Information and Organisational Security

Guides for Practical Classes

João Paulo Barraca and Vitor Cunha

Department of Electronics, Telecommunications and Informatics
University of Aveiro

2019–2020

2

Attacks to WEP and WPA2

2.1 Introduction

The objective of this guide is to demonstrate in practice, the security limitations of a wireless network that uses WEP or WPA2. For this purpose it will be required to use a Linux system, with the required attack tools.

The attack will be conducted against a vulnerable Access Point (AP), dedicated to the laboratory. Other access points may be used, but the results may be different from the ones described. Also, due to differences in the hardware of each laptop, it may be required to do some further steps or change some aspects of this guide. Whenever possible, consult the documentation of the software used **and request help from your professor**.

2.2 Installation and Setup

If you have a native Linux installation, and a laptop with Linux, you can use this setup directly. Some versions of Windows are also supported with different rates of success.

Independently of the operating system (Linux, OS X or Windows), the following software will be required:

- Bettercap-ng: A swiss army knife for 802.11, available here: `https://www.bettercap.org/`
- Aircrack-ng: A suite to assess Wifi network security `https://www.aircrack-ng.org/`

- Hashcat: (Optional) A password recovery software `https://hashcat.net/hashcat/`
- John The Ripper: (Optional) An alternative to Hashcat. Available here: `https://www.openwall.com/john/`
- HashSuite: (Optional) Yet another alternative to Hashcat and John for Windows. Available here: `https://hashsuite.openwall.net/`

2.3 WEP Attacks

2.3.1 Monitor mode

The attack requires the attacker to configure his WiFi interface in **monitor** mode. In this mode, the WiFi interface will not be able to send any standard packet, but it will be able to receive all frames sent in a WiFi network, including the IEEE 802.11 headers, from all stations in the same channel. The WiFi interface will not be associated to any Access Point, and will receive packets from all devices. Some drivers also allow to do packet injection.

Execute the following command:

```
$ iwconfig
```

and analyze the WiFi interfaces available in your computer, including its mode of operation (usually it is `Managed`). Run the following command[1], replacing `DEVNAME` with the name of the WiFi device(s).

```
airmon-ng start DEVNAME
```

Execute the `iwconfig` again and check the result. You should see the existence of a new interface (something like `mon0`), operating in the `Monitor` mode. This is the interface to use in subsequent steps.

2.3.2 Detect the victim AP or network

This step consists in the identification of the operational characteristics of the victim. The immediate victim will be the AP, but if it is compromised, the attacker will have access to all hosts currently associated to it, and the unprotected networks to which it connects. The AP will not be damaged, or modified as the attack focus in the analysis of the packets transmitted.

[1] requires the **aircrack-ng** package

Start by searching the existing APs, their networks and the ESSID (Extended Service Set IDentifier). In this case you should focus your search in the APs that are dedicated to this attack. Confirm that they use WEP, and register the MAC address and channel (1-14). This will be required in the following steps. To search the APs, execute the following command:

```
$ airodump-ng DEVNAME
```

If no AP is shown, you may need to manually set the wireless card to monitor mode. To achieve this, consider the following commands:

```
$ ifconfig DEVNAME down
$ iw DEVNAME set monitor none
$ iw DEVNAME set channel 1
$ ifconfig  DEVNAME up
```

2.3.3 Analise the interaction with an AP

After the AP is selected, it is required to check if it is possible to inject frames to it. The injection may fail if the AP is applying a policy to filter MAC addresses. It can also happen if the WiFi card doesn't support packet injection.

The injection of frames can be executed without raising any suspect, if using standard probe frames such as Probe Request; The answer to this frame should be a Probe Response, unless the AP is filtering the MAC addresses of the clients, restricting its interaction with a predefined list of stations. It should be noticed that this policy provides little to no security as an attacker is free to listen the medium, enumerate the existing clients, and use the MAC address from one of the clients.

Execute the following command to set the WiFi monitoring interface (`MON0` in this case) to a specific channel.

```
$ airmon-ng stop MON0
$ airmon-ng start DEVNAME channel
```

replace the field `channel` with the channel used by the AP.

Then activate the injection test by executing the following command:

```
    aireplay-ng -9 -e target MON0
```

the option `-9` is used to specify that the tool should test the injection of frames. The result should consist of several messages indicating the success or failure of the process.

2.3.4 Find the network key

To find the network key the attacker must capture encrypted frames, which requires the observation of the traffic of a client with a MAC address involved in an association with the AP and is using WEP. Because the key is shared between the client and the AP, we will be using the MAC address of the AP, we it was registered in the beginning of this guide.

The following command will search the wireless medium for frames that contain a specific `MAC`, and will store this frames in a set of files with the prefix `capture`. We will consider this prefix in the remaining of this guide, but you are free to use a different prefix.

```
airodump-ng -bssid MAC -c channel -w capture MON0.
```

2.3.5 Increasing the attack speed

This attack is based on the amount of traffic exchanged with the target AP. If the amount of encrypted traffic is low, this attack may prove to be unsuccessful, as the required amount of special IVs (Initialization Vector) will be insufficient for the disclosure of the entire key.

If there are some clients connected to the AP, this issue can be circumvented. If the AP is an AP/Router with wired connections to other networks or Ethernet devices, this will also be adequate.

The purpose is to inject (actually replay) encrypted packets in order to provoke an encrypted response. The packet used, and the reply received are not relevant as they are encrypted and the attacker won't have access to the frame contents. The only objective is to increase the number of encrypted packets sent into the network, so that more special IVs can be received.

If there are no clients, and no traffic is sent by the AP, due to the large number of packets required, the attack will not succeed. If there is only some encrypted traffic between the victim AP and a terminal associated to it, the `airodump-ng` command will be able to extract pieces of the stream cipher for a specific IV, used for well-known frame (i.e., the challenge used in the Shared Key Authentication). With this pieces it is possible to forge frames that when sent to the AP, while impersonating another client (through MAC Spoofing), will trigger some response. This will be a unique packet that is sent encrypted to the network, thus also increasing the speed of the attack.

Search for a file having the name `capture-NUMBER-MAC.xor`, where `NUMBER` is a sequential version number and `MAC` is the `MAC` address of the victim AP.

From this point forward this file will be referred as the `SKAfile`.

In a new terminal (a new tab or window), execute the following command:

```
$ aireplay-ng -1 100000 -e network -y SKAfile MON0
```

where `network` is the name of the network (ESSID). This command will forge a Shared Key Authentication process, with the objective of allowing the attacker equipment to associate to the victim AP. Once it is associated, the attacker may inject encrypted packets. The association (which is required with the `-1` will last for 100 000 seconds).

In a new terminal, execute the following command:

```
$ aireplay-ng -3 -b MAC MON0
```

where `MAC` is the MAC address of the AP. This command will wait until an ARP packet is sent (frequently it is an ARP Request), and will re-inject it continuously. If the attack is working, you should observe a significant increase in the number of data frames (not Beacons) captured by the `airodump-ng` tool. This has the meaning that more encrypted traffic is being captured. Therefore, more IVs are also being used, and the higher are the possibilities of finding the network key.

2.3.6 Discover the network key

Now, everything should be configured and ready to actually start finding the network key from the encrypted frames that are captured.

In another terminal, execute the following command:

```
$ aircrack-ng capture-NUMBER.cap
```

where `NUMBER` is the first value of the sequence. The command will process the encrypted frames and will try to find the network key.

In the end, the key should be displayed. You can explore the several options of this command and use different attacks, both to compare the result, but also to analyze the success rate and complexity in terms of CPU usage, Time, or number of packets required.

After all tests, restore the configuration of the WiFi interface and connect to the network. You should be able to access and and have a valid IP address.

2.4 WPA attacks

The current setup is directly vulnerable to two attacks, that enable attackers to recover the secret used to generate the PSK/PMK. Two attacks will be considered, as described next.

WPA-EAPOL-PBKDF2

The first attack will consist in capturing frames from the Four Way Handshake (4WH), and then retrieve the secret using a brute force or diccionary attack. This attack only works if there are clients! If no clients are associated, no handshake will be captured.

This works because the Pairwise Transient Key is calculated in a predictable way and based on information available to the attacker, except for the secret. As discussed in the theoretical classes, the Pairwise Master Key (PMK) is computed as follows: `PMK=PBKDF2(HMAC-SHA1, password, ssid, 4096, 256)`, and then used to compute the Pairwise Transient Key (PTK) as follow: `PTK = PRF(PMK, ANonce, SNonce, AP MAC, STA MAC)`.

The process is rather simple, and most of the difficulty resides in the number of rounds used for the PBKDF2 (4096), the Digest used (SHA1), and finally the length of the password. Small passwords can be retrieved rather easily. Passwords larger than 12-13 chars are not easily found using a brute force approach. Of course, people doesn't generate random passwords and diccionaries can really help an attacker.

WPA-PMKID

The second attack exploits the fact that some Access Points will send information to facilitate the authentication process for roaming clients. This information element is named the PMKID (Parwise Master Key Identifier) which is constructed as follows: `PMKID=HMAC-SHA1-128(PMK, "PMK Name", MAC_AP, MAC_STA)`.

There aren't much diferences from the previous, with the exception that no clients are required. An attacker can develop the attack to any WPA2 network where the AP exposes the PMKID.

The recovery process is similar to the previous case, and requires the use of some software to do a brute force or diccionary attack.

2.4.1 Attacking WPA2 Networks

The first step of any attack resorts to capturing data (PMKIDs or 4WH), which can be done with the `bettercap` tool.

In linux, you can start `bettercap` by issuing `sudo bettercap -iface wlan0`. You need to provide an actual wireless network interface instead of `wlan0`.

Then, enable the `wifi.recon` module by issuing `wifi.recon on`. From this moment on, `bettercap` will listen too all Wi-Fi channels, looking for hashes (4WH or PMKIDS), and saving those hashes to the storage. The process is fully automated an no other interaction is required. You can see what BSSIDs are known by issuing `wifi.show`.

Sometimes there will be no clients or PMKIDs, or the clients are not authenticating to the network. In this case it is required to *force* clients to reauthenticate. Using `bettercap`, issue `wifi.deauth BSSID`, where `BSSID` represents the address of the target Access Point.

If there are clients associated, they will be deauthenticated. Clients will naturally reauthenticate to the network and handshakes will be captured.

2.4.2 Recovering the Password

For this step we will take the capture obtained by `bettercap` and use an additional software to attack the hashes. Depending on you operating system, choose one of the suggested tools (Hashcat, John, Hashsuite).

Hashcat

To use Hashcat, first we need to convert the `pcap` file to a format understood by Hashcat. For attacking the 4WH, the following steps must be followed:

1. Extract the handshakes:
   `cap2hccapx bettercap-wifi.pcap bettercap-wifi.hccapx`
2. Run Hashcat to obtain the secret:
   `hashcat -m2500 -a3 -w3 bettercap-wifi.hccapx '?l?l?l?l?l?l?l?l'`

For attacking the PMKIDs, the following steps must be followed:

1. Extract the PMKIDs:
   `hcxpcaptool -z bettercap-wifi.pmkid bettercap-wifi.pcap`
2. Run Hashcat to obtain the secret:
   `hashcat -m16800 -a3 -w3 bettercap-wifi.pmkid '?l?l?l?l?l?l?l?l'`

There are several attacks possible by Hashcat, and defined by the `-a` argument. For compreensive information please check `https://hashcat.net/wiki/#core_attack_modes`

For diccionaries, that may assist you, check `https://bit.ly/2OBGB7K`

John the Ripper and Hashsuite

Hashsuite is a fork from John The Ripper, so most steps will be similar. Hashsuite has a GUI which also facilitates the process.

To attack the 4WH, the following steps must be followed:

1. Extract the handshakes: `cap2hccapx bettercap-wifi.pcap bettercap-wifi.hccap`
2. Convert the handshakes to a form compatible by John: `hccap2john bettercap-wifi.h`
3. Execute `john -form=wpapsk bettercap-wifi.john diccionary.txt`

To attack the PMKIDs, the following steps must be followed:

1. Using Wireshark, open the `pcap` file and locate an RSN being transmitted. Copy the **HEX** values of the required fields to a text file with the following format `PMKID*bSSID*STA MAC*SSID`. All values should be in hex.
2. Execute `john pmkids.txt diccionary.txt`

2.5 Bibliography

`http://www.aircrack-ng.org`