: pointer;
}h4{
    position: relative;
    top: 260px;
    left: 55%;
    text-align: left;
    font-weight: bold;
    font-size: 30px;
    line-height: 1.7em;
    width: 40%;
    padding-bottom: 30px;
}</style>
    <div class="row" style="text-align: center;">
        <div class="col-sm-12">
            <h3>Packet Hunter</h3>
        </div>
        <div img>
        <img id="school" src="/static/aaaaa.png">
        </div>
        <div text><h5>Information<br/>Security</h5></div>
        <nav id ="topMenu">
            <ul>
                <li><a class="menuLink" href="{% url 'main' %} ">HOME</a></li>
                <li><a class="menuLink" href="{% url 'block_log' %}">Block
Logs</a></li>
                <li><a class="menuLink" href="{% url 'member' %}">Member</a></li>
            </ul>
        </nav>
        <button type="button"
onclick="location.href='http://localhost:8000/admin/login/?next=/admin/'
">관리자</button>
```

```
        <img id="mainimg" src="/static/bbb.PNG">
        <h4>- 무선네트워크 시장은 확대되고 사용률이 증가하는 반면,
<br/>  무선 네트워크에 대한 낮은 보안인식 때문에 <br/>  무선
공격에 대한 탐지 및 분석을 통해<br/>  실시간으로 감지하는 WIPS 시스템을
개발<br/><br/>- 허가되지 않은 무선네트워크공유 탐지<br/>- 허가되지않은 외부 AP 접속
탐지<br/>- 보안정책에 위반되는 보안설정 탐지<br/>- Fake AP, AP Mac
Spoofing탐지</h4>
        <footer>
            <div class="footer">
            <blockquote>
                <p>Copyright(c)2018 PacketHunter All rights reserved.</p>
            </blockquote>
        </footer>
        <!--<div class="col-sm-12">
            <p><a href="{% url 'block_log' %}">Block Logs</p>
            <p><a href="{% url 'white_list' %}">White list</p>
            <p><a href="{% url 'black_list' %}">Black list</p>
        </div>-->
    </div>
{% endblock %}
```

[block_log.html]

```
{% extends 'base.html' %}
{% load static %}
{% block content %}
<link
href='http://fonts.googleapis.com/css?family=Love+Ya+Like+A+Sister'rel='stylesheet'type='te
xt/css'>
<style type="text/css">
h3 {
    font-size: 87px;
    font-family: "Love Ya Like A Sister", cursive;
    color: white;
    background-color: black;
    position: absolute;
    height: 150px;
    width: 100%;
    z-index: 1;
    text-align: center;
    margin-bottom: 10px;
    padding: 30px;
    left: -15px;
    top: 1px;
}h5 {
    font-family: "Love Ya Like A Sister","Courier New";
    font-size: 30px;
    color: white;
    background-color: black;
    position: absolute;
```

```
	display: table-cell;
	text-align: center;
	padding-top: 40px;
	left: 10%;
	height: 150px;
	width: 12%;
	z-index: 2
}a {
	text-decoration: none;
	color: #333333;
}#topMenu {
	position: absolute;
	top: 160px;
	height: 50px;
	width: 100%;
}#topMenu ul li {
	list-style: none;
	color: white;
	background-color: #2d2d2d;
	float: left;
	width: 31.7%;
	line-height: 53px;
	vertical-align: middle;
	text-align: center;
	border-left: 1px solid white;
}#topMenu .menuLink {
	text-decoration: none;
	color: white;
	display: block;
	width: 100%;
	font-size: 26px;
	font-weight: bold;
	/*font-family: "Trebuchet MS", Dotum, Arial;*/
	font-family: "Love Ya Like A Sister","Courier New";
}#topMenu .menuLink:hover{
	color: red;
	background-color: #4d4d4d;
}#school {
	z-index: 3;
	position: absolute;
	top: 3px;
	width: 9%;
	height: 145px;
}table {
	position: absolute;
	width: 100px;
	top: 215px;
	left: 65px;
	text-align: center;
	padding-bottom: 30px;
```

```
}button {
    position: absolute;
    top: 50px;
    float: right;
    right: 80px;
    z-index: 4;
    width: 100px;
    background-color: #f8585b;
    border: none;
    color: white;
    padding: 15px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 18px;
    font-weight: bold;
    border-radius: 10px;
    cursor: pointer;
}footer {
    position: fixed;
    left: 0px;
    bottom: 0px;
    height: 30px;
    width: 100%;
    background: #5D5D5D;
    color: white;
    padding-top: 5px;
}footer p {
    text-align: center;
}</style>
    <div class="col-sm-12">
            <h3>Packet Hunter</h3>
        </div>
        <div img>
        <img id="school" src="/static/aaaaa.png">
        </div>
        <div text><h5>Information<br/>Security</h5></div>
        <nav id ="topMenu">
            <ul>
                <li><a class="menuLink" href="{% url 'main' %} ">HOME</a></li>
                <li><a class="menuLink" href="{% url 'block_log' %}">Block
Logs</a></li>
                <li><a class="menuLink" href="{% url 'member' %}">Member</a></li>
            </ul>
        </nav>
        <button type="button"
onclick="location.href='http://localhost:8000/admin/login/?next=/admin/'
">관리자</button>
    <table class="table col-sm-11">
        <thead>
```

```
            <tr>
                <th scope="col">number</th>
                <th scope="col">mac</th>
                <th scope="col">atk_type</th>
                <th scope="col">block_stat</th>
            </tr>
            </thead>
            <tbody>
            {% for row in wips_home_blocklog %}
                <tr>
                    <td>{{ row.number }}</td>
                    <td>{{ row.mac }}</td>
                    <td>{{ row.atk_type }}</td>
                    <td>{{ row.block_stat }}</td>
                </tr>
            {% endfor %}
            </tbody>
        </table>
            <footer>
                <div class="footer">
                <blockquote>
                    <p>Copyright(c)2018 PacketHunter All rights reserved.</p>
                </blockquote>
            </footer>
{% endblock %}
```

[member.html]

```
{% extends 'base.html' %}
{% load static %}
{% block content %}
<link
href='http://fonts.googleapis.com/css?family=Love+Ya+Like+A+Sister'rel='stylesheet'type='te
xt/css'>
<link href="https://fonts.googleapis.com/css?family=Source+Sans+Pro" rel="stylesheet">
<style type="text/css">
h3 {
    font-size: 87px;
    font-family: "Love Ya Like A Sister", cursive;
    color: white;
    background-color: black;
    position: absolute;
    height: 150px;
    width: 100%;
    z-index: 1;
    text-align: center;
    margin-bottom: 10px;
    padding: 30px;
    left: -15px;
    top: 1px;
```

```
}h5 {
	font-family: 'Source Sans Pro', sans-serif;
	font-size: 30px;
	color: white;
	background-color: black;
	position: absolute;
	display: table-cell;
	text-align: center;
	padding-top: 40px;
	left: 10%;
	height: 150px;
	width: 10%;
	z-index: 2
}a {
	text-decoration: none;
	color: #333333;
}#topMenu {
	position: absolute;
	top: 160px;
	height: 50px;
	width: 100%;
}#topMenu ul li {
	list-style: none;
	color: white;
	background-color: #2d2d2d;
	float: left;
	width: 31.7%;
	line-height: 53px;
	vertical-align: middle;
	text-align: center;
	border-left: 1px solid white;
}#topMenu .menuLink {
	text-decoration: none;
	color: white;
	display: block;
	width: 100%;
	font-size: 26px;
	font-weight: bold;
	/*font-family: "Trebuchet MS", Dotum, Arial;*/
	font-family: "Love Ya Like A Sister","Courier New";
}#topMenu .menuLink:hover{
	color: red;
	background-color: #4d4d4d;
}#school {
	z-index: 3;
	position: absolute;
	top: 3px;
	width: 9%;
	height: 145px;
}div.member_1 {
```

```
        position: relative;
        width: 50%;
        float: left;
        top: 230px;
        font-size: 30px;
}div.member_2 {
        position: relative;
        width: 50%;
        float: left;
        top: 230px;
        font-size: 30px;
}div.member_3 {
        position: relative;
        width: 50%;
        float: left;
        top: 260px;
        font-size: 30px;
}div.member_4 {
        position: relative;
        width: 50%;
        float: left;
        top: 260px;
        font-size: 30px;
}div.member_5 {
        position: relative;
        width: 50%;
        float: left;
        top: 290px;
        font-size: 30px;
}div.member_6 {
        position: relative;
        width: 50%;
        float: left;
        top: 290px;
        font-size: 30px;
}#memberimage {
        width: 200px;
        height: 200px;
}.col-md-6{
        position: absolutes;
        width: 50%;
        top: 230px;
        padding-bottom: 30px;
}h4 {
        position: absolute;
        font-size: 30px;
        top: 60px;
        left: 250px;
}footer {
        position: fixed;
```

```
    left: 0px;
    bottom: 0px;
    height: 30px;
    width: 100%;
    background: #5D5D5D;
    color: white;
    padding-top: 5px;
}footer p {
    text-align: center;
}button {
    position: absolute;
    top: 50px;
    float: right;
    right: 80px;
    z-index: 4;
    width: 100px;
    background-color: #f8585b;
    border: none;
    color: white;
    padding: 15px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 18px;
    font-weight: bold;
    border-radius: 10px;
    cursor: pointer;
}</style>
    <div class="col-sm-12">
            <h3>Packet Hunter</h3>
        </div>
        <div img>
        <img id="school" src="/static/aaaaa.png">
        </div>
        <div text><h5>Information<br/>Security</h5></div>
        <nav id ="topMenu">
            <ul>
                <li><a class="menuLink" href="{% url 'main' %} ">HOME</a></li>
                <li><a class="menuLink" href="{% url 'block_log' %}">Block
Logs</a></li>
                <li><a class="menuLink" href="{% url 'member' %}">Member</a></li>
            </ul>
        </nav>
        <button type="button"
onclick="location.href='http://localhost:8000/admin/login/?next=/admin/'
">관리자</button>
        <div>
            <div class="row">
                <div class="col-md-6"><img id="memberimage"
src="/static/40725741.jpg"><h4><strong>조장 신현석</strong><br/> - Mis-configured
```

```
AP 탐지</h4></div>
                    <div class="col-md-6"><img id="memberimage"
src="/static/40725741.jpg"><h4><strong>조원 김경수</strong><br/> - 방지시스템
구현</h4></div>
                    <div class="col-md-6"><img id="memberimage"
src="/static/40725741.jpg"><h4><strong>조원 박민석</strong><br/> - Ad-Hoc Connect
탐지</h4></div>
                    <div class="col-md-6"><img id="memberimage"
src="/static/40725741.jpg"><h4><strong>조원 유원상</strong><br/> - Client
Mis-association 탐지</h4></div>
                    <div class="col-md-6"><img id="memberimage"
src="/static/40725741.jpg"><h4><strong>조원 전경준</strong><br/> - Fake AP, AP Mac
Spoofing 탐지</h4></div>
                    <div class="col-md-6"><img id="memberimage"
src="/static/40725741.jpg"><h4><strong>조원 윤서완</strong><br/> - Log Viewer
구현<br/> - Web Interface[Front & Back End]</h4></div>
          <footer>
               <div class="footer">
               <blockquote>
                    <p>Copyright(c)2018 PacketHunter All rights reserved.</p>
               </blockquote>
          </footer>
{% endblock %}
-----------------------------------------------------------------------------------------------------
[member.html]

{% extends 'base.html' %}
{% load static %}
{% block content %}
<link
href='http://fonts.googleapis.com/css?family=Love+Ya+Like+A+Sister'rel='stylesheet'type='te
xt/css'>
<link href="https://fonts.googleapis.com/css?family=Source+Sans+Pro" rel="stylesheet">
<style type="text/css">
h3 {
     font-size: 87px;
     font-family: "Love Ya Like A Sister", cursive;
     color: white;
     background-color: black;
     position: absolute;
     height: 150px;
     width: 100%;
     z-index: 1;
     text-align: center;
     margin-bottom: 10px;
     padding: 30px;
     left: -15px;
     top: 1px;
}h5 {
     font-family: 'Source Sans Pro', sans-serif;
```

```
	font-size: 30px;
	color: white;
	background-color: black;
	position: absolute;
	display: table-cell;
	text-align: center;
	padding-top: 40px;
	left: 10%;
	height: 150px;
	width: 10%;
	z-index: 2
}a {
	text-decoration: none;
	color: #333333;
}#topMenu {
	position: absolute;
	top: 160px;
	height: 50px;
	width: 100%;
}#topMenu ul li {
	list-style: none;
	color: white;
	background-color: #2d2d2d;
	float: left;
	width: 31.7%;
	line-height: 53px;
	vertical-align: middle;
	text-align: center;
	border-left: 1px solid white;
}#topMenu .menuLink {
	text-decoration: none;
	color: white;
	display: block;
	width: 100%;
	font-size: 26px;
	font-weight: bold;
	/*font-family: "Trebuchet MS", Dotum, Arial;*/
	font-family: "Love Ya Like A Sister","Courier New";
}#topMenu .menuLink:hover{
	color: red;
	background-color: #4d4d4d;
}#school {
	z-index: 3;
	position: absolute;
	top: 3px;
	width: 9%;
	height: 145px;
}div.member_1 {
	position: relative;
	width: 50%;
```

```
        float: left;
        top: 230px;
        font-size: 30px;
}div.member_2 {
        position: relative;
        width: 50%;
        float: left;
        top: 230px;
        font-size: 30px;
}div.member_3 {
        position: relative;
        width: 50%;
        float: left;
        top: 260px;
        font-size: 30px;
}div.member_4 {
        position: relative;
        width: 50%;
        float: left;
        top: 260px;
        font-size: 30px;
}div.member_5 {
        position: relative;
        width: 50%;
        float: left;
        top: 290px;
        font-size: 30px;
}div.member_6 {
        position: relative;
        width: 50%;
        float: left;
        top: 290px;
        font-size: 30px;
}#memberimage {
        width: 200px;
        height: 200px;
}.col-md-6{
        position: absolutes;
        width: 50%;
        top: 230px;
        padding-bottom: 30px;
}h4 {
        position: absolute;
        font-size: 30px;
        top: 60px;
        left: 250px;
}footer {
        position: fixed;
        left: 0px;
        bottom: 0px;
```

```
    height: 30px;
    width: 100%;
    background: #5D5D5D;
    color: white;
    padding-top: 5px;
}footer p {
    text-align: center;
}button {
    position: absolute;
    top: 50px;
    float: right;
    right: 80px;
    z-index: 4;
    width: 100px;
    background-color: #f8585b;
    border: none;
    color: white;
    padding: 15px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 18px;
    font-weight: bold;
    border-radius: 10px;
    cursor: pointer;
}</style>
    <div class="col-sm-12">
            <h3>Packet Hunter</h3>
        </div>
        <div img>
        <img id="school" src="/static/aaaaa.png">
        </div>
        <div text><h5>Information<br/>Security</h5></div>
        <nav id ="topMenu">
            <ul>
                <li><a class="menuLink" href="{% url 'main' %} ">HOME</a></li>
                <li><a class="menuLink" href="{% url 'block_log' %}">Block
Logs</a></li>
                <li><a class="menuLink" href="{% url 'member' %}">Member</a></li>
            </ul>
        </nav>
        <button type="button"
onclick="location.href='http://localhost:8000/admin/login/?next=/admin/'
">관리자</button>
        <div>
            <div class="row">
                <div class="col-md-6"><img id="memberimage"
src="/static/40725741.jpg"><h4><strong>조장 신현석</strong><br/> - Mis-configured
AP 탐지</h4></div>
                <div class="col-md-6"><img id="memberimage"
```

```
src="/static/40725741.jpg"><h4><strong>조원 김경수</strong><br/> - 방지시스템
구현</h4></div>
                <div class="col-md-6"><img id="memberimage"
src="/static/40725741.jpg"><h4><strong>조원 박민석</strong><br/> - Ad-Hoc Connect
탐지</h4></div>
                <div class="col-md-6"><img id="memberimage"
src="/static/40725741.jpg"><h4><strong>조원 유원상</strong><br/> - Client
Mis-association 탐지</h4></div>
                <div class="col-md-6"><img id="memberimage"
src="/static/40725741.jpg"><h4><strong>조원 전경준</strong><br/> - Fake AP, AP Mac
Spoofing 탐지</h4></div>
                <div class="col-md-6"><img id="memberimage"
src="/static/40725741.jpg"><h4><strong>조원 윤서완</strong><br/> - Log Viewer
구현<br/> - Web Interface[Front & Back End]</h4></div>
        <footer>
            <div class="footer">
            <blockquote>
                <p>Copyright(c)2018 PacketHunter All rights reserved.</p>
            </blockquote>
        </footer>
{% endblock %}
```

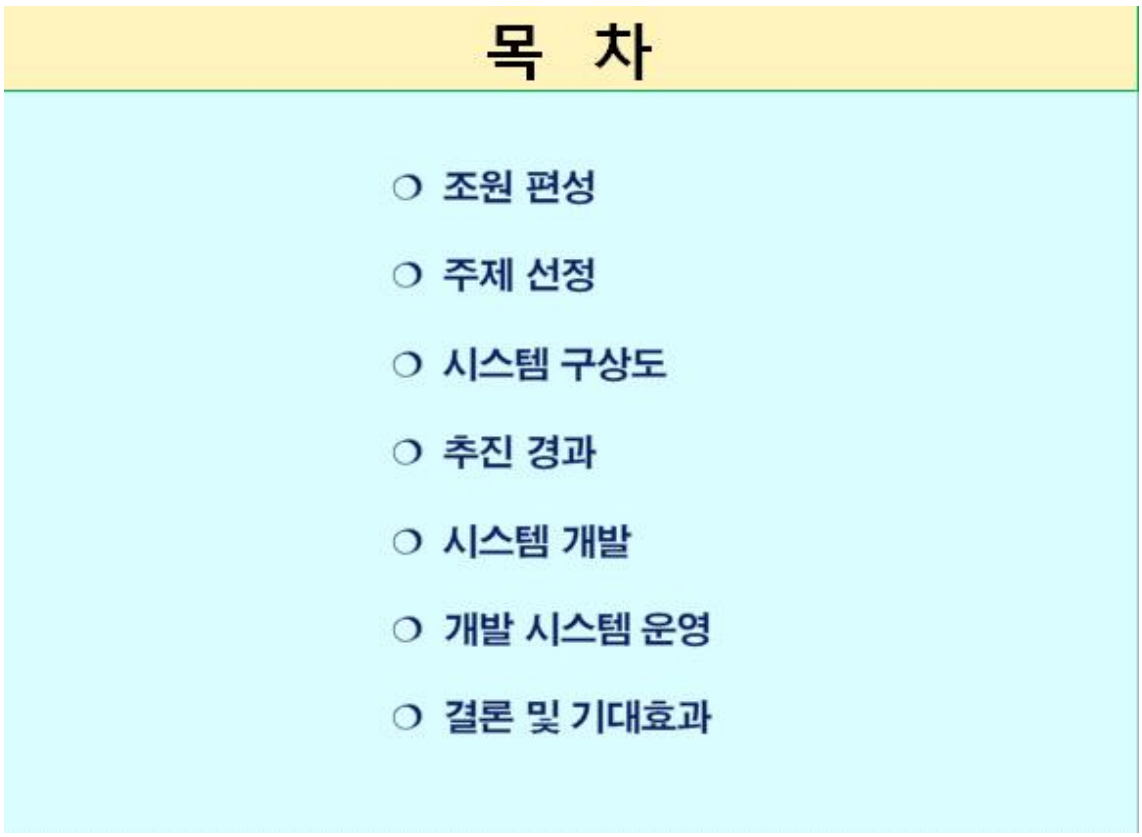
[base.html]

```
<!DOCTYPE html>
{% load staticfiles %}
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>PacketHunter</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdk
nLPMO" crossorigin="anonymous">
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49" crossorigin="anonymous">></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
integrity="sha384-ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy" crossorigin="anonymous">></script>
        <link rel="stylesheet" href="{% static '/static/css/index.css' %}">
        <style type="text/css">
            body {
                background-color: #FFFFFF;
            }
        </style>
</head>
```

```
<body>
    <div class="container-fluid">
        {% block content %}{% endblock %}
    </div>
</body>
</html>
```

5.2 발표 PPT

Wireless Intrusion
Detection & Prevention
WI-DP 침입탐지
및 방지시스템 구현
2018.11.7
정보보호학과
지도교수: 유승재 교수님
6조: 패킷사냥꾼
김경수 박민석 유원상
신현석 전경준 윤서완


목 차
❍ 조원 편성
❍ 주제 선정
❍ 시스템 구상도
❍ 추진 경과
❍ 시스템 개발
❍ 개발 시스템 운영
❍ 결론 및 기대효과

조원 편성

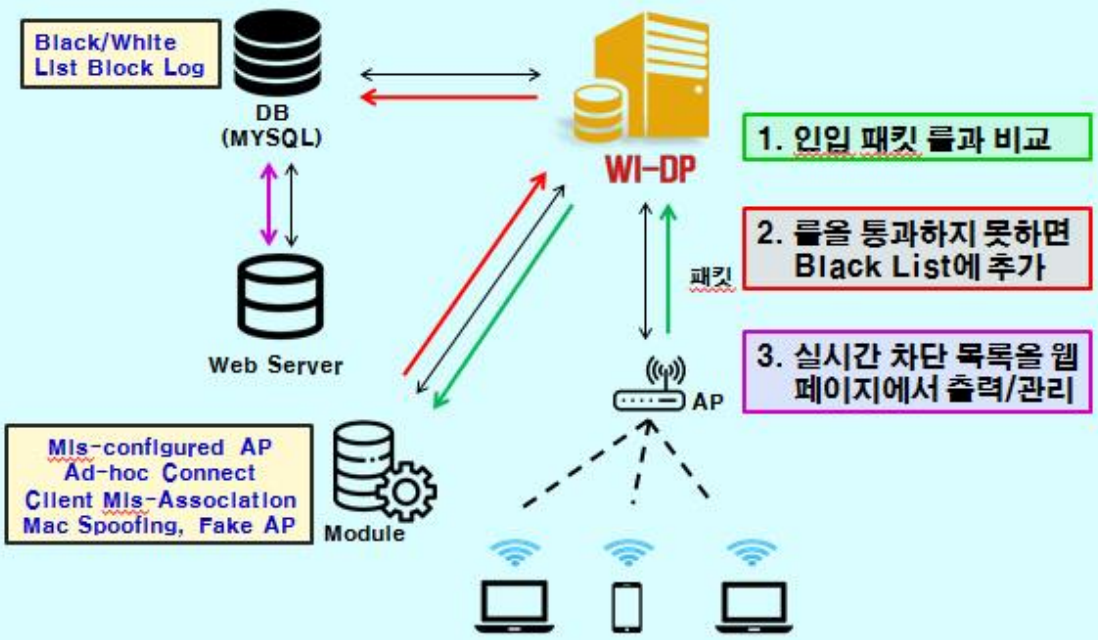
이 름	역 할
신현석(조장)	Mis-configured AP 탐지
김경수	방지시스템 구현
박민석	Ad-Hoc Connect 탐지
유원상	Client Mis-association 탐지
전경준	Fake AP, AP Mac Spoofing 탐지
윤서완	Log Viewer 구현 Web Interface (Front & Back End)

주제 선정

무선 네트워크 시장 확대 및 사용률 증가와 무선 네트워크에 대한 낮은 보안 인식

⇨ 무선 공격을 실시간으로 감지하는 WIPS 시스템을 개발, 안전한 네트워크 구축

시스템 구상도



추진 경과

기간 (2018년) / 작업	세부작업(월)	3	4	5	6	7	8	9	10
기획	자료수집								
Module 개발	Mis-configured AP								
	Ad-hoc Connect								
	Client Mis-association								
	Fake AP, AP Mac Spoofing								
Web 구축	HTML								
	django								
	DataBase								
종합	시험 및 보완								

시스템 개발(1/5)

개발 환경

시스템 개발(2/5)

요소 기술 개발(1/4)

❍ **Ad-Hoc 연결**

Ad hoc Connection

Ad-Hoc 네트워크 : P2P(Peer-To-Peer)라

Ad-hoc 탐지용 프로그램

```
void usrfunc::adhocFunc(listload& listMan2)
{
        WHTFlag = false; //true in whitelisted packet
        BLKFlag = false; //true in blacklisted packetd
//wht list!
        printf("CHECK IBSS\n\n");
        for(it = listMan2.WhiteList.begin();it !=listMan2.WhiteList.end();it++){
                bwDatas = (listload::bwList)it->second;
                 macCmpFlag = memcmp(&(bwDatas.apMac),&(MN->BSS),6);
                printf("MAP의 IBSS 값 %d 과 비교중\n",bwDatas.adHocStat);
                if(macCmpFlag == 0){
                        cmpFlag = memcmp(&(bwDatas.adHocStat),&(IBSS_Status),1);
                if(cmpFlag == 0){
                        printf("이미 저장된 IBSS!(WHITE)\n");
                        WHTFlag = true;
                        break;
```

~ 이하 생략 ~

시스템 개발(3/5)

요소 기술 개발(2/4)

❍ 비인가 AP 접속(Client Mis-Association)

Client Mis-association

인가된 클라이언트가 외부의 비인가 AP에 접속

비인가 AP 접속 탐지용 프로그램

```
void usrfunc::macCmp(listload& listMan2)
{
        WHTFlag = false; //true in whitelisted packet
        BLKFlag = false; //true in blacklisted packet
        RadiotapHeader *RH = (RadiotapHeader*)(pktPoint);
        int length = RH->length;
        ManagementFrame *MF = (ManagementFrame*)(pktPoint+length);
        int cmpFlag = 0;
        listload::bw_list::iterator it;
        printf("AP MAC CHECK\n\n");
//wht list!
        for(it = listMan2.WhiteList.begin();it !=listMan2.WhiteList.end();it++){
                bwDatas = (listload::bwList)it->second;
                printf("pkt compare with %02x %02x %02x %02x %02x %02x\n",bwDatas.a
                pMac[0],b wDatas.apMac[1],bwDatas.apMac[2],bwDatas.apMac[3],bwDat
                as.apMac[4],bwDatas.apMac[5]);
                        ~ 이하 생략 ~
```

시스템 개발(4/5)

요소 기술 개발(3/4)

❍ 보안정책 위반(Mis-Configured) AP

Mis-configured AP

암호화 미적용, WEP와 같은 약한 수준의 보안

보안정책 위반 탐지용 프로그램

```
int usrfunc::misconfigureAP(listload& listMan2)
{
                ~ 중략 ~
//*********************Flag  set******************
        if(sM.oUI[0] == 0x00 && sM.oUI[1] == 0x50 && sM.oUI[2] == 0xf2 &&
        sM.gCSS[0] == 0x00 && sM.gCSS[1] == 0x0f && sM.gCSS[2] == 0xac)//OUI 00-50-f2 &&
        OUI 00-0f-ac -> WPA2
        {
                //a = 20; //WPA-2 flsg:20
                sF.enc = WPA2;
                sF.groupCipher = Cipher(sM.gCSS[3]);
                sF.pairwiseCipher = Cipher(sM.pCSS[3]);
                sF.auth = Auth(sM.aSS[3]);
        }
        else if(sM.oUI[0] == 0x00 && sM.oUI[1] == 0x50 && sM.oUI[2] == 0xf2)//OUI 00-50-f2
        -> WPA-1
                        ~ 이하 생략 ~
```

시스템 개발(5/5)

요소 기술 개발(4/4)

❍ 불법복제 AP(Fake AP)

Fake AP

침입자는 기존의 무선 AP들과 이름이 같은

Fake AP 탐지용 프로그램

```
void usrfunc::fakeAp(listload& listMan2)
{
        WHTFlag = false; //true in whitelisted packet
        BLKFlag = false; //true in blacklisted packet
        uint8_t* pktPoint2 = this->pktPoint;
        printf("==========================================\n");
        printf("FAKE AP CHECK\n\n");
        struct packframes::WifiName *wifiName;
//BSSID*****************************************************************//
        printf("BSSID: ");
        for(int i=0; i<6; i++)
        {
                printf("%02x ", mgmtFrame->addr3[i]);
        }
        printf("\n");
//SSID******************************************************************//
        pktPoint2 += (RTHeader->rth_length + sizeof(struct packframes::ManagementFrame)
```

~ 이하 생략 ~

개발 시스템 운영(1/7)

시스템 구동

구동화면 1

```
File  Edit  View  Search  Terminal  Help
성공적으로 연결되었습니다.
BLACKLIST ENROLL
list enroll DB to MAP!

WHITELIST ENROLL
list enroll DB to MAP!

packets waiting...
=*=*=*=*=*=*=*=*=*=*=*=*=*=*PACKET CAPTURE*=*=*=*=*=

AP MAC CHECK

pkt compare with ac a3 1e 89 c3 e0
현재 패킷의 bssid  00 27 1c b7 9f 09
리스트에 없는 AP MAC(WHITE)

disordered!(AP MAC)

==========================================
CHECK IBSS

MAP의 IBSS 값 0 과 비교중
리스트에 없는 IBSS(WHITE)
리스트에 없는 IBSS(BLACK)

disordered!(ad-hoc)
==========
```

Ap MAC / Ad-hoc 체크

구동화면 2

```
==============================
FAKE AP CHECK

BSSID: 00 27 1c b7 9f 09
현재 패킷 AP SSID길이 = 9
LIST에 저장된 SSID JBU-WiFi와 비교중

리스트에 없는 SSID(WHITE)
리스트에 없는 SSID(BLACK)

disordered!(FAKE AP)

==========================================
AP RULE CHECK

AP MAC: 00:27:1c:b7:9f:09
-- 현재 잡힌 패킷의 정책 --
AP_AUTH = PSK 인증
AP_ENC = WPA2 사용
AP_CIPHER = CCMP 사용
----------------------
리스트에 없는 AP RULE(WHITE)
리스트에 없는 AP RULE(BLACK)

disordered!(AP RULE)
==========
```

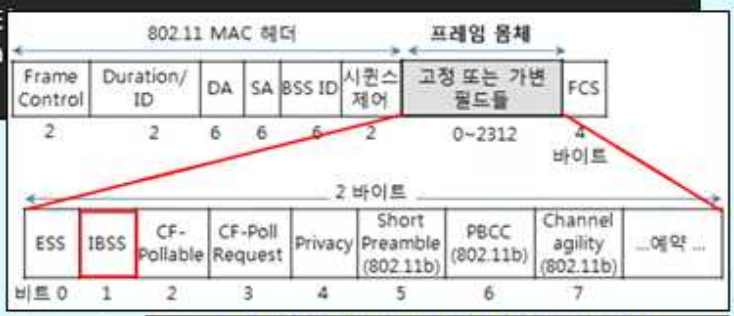
Fake Ap / Ap의 보안설정 체크

개발 시스템 운영(2/7)

탐지 1 : Ad-hoc 네트워크 탐지

Ad-hoc 탐지

```
CHECK IBSS

cmp ibss with 0
LISTED IBSS!(WHITE
BSSID: ac a3 1e 8a
cur leng is 8
```



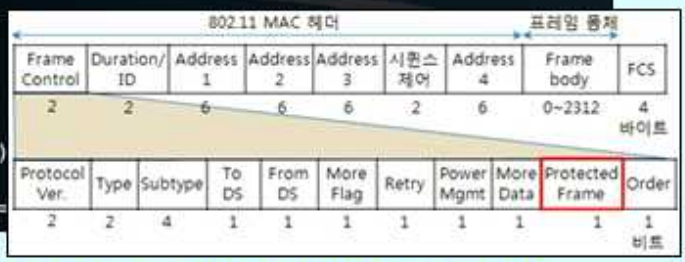
프레임 몸체의 IBSS 값이 1일 경우 Ad-Hoc 네트워크로 판별

개발 시스템 운영(3/7)

탐지 2 : 보안정책 위반 탐지

보안 정책 위반 탐지

```
=*=*=*=*=*=*=*=*=*=*=*=*=*=*PACKET CAPTURE*=*=*=*=*=
=========================================
AP RULE CHECK

AP MAC: 00:27:1c:b7:9f:09
-- 현재 잡힌 패킷의 정책 --
AP_AUTH = PSK 인증
AP_ENC = WPA2 사용
AP_CIPHER = CCMP 사용
-----------------------
리스트에 없는 AP RULE(WHITE)
이미 저장된 AP RULE!(BLACK)
=========================
```



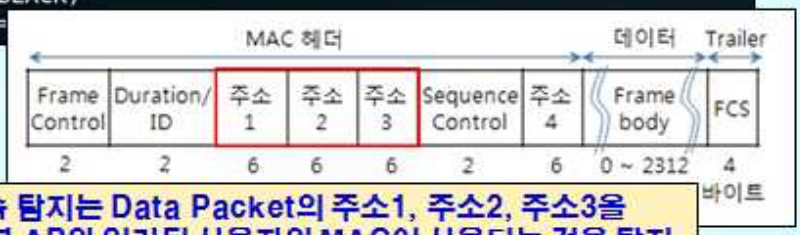
Protected Frame 필드가 0 이면 OPEN이고, 1이면 WEP 설정으로 보안상 취약하므로 보안정책 위반으로 판별

개발 시스템 운영(4/7)

탐지 3 : 외부 AP 접속 탐지

외부 AP 접속 탐지

```
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*PACKET CAPTURE*=*=*=*=*=*=*

AP MAC CHECK

pkt compare with ac a3 1e 89 c3 e0
현재 패킷의 bssid  70 25 59 16 9e 20
리스트에 없는 AP MAC(WHITE)
cmp with 00 27 1c b7 9f 09
cmp with 70 25 59 16 9e 20
이미 저장된 AP MAC!(BLACK)
=====================
```



외부 AP 접속 탐지는 Data Packet의 주소1, 주소2, 주소3을 분석하여 외부 AP와 인가된 사용자의 MAC이 사용되는 것을 탐지

개발 시스템 운영(5/7)

탐지 4 : Fake AP 탐지

Fake Ap 탐지

```
=========================================
FAKE AP CHECK

BSSID: 18 c5 01 10 d3 86
현재 패킷 AP SSID길이 = 9
LIST에 저장된 SSID JBU-WiFi

리스트에 없는 SSID(WHITE)
cmp ssid U+NetD387
이미 저장된 SSID(BLACK)
```

1. 패킷의 SSID와 일치하는 화이트리스트상의 AP의 SSID가 있는지 확인
2. 동일한 SSID가 존재하는 경우, 패킷의 BSS ID와 해당 AP의 MAC 주소를 비교
3. 패킷의 BSSID와 AP의 MAC 주소가 일치하지 않을 때 Fake AP 공격발생 판단

개발 시스템 운영(6/7)

관리자 페이지

관리자 페이지에서 Black / White 리스트
내용 등의 데이터베이스 값 조회 및 수정이 가능

개발 시스템 운영(7/7)

웹 페이지

WI-DP에 의해 차단된 무선공격 패킷의 로그 확인 가능

결론 및 기대효과

❍ 침입탐지 및 방어시스템 개발 성과

- 무선 공격에 대한 위협 탐지 및 차단 기능을 제공하고, 화이트/블랙 리스트 관리의 편리성을 높이며 차단패킷 정보 확인이 가능

- 또한 무선 패킷분석 및 취약점 점검도구로 활용이 가능하며 모듈화로 프로그램 확장 및 재사용 용이

❍ 기대 효과 및 교훈

- 이 시스템 운영 시 보안이 대폭 강화된 무선 네트워크 환경 구축이 가능할 것으로 기대

- 무선 프로토콜 학습 및 소프트웨어 개발 경험을 통해 이 분야 실무 기술 역량을 배양하고 팀워크의 중요성을 체험하는 기회

감사합니다.

Q & A

5.3 참고문헌

출 처	
위키피디아	https://ko.wikipedia.org/wiki/%EC%9E%A5%EA%B3%A0_(%EC%9B%B9_%ED%94%84%EB%A0%88%EC%9E%84%EC%9B%8C%ED%81%AC)
Django란 무엇인가?	http://stickie.tistory.com/10
802.11 MAC 프레임[정보통신기술용어해설]	http://www.ktword.co.kr/abbr_view.php?m_temp1=3352
기술자료- 무선랜 802.11 프로토콜 구조	http://mrbas.co.kr/xe/board_Txkm56/77205
[암호화/인증] 802.11 Sniffer Capture Analysis - WPA/WPA2 with PSK or EAP	http://fmyson.tistory.com/100
Hitch Hiker's Guide to Learning Learn on the Go WLAN. C Programming and Linux	http://www.hitchhikersguidetolearning.com/2017/09/17/rsn-information-element/
What Is IEEE 802.11i?	http://etutorials.org/Networking/802.11+security.+wi-fi+protected+access+and+802.11i/Part+II+The+Design+of+Wi-Fi+Security/Chapter+7.+WPA+RSN+and+IEEE+802.11i/What+Is+IEEE+802.11i/
불법AP 탐지 위협 예방을 위한 무선침입방지 시스템의 구축 (이기혁, 윤재동)	한국통신학회 학술대회논문집
무선 구간 모니터링을 통한 Rogue AP 탐지 시스템 (김동필, 김상욱)	한국통신학회 학술대회 및 강연회
MSDN STL MAP 클래스	https://msdn.microsoft.com/ko-kr/library/