

Laboratory 2: Unauthorized Access in Wireless Networks

Niclas Scheuing and Vasileios Dimitrakis

November 12, 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*.^[1]

- *aircrack-ng*: Is an 802.11 WEP and WPA-PSK keys cracking program that can recover keys once enough data packets have been captured. [1]
- *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.[1]

3 Terminology

The machine serving as wireless node 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* one. 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.

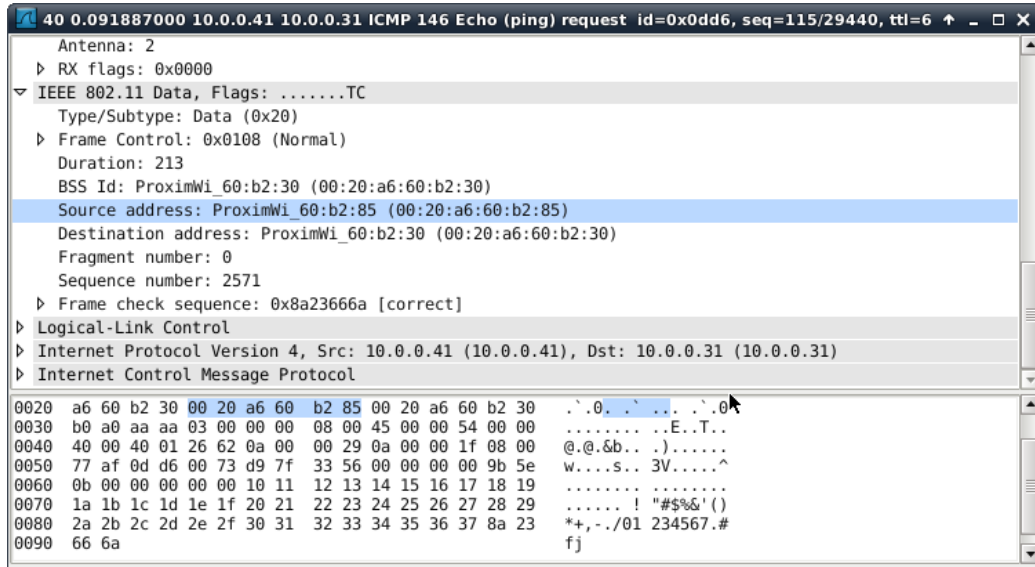


Figure 1: Finding the *User's* MAC address using *Wireshark*.

```

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*.

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

Knowing this MAC address, the *Attacker* set 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. Although, it is very easily implemented and it does not introduce additional overhead, which would reduce the networks throughput. Moreover, although it has

this severe security flaw, nowadays it is used in wired networks, because it is harder for the *Attacker* to sniff valid *MAC* addresses. Furthermore, it can be used in small WLANs, where the users trust one another.

When doing the measurements of our Cantenna, we used *MAC* filtering with a white-list as a simple security mechanism to disallow other devices to simply connect to our access point. Only little computational and little data overhead was introduced and for this setting the amount of security was enough.¹

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 the encryption, because the *secret* remains the same over all transmissions. 24bit is short enough to be brute-forced.[2]

4.2.1 Running the Experiment

User1 and *User2* were 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 establish a connection.

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.

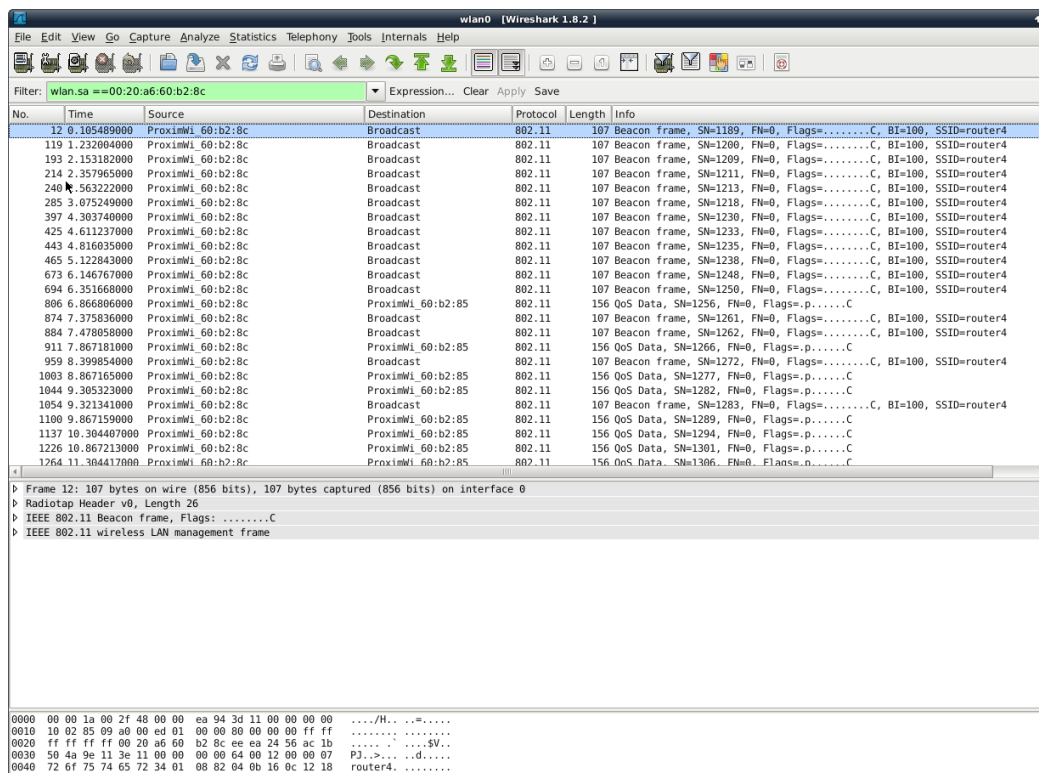


Figure 3: Capturing the traffic using source *MAC* address filtering.

```
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*

4.3 Breaking WPA2 Personal Passwords

To deal with the weaknesses of WEP, WPA was introduced. WPA2 provides a mode using AES encryption with a 256bit key. It can be run in the *personal mode* using pre-shared keys or the *enterprise mode* using authentication protocols.

To establish a session key, the WPA2 protocol performs a 4-way handshake between the access point and the client. During this a *Message Integrity Check (MIC)* is computed and transmitted. The computation of the MIC uses the pre-shared *Pairwise Master Key (PMK)*. An attacker can capture this handshake and break the MIC by brute-forcing the PMK.

4.3.1 Running the Experiment

AP and *User* are connected using a WPA2-PSK-AES setup in infrastructure mode. The key used was *motorbike*, a word which is part of most dictionaries used for cracking passwords.

The *Attacker* was not able to connect without knowing the password, but could observe the encrypted traffic using Wireshark.

To get the *MIC*, the *Attacker* performed a de-authentication attack by sending de-authentication packets with *aireplay-ng*. This caused the *AP* and *User* to reestablish the connection by performing the 4-way handshake. The handshake was captured by the *Attacker* using *airodump-ng*. *aircrack-ng* cracked the MIC using the captured handshake and a dictionary.

The machine the attack was performed could process *3535.59 keys per second* or *212135.4 keys per minute*. It took 10 seconds to find the password.

This machine could perform *305 Mkeys per day*. Since $\lg(305000000) \approx 28$, the a key with more than 28 bits could not be cracked within a day.

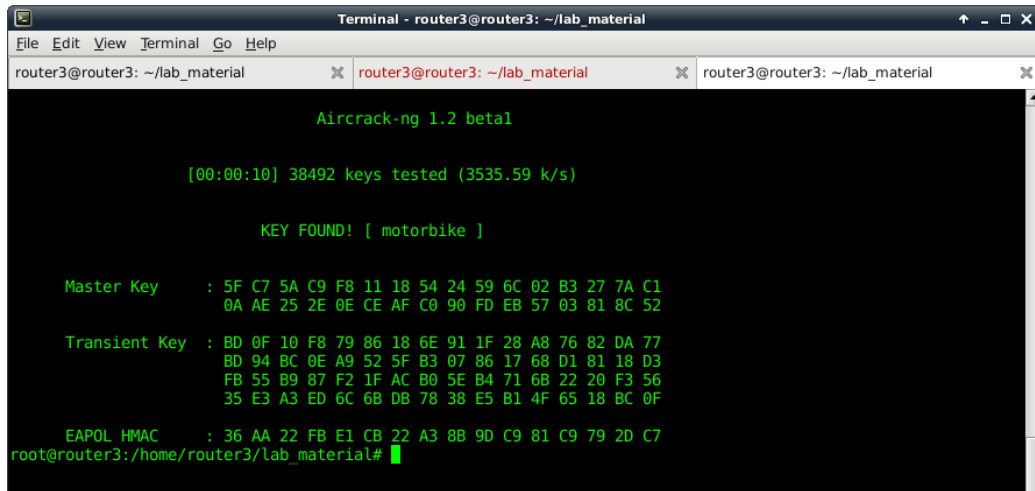
This attack was possible, because the password was contained in the dictionary. Using a more random password would have prevented a simple dictionary attack.

if password is long or not in dictionary, it is not possible for the attacker to retrieve the password.

5 Analysis

In this lab we demonstrated that some widely used security mechanisms as MAC-filtering, WEP and WPA2 are prone to rather simple attacks.

- *MAC filtering*: As we mentioned in the introduction, this security technique is quite simple, but it can be overpassed easily enough. For this



```
Terminal - router3@router3: ~/lab_material
router3@router3: ~/lab_material
Aircrack-ng 1.2 beta1

[00:00:10] 38492 keys tested (3535.59 k/s)

KEY FOUND! [ motorbike ]

Master Key   : 5F C7 5A C9 F8 11 18 54 24 59 6C 02 B3 27 7A C1
              0A AE 25 2E 0E CE AF C0 90 FD EB 57 03 81 8C 52

Transient Key : BD 0F 10 F8 79 86 18 6E 91 1F 28 A8 76 82 DA 77
              BD 94 BC 0E A9 52 5F B3 07 86 17 68 D1 81 18 D3
              FB 55 B9 87 F2 1F AC B0 5E B4 71 6B 22 20 F3 56
              35 E3 A3 ED 6C 6B DB 78 38 E5 B1 4F 65 18 BC 0F

EAPOL HMAC   : 36 AA 22 FB E1 CB 22 A3 8B 9D C9 81 C9 79 2D C7
root@router3:/home/router3/lab_material#
```

Figure 5: Cracked the WPA2 password with *aircrack-ng*.

reason, nowadays it is not often used and the only cases that is implemented are in wired networks and small WLANs, where the packet sniffing is difficult and there is mutual trust between the users, respectively.

- *WEP encryption*: This security method is also quite simple, but it offers weak protection and it is very easily exploited, due to specific inherent vulnerabilities of the protocol, such as small length of the IVs.
- *WPA2 encryption*: WPA2 personal is the most secure technique of the other two ones. It uses block cipher for the encryption mode and this increases the process time. Although, WPA2 personal seems to be secure, it is susceptible to dictionary attacks, which can be avoided, if the user chooses a stronger password.

References

- [1] Aircrack-ng. Aircrack-ng reference manual. <http://www.aircrack-ng.org>, 2015.
- [2] Techwriters Future. Wep - wired equivalent privacy. <http://ipv6.com/articles/wireless/Wired-Equivalent-Privacy.htm>, 2015.