

Laboratory 2: Unauthorized Access in Wireless Networks

Niclas Scheuing and Vasileios Dimitrakis

November 6, 2015

1 Introduction

In this laboratory exercise, we learn about and experiment on the weakness in various wireless security mechanisms. More specifically, in the first chapter, we hack the MAC filtering, in the second one, we crack the WEP encryption and in the last part, we break the WPA2 Personal Passwords.

2 Materials and Methods

Throughout the lab exercise 2 we used different materials and methods, that are presented below:

- *ifconfig*: Is used to configure the network interfaces.
- *hostapd*: Is a user space daemon for Access Point and authentication servers.
- *iwconfig*: Is used to configure the wireless network interfaces.
- *iwlist*: Is used to display some additional information from a wireless network interface that is not displayed by *iwconfig*
- *wireshark*: Is an open source packet analyser.
- *macchanger*: Is a Linux command that changes the MAC address of a network interface.
- *airodump-ng*: Is used for packet capturing of raw 802.11 frames and is particularly suitable for collecting WEP IVs (Initialization Vector) for the intent of using them with *aircrack-ng*.

- *aircrack-ng*: Is an 802.11 WEP and WPA-PSK keys cracking program that can recover keys once enough data packets have been captured.
- *ping*: Is a computer network administration software utility used to test the reachability of a host on an Internet Protocol (IP) network and to measure the round-trip time for messages sent from the originating host to a destination computer and back.
- *wpa_supplicant*: Is a cross-platform supplicant with support for WEP, WPA and WPA2.
- *aireplay-ng*: Inject ARP-request packets into a wireless network to generate traffic.

3 Terminology

The machine serving as wireless nodes is denoted by *User* and the machine used to attack the communication by *Attacker*. The machine used as access point is called *AP* in the following.

4 Experiments

In the following we will go through the experiments one by one and introduce the corresponding theory, setup and results.

4.1 Hacking MAC Filtering

MAC filtering is a security technique to prevent unauthorized users from accessing a wireless network. All network devices have a unique 48bit *MAC* address. The access-point grants or denies access to devices based on the *MAC* address communicated by the device itself. This is why a device can spoof the *MAC* address. For filtering black-lists and white-lists are used, granting access to devices contained in the white-list and denying access to those in the black-list.

4.1.1 Running the Experiment

AP used a white-list containing the *User's MAC* address, but not the *Attacker's*. The *AP* and the *User* were transmitting data. The *Attacker* was not able to connect to the *AP*, because he was not on the white-list.

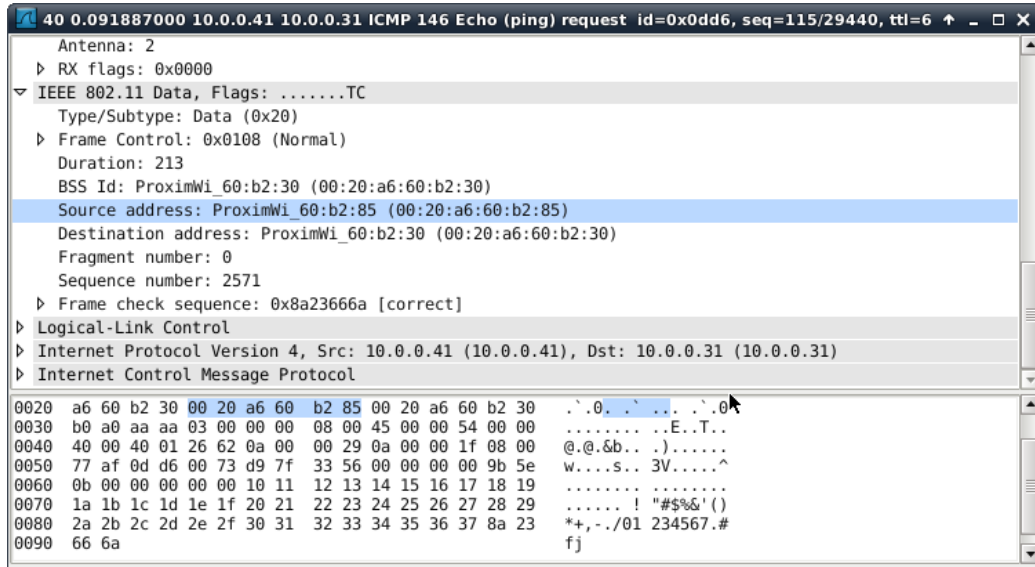


Figure 1: Finding the *User's* MAC address using *Wireshark*.

To find the channel and *AP's* MAC, the *Attacker* used the *iwlist* tool. The *Attacker* is observing the communication running its Wifi adapter in *monitor* mode and capturing the observed traffic with *Wireshark*. This way he was able to extract the *User's* MAC. See Figure 1

Knowing this MAC, the *Attacker* sets his own MAC address to the *User's* address using the *macchanger* tool. See Figure 2. The *Attacker* *User* were then able to connect to the *AP*.

4.1.2 Analysis

MAC filtering can easily be bypassed with some limitations. The *Attacker* does not get his own MAC address, but share it with the *User*, which leads to undesired side effects, such as both receiving the same packages. But it is very easily implemented and in wired networks it is harder for the *Attacker* to sniff valid MAC addresses. Also it does not introduce additional overhead which would reduce the networks throughput.

4.2 Cracking WEP Encryption

WEP is security mechanism for wireless networks using a RC4 encryption with a 40bit *secret* concatenated with a 24bit initialization vector. This key is usually represented as a string of 10 hexadecimal values. Besides the initialization vector, there is no additional randomness in the key used for

```
Terminal - station4@station4: ~
File Edit View Terminal Go Help

Channel:112
Frequency:5.56 GHz (Channel 112)
Quality=21/70 Signal level=-89 dBm
Encryption key:off
ESSID:"public-5"
Bit Rates:6 Mb/s; 9 Mb/s; 12 Mb/s; 18 Mb/s; 24 Mb/s
          36 Mb/s; 48 Mb/s; 54 Mb/s
Mode:Master
Extra:tsf=0000002780a5991b0
Extra: Last beacon: 1532ms ago
IE: Unknown: 00087075626C69632D35
IE: Unknown: 01088C129824B048606C
IE: Unknown: 050400010000
IE: Unknown: 070C43484924080F640510840310
IE: Unknown: 2D1AEE1103FFFFFF000000000000000000000000000000000000
0000000000
0
22F00
00000000000000000000
0000000000
IE: Unknown: 3D16700F0400000000000000000000000000000000000000000000
IE: Unknown: DD180050F2020101800003A4000027A4000042435E00623
IE: Unknown: DD1E00904C33EE1103FFFFFF0000000000000000000000000000
IE: Unknown: DD1A00904C34700F040000000000000000000000000000000000
root@station4:/home/station4# ifconfig wlan0 down
root@station4:/home/station4# macchanger wlan0 -m 00:20:a6:60:b2:85
Permanent MAC: 00:20:a6:60:b2:8c (Proxim Wireless)
Current MAC: 00:20:a6:60:b2:8c (Proxim Wireless)
New MAC: 00:20:a6:60:b2:85 (Proxim Wireless)
root@station4:/home/station4#
```

Figure 2: Changing the *Attacker's* MAC address using *macchanger*.

the encryption, because the *secret* remains the same over all transmissions. 24bit is short enough to be brute-forced.

4.2.1 Running the Experiment

User1 and *User2* connected in ad-hoc mode using a WEP encryption. At first they used a different *secret* each, and observed, they were not able to connect. Then they used the same *secret* and could connect.

The *Attacker* who did not set any key, could not connect. The *Attacker* did observe the network using the monitor mode and wireshark. The packets sniffed using wireshark can be seen in Figure 3. It was a flood of ping messages, but encrypted with the WEP's RC4 encryption.

The *Attacker* then started the *airodump-ng* to collect the traffic and *aircrack-ng* tool to crack the key. See Figure 4. It had to capture 38989 initialization vectors and based on that the tool had to try 236 keys to successfully guess the correct one. This took 7 seconds. The *Attacker* could then join the network using the found key.

This attack requires a few thousand captured packets. Depending on the network activity, this will take from a few seconds up to 10 minutes.

Observing the traffic in our home network generated by a file server, a multimedia station and a notebook running, we got about 5000 packets per minute. So the attack would take a few minutes.

In a office network with more traffic, this time would be significantly shorter.

4.3 Breaking WPA2 Personal Passwords

4.3.1 Running the Experiment

5 Analysis

References

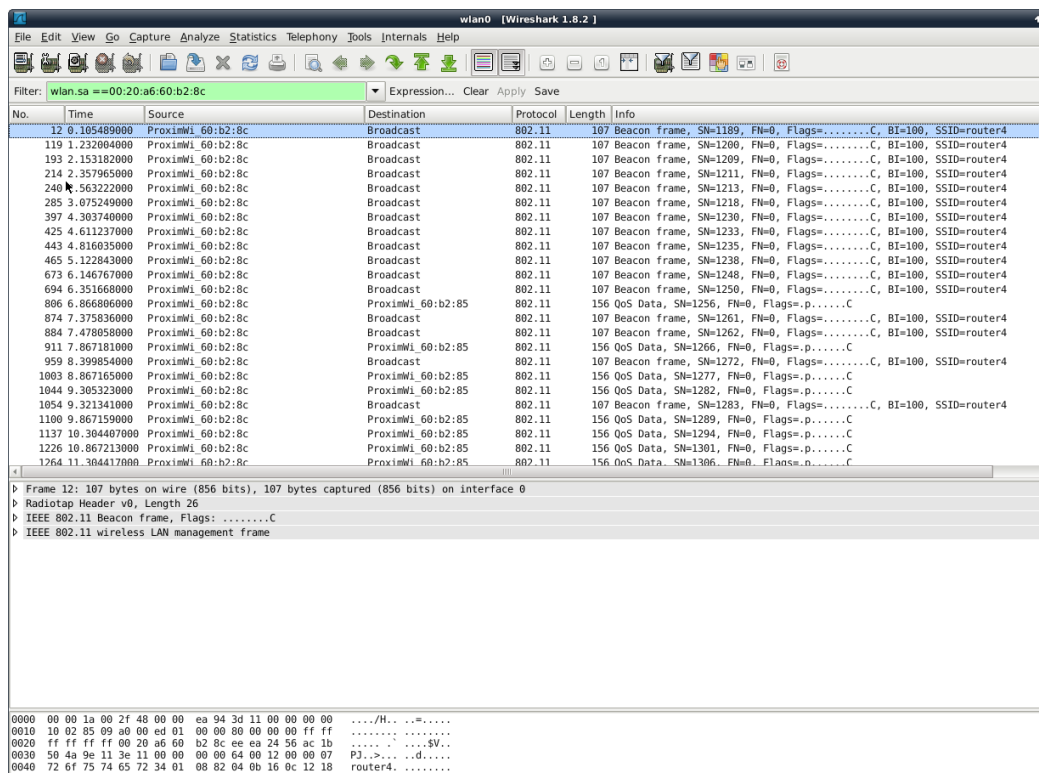


Figure 3: Capturing the traffic using source *MAC* address filtering.

```
Terminal - station3@station3: ~
File Edit View Terminal Go Help

Aircrack-ng 1.2 beta1

[00:00:07] Tested 236 keys (got 38989 IVs)

KB    depth  byte(vote)
0      0/ 1    FA(56064) 2C(46848) 0E(46592) 75(46592) 85(46592)
1      0/ 13    CE(48640) FF(48384) 24(47104) 6F(46336) B4(46080)
2      0/ 1    D1(53504) AF(47872) 07(47104) B0(46848) 6E(46592)
3     15/ 19    DD(44544) 52(44032) 77(44032) BC(44032) 18(43776)
4      0/ 1    45(53760) DA(49408) F8(48384) E1(46592) 89(45568)

KEY FOUND! [ FA:CE:D1:23:45 ]
StartingDecrypted correctly: 100%.

root@station3:/home/station3#
```

Figure 4: Cracking the WEP key using *aircrack-ng*