Aircrack-ng

Airmon-ng

Description

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

Usage

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. For “check kill” see

Usage Examples

Typical Uses

Check status and/or listing wireless interfaces

```
~# airmon-ng
PHY     Interface       Driver          Chipset

phy0    wlan0           ath9k_htc       Atheros Communications, Inc. AR9271 802.11n
```

Checking for interfering processes

When putting a card into monitor mode, it will automatically check for interfering processes. It can also be done manually by running the following command:

```
~# airmon-ng check
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
  718 NetworkManager
  870 dhclient
 1104 avahi-daemon
 1105 avahi-daemon
 1115 wpa_supplicant
```

Killing interfering processes

This command stops network managers then kill interfering processes left:

```
~# airmon-ng check kill
Killing these processes:

  PID Name
  870 dhclient
 1115 wpa_supplicant
```

Enable monitor mode

Note: It is very important to kill the network managers before putting a card in monitor mode!

```
~# 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
  718 NetworkManager
  870 dhclient
 1104 avahi-daemon
 1105 avahi-daemon
 1115 wpa_supplicant

PHY     Interface       Driver          Chipset
```

```
phy0    wlan0           ath9k_htc       Atheros Communications, Inc. AR9271 802.11n
                (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
                (mac80211 station mode vif disabled for [phy0]wlan0)
```

As you can see, it created a monitor mode interface called wlan0mon and it notified there are a few process that will interfere with the tools.

Disable monitor mode

```
~# airmon-ng stop wlan0mon
PHY     Interface       Driver          Chipset

phy0    wlan0mon        ath9k_htc       Atheros Communications, Inc. AR9271 802.11n
                (mac80211 station mode vif enabled on [phy0]wlan0)
                (mac80211 monitor mode vif disabled for [phy0]wlan0mon)
```

Don't forget to restart the network manager. It is usually done with the following command:

```
service network-manager start
```

Madwifi-ng driver monitor mode

This describes how to put your interface into monitor mode. After starting your computer, enter “iwconfig” to show you the current status of the wireless interfaces. It likely looks similar the following output.

Enter “iwconfig”:

```
 lo        no wireless extensions.

 eth0      no wireless extensions.

 wifi0     no wireless extensions.

 ath0      IEEE 802.11b  ESSID:""  Nickname:""
           Mode:Managed  Channel:0  Access Point: Not-Associated
           Bit Rate:0 kb/s   Tx-Power:0 dBm   Sensitivity=0/3
           Retry:off   RTS thr:off   Fragment thr:off
           Encryption key:off
           Power Management:off
           Link Quality:0  Signal level:0  Noise level:0
           Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
           Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

If you want to use ath0 (which is already used):

```
airmon-ng stop ath0
```

And the system will respond:

```
 Interface       Chipset         Driver

 wifi0           Atheros         madwifi-ng
 ath0            Atheros         madwifi-ng VAP (parent: wifi0) (VAP destroyed)
```

Now, if you do “iwconfig”:

System responds:

```
 lo        no wireless extensions.

 eth0      no wireless extensions.

 wifi0     no wireless extensions.
```

You can see ath0 is gone.

To put wifi0 in monitor mode:

```
airmon-ng start wifi0
```

System responds:

```
 Interface       Chipset         Driver

 wifi0           Atheros         madwifi-ng
 ath0            Atheros         madwifi-ng VAP (parent: wifi0) (monitor mode enabled)
```

Now enter “iwconfig”

System responds:

```
 lo        no wireless extensions.

 eth0      no wireless extensions.

 wifi0     no wireless extensions.

 ath0      IEEE 802.11g  ESSID:""
        Mode:Monitor  Frequency:2.452 GHz  Access Point: 00:0F:B5:88:AC:82
        Bit Rate=2 Mb/s   Tx-Power:18 dBm   Sensitivity=0/3
```

```
        Retry:off   RTS thr:off   Fragment thr:off
        Encryption key:off
        Power Management:off
        Link Quality=0/94  Signal level=-96 dBm  Noise level=-96 dBm
        Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
        Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

You can see ath0 is in monitor mode. Also make sure the essid, nickname and encryption have not been set. The access point shows the MAC address of the card. The MAC address of the card is only shown when using the madwifi-ng driver. Other drivers do not show the MAC address of the card.

If ath1/ath2 etc. is running then stop them first prior to all the commands above:

```
 airmon-ng stop ath1
```

You can set the channel number by adding it to the end: airmon-ng start wifi0 9

Usage Tips

Confirming the Card is in Monitor Mode

To confirm that the card is in monitor mode, run the command “iwconfig”. You can then confirm the mode is “monitor” and the interface name.

For the madwifi-ng driver, the access point field from iwconfig shows your the MAC address of the wireless card.

Determining the Current Channel

To determine the current channel, enter “iwlist <interface name> channel”. If you will be working with a specific access point, then the current channel of the card should match that of the AP. In this case, it is a good idea to include the channel number when running the initial airmon-ng command.

How Do I Put My Card Back into Managed Mode?

It depends on which driver you are using. For all drivers except madwifi-ng:

```
airmon-ng stop <interface name>
```

For madwifi-ng, first stop ALL interfaces:

```
airmon-ng stop athX
```

Where X is 0, 1, 2 etc. Do a stop for each interface that iwconfig lists.

Then:

```
wlanconfig ath create wlandev wifi0 wlanmode sta
```

See madwifi-ng site documentation [http://madwifi-project.org/wiki/UserDocs/StationInterface].

For mac80211 drivers, nothing has to be done, as airmon-ng keeps the managed interface alongside the monitor mode one (mac80211 uses interface types rather than modes of operation). If you no longer need the monitor interface and want to remove it, use the following:

```
airmon-ng stop monX
```

X is the monitor interface number - 0 unless you run multiple monitoring interfaces simultaneously.

Debugging issues

airmon-ng has two options to show more information, which can be useful when reporting or debugging issues.

--verbose flag

It gives information about the system as well as details about the wireless card.

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

No LSB modules are available.
Distributor ID: Kali
Description:    Kali GNU/Linux Rolling
Release:        2019.1
Codename:       n/a

Linux kali 4.19.0-kali4-amd64 #1 SMP Debian 4.19.28-2kali1 (2019-03-18) x86_64 GNU/Linux
Detected VM using lspci
This appears to be a VMware Virtual Machine
If your system supports VT-d, it may be possible to use PCI devices
If your system does not support VT-d, you can only use USB wifi cards

K indicates driver is from 4.19.0-kali4-amd64
V indicates driver comes directly from the vendor, almost certainly a bad thing
S indicates driver comes from the staging tree, these drivers are meant for reference not actual use, BEWARE
? indicates we do not know where the driver comes from... report this


X[PHY]Interface         Driver[Stack]-FirmwareRev               Chipset                 Extended Info

K[phy1]wlan0            ath9k_htc[mac80211]-1.4                 Qualcomm Atheros Communications AR9271 802.11n                                  mode managed
```

In this case, the following additional information can be seen:

1. Detailed information about the Linux distribution as well as kernel version
2. System is a virtual machine (and detailed information about supported features)
3. Detailed driver information (kernel, vendor driver, staging or unknown source), wireless stack, current operating mode and firmware version

--debug flag

It will give the same information as verbose and add more details:

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

/bin/sh -> /usr/bin/dash

SHELL is GNU bash, version 5.0.3(1)-release (x86_64-pc-linux-gnu)
Copyright (C) 2019 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>

This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

No LSB modules are available.
Distributor ID: Kali
Description:    Kali GNU/Linux Rolling
Release:        2019.1
Codename:       n/a

Linux kali 4.19.0-kali4-amd64 #1 SMP Debian 4.19.28-2kali1 (2019-03-18) x86_64 GNU/Linux
Detected VM using lspci
This appears to be a VMware Virtual Machine
If your system supports VT-d, it may be possible to use PCI devices
If your system does not support VT-d, you can only use USB wifi cards

K indicates driver is from 4.19.0-kali4-amd64
V indicates driver comes directly from the vendor, almost certainly a bad thing
S indicates driver comes from the staging tree, these drivers are meant for reference not actual use, BEWARE
? indicates we do not know where the driver comes from... report this


X[PHY]Interface         Driver[Stack]-FirmwareRev               Chipset                 Extended Info

getStack mac80211
getBus usb
getdriver() ath9k_htc
getchipset() Qualcomm Atheros Communications AR9271 802.11n
BUS = usb
BUSINFO = 0CF3:9271
DEVICEID =
getFrom() K
getFirmware 1.4
K[phy1]wlan0            ath9k_htc[mac80211]-1.4                 Qualcomm Atheros Communications AR9271 802.11n                                  mode managed
```

Additional information:

1. Shell name and version
2. Debug information regarding the wireless adapter and loaded driver

Usage Troubleshooting

Madwifi-ng

Quite often, the standard scripts on a linux distribution will setup ath0 and or additional athX interfaces. These must all be removed first per the instructions above. Another problem is that the script set fields such as essid, nickname and encryptions. Be sure these are all cleared.

Airmon-ng says the interface is not in monitor mode

```
~# airmon-ng stop wlan0mon
PHY     Interface       Driver          Chipset

phy0    wlan0mon        ath9k_htc       Atheros Communications, Inc. AR9271 802.11n

You are trying to stop a device that isn't in monitor mode.
Doing so is a terrible idea, if you really want to do it then you
need to type 'iw wlan2mon del' yourself since it is a terrible idea.
Most likely you want to remove an interface called wlan[0-9]mon
If you feel you have reached this warning in error,
please report it.
```

It most likely mean the interface mode was changed from monitor to managed mode by a network manager. In this case, when stopping monitor mode, this is not a problem.

My interface was put in monitor mode but tools says it is not

It usually means the interface was put in monitor mode prior to killing network managers. And the network manager put the card back in managed mode.

Refer to the documentation above to kill network managers and put it back into monitor mode.

Interface athX number rising (ath0, ath1, ath2.... ath45..)

The original problem description and solution can be found in this forum thread [http://forum.aircrack-ng.org/index.php?topic=1641.0].

Problem: Every time the command “airmon-ng start wifi0 x” is run, a new interface is created as it should, but there where two problems. The first is that for each time airmon-ng is run on wifi0 the interface number on ath increases: the first time is ath1, the second ath2, the third ath3, and and so on. And this continues so in a short period of time it is up to ath56 and continuing to climb. Unloading the madwifi-ng driver, or rebooting the system has no effect, and the number of the interface created by airmon-ng continues to increase.

The second problem is that if you run airmon-ng on wifi0 the athXX created does not show as being shown as in Monitor mode, even though it is. This can be confirmed via iwconfig.

All these problem related to how udev assigns interface names. The answer is in this ticket: http://madwifi-project.org/ticket/972#comment:12 [http://madwifi-project.org/ticket/972#comment:12] Thanks to lucida. The source of the problem comes from the udev persistent net rules generator.

Each distro is different… So here is a solution specifically for Gentoo. You should be able to adapt this solution to your particular distribution.

Gentoo 2.6.20-r4 Udev 104-r12 Madwifi 0.9.3-r2 Aircrack-ng 0.7-r2

Solution:

Change the file /etc/udev/rules.d/75-persistent-net-generator.rules

From: KERNEL==“eth*|ath*|wlan*|ra*|sta*…….. To: KERNEL==“eth*|Ath*|wlan*|ra*|sta*…….

In other words, you just capitalize the a. ath* becomes Ath*. Save the file.

Now delete the file /etc/udev/rules.d/70-persistent-net.rules.

Remove the driver and insert back.

Removing ath also works: KERNEL==“eth*|wlan*|ra*|sta*….

This is also on Gentoo, both 2.6.19-gentoo-r5 and 2.6.20-gentoo-r6

For Ubuntu, see this Forum posting [http://forum.aircrack-ng.org/index.php?topic=2674.msg14904#msg14904]. The modified version of /etc/udev/rules.d/75-persistent-net-generator.rules is:

```
 # these rules generate rules for persistent network device naming

 ACTION=="add", SUBSYSTEM=="net", KERNEL=="eth*|Ath*|wlan*|ra*|sta*" \
NAME!="?*", DRIVERS=="?*", GOTO="persistent_net_generator_do"

 GOTO="persistent_net_generator_end"
 LABEL="persistent_net_generator_do"

 # build device description string to add a comment the generated rule
 SUBSYSTEMS=="pci", ENV{COMMENT}="PCI device attr{vendor}:$attr{device}($attr{driver})"
 SUBSYSTEMS=="usb", ENV{COMMENT}="USB device 0x$attr{idVendor}:0x$attr{idProduct}($attr{driver})"
 SUBSYSTEMS=="ieee1394", ENV{COMMENT}="Firewire device $attr{host_id})"
 SUBSYSTEMS=="xen", ENV{COMMENT}="Xen virtual device"
 ENV{COMMENT}=="", ENV{COMMENT}="$env{SUBSYSTEM} device ($attr{driver})"

 IMPORT{program}="write_net_rules $attr{address}"

 ENV{INTERFACE_NEW}=="?*", NAME="$env{INTERFACE_NEW}"

 LABEL="persistent_net_generator_end"
```

Interface ath1 created instead of ath0

This troubleshooting tip applies to madwifi-ng drivers. First try stopping each VAP interface that is running (“airmon-ng stop IFACE” where IFACE is the VAP name). You can obtain the list from iwconfig. Then do “airmon-ng start wifi0”.

If this does not resolve the problem then follow the advice in this thread [http://forum.aircrack-ng.org/index.php?topic=2044.0].

Why do I get ioctl(SIOCGIFINDEX) failed?

If you get error messages similar to:

- Error message: “SIOCSIFFLAGS : No such file or directory”
- Error message: “ioctl(SIOCGIFINDEX) failed: No such device”

Then See this FAQ entry.

Error message: "wlanconfig: command not found"

If you receive “wlanconfig: command not found” or similar then the wlanconfig command is missing from your system or is not in the the path. Use locate or find to determine if it is on your system and which directory it is in.

If it is missing from your system then make sure you have done a “make install” after compiling the madwifi-ng drivers. On Ubuntu, do “apt-get install madwifi-tools”.

If it is not in a directory in your path then move it there or add the directory to your path.

airmon-ng shows RT2500 instead of RT73

See this entry under installing the RT73 driver.

Error "add_iface: Permission denied"

You receive an error similar to:

```
 Interface       Chipset         Driver

 wlan0                    iwl4965 - [phy0]/usr/sbin/airmon-ng: line 338: /sys/class/ieee80211/phy0/add_iface: Permission denied
                                mon0: unknown interface: No matching device found
                                (monitor mode enabled on mon0)
```

or similar to this:

```
 wlan0   iwlagn - [phy0]/usr/local/sbin/airmon-ng: 856: cannot create /sys/class/ieee80211/phy0/add_iface: Directory nonexistent
 Error for wireless request "Set Mode" (8B06) :
  SET failed on device mon0 ; No such device.
 mon0: ERROR while getting interface flags: No such device
```

This means you have an old version of airmon-ng installed. Upgrade to at least v1.0-rc1. Preferably you should upgrade to the current version. See the installation page for more details. Also, don't forget you need to be root to use airmon-ng (or use sudo).

check kill fails

Distros from now on are going to adopt 'upstart' which is going to replace the /sbin/init daemon which manages services and tasks during boot.

Basically do:

```
 service network-manager stop
 service avahi-daemon stop
 service upstart-udev-bridge stop
```

and then proceed with greping and killing the pids of dhclient and wpa_supplicant.

This is the only way to kill ALL of the potentially problematic pids for aireplay-ng permanently. The trick is the kill the daemons first and then terminate the 'tasks'.

Source thread: http://forum.aircrack-ng.org/index.php?topic=6398.0 [http://forum.aircrack-ng.org/index.php?topic=6398.0] and http://forum.aircrack-ng.org/index.php?topic=8573 [http://forum.aircrack-ng.org/index.php?topic=8573]

SIOCSIFFLAGS: Unknown error 132

If you have an output similar to:

```
# airmon-ng start wlan0
Interface       Chipset         Driver
wlan0           Broadcom        b43 - [phy0]SIOCSIFFLAGS: Unknown error 132
                                (monitor mode enabled on mon0)
```

It indicates that RF are blocked. It needs to be enabled by using the switch on your laptop and/or using the following command:

```
rfkill unblock all
```

See also http://ubuntuforums.org/showthread.php?t=1311886 [http://ubuntuforums.org/showthread.php?t=1311886]

ERROR adding monitor mode interface: command failed: Operation not supported (-95)

It is known to happen on the Raspberry Pi, when using airmon-ng. When that happens, the following can be seen in dmesg:

```
brcmfmac: brcmf_vif_add_validate: Attempt to add a MONITOR interface...
brcmfmac: brcmf_vif_add_validate: ... there is already a monitor interface, returning EOPNOTSUPP
brcmfmac: brcmf_cfg80211_add_iface: iface validation failed: err=-95
```

There may be instances of the following in dmesg as well prior to the above output:

```
brcmfmac: brcmf_vif_add_validate: Attempt to add a MONITOR interface...
brcmfmac: brcmf_mon_add_vif: brcmf_mon_add_vif called
brcmfmac: brcmf_mon_add_vif: Adding vif "wlan0mon"
brcmfmac: brcmf_cfg80211_get_channel: chanspec failed (-52)
```

Even though dmesg says the interface is already in monitor mode and “iw dev wlan0 info” confirms it is, airodump-ng will fail and report the interface data linktype is Ethernet. This is a bug in the driver and/or firmware, and the workaround is to reboot the system or to reload the driver:

```
rmmod brcmfmac
modprobe brcmfmac
```
