Linux Commando

This blog is about the Linux Command Line Interface (CLI), with an occasional foray into GUI territory. Instead of just giving you information like some man page, I hope to illustrate each command in real-life scenarios.

Saturday, October 19, 2013

How to connect to a WPA/WPA2 WiFi network using Linux command line

This is a step-to-step guide for connecting to a WPA/WPA2 WiFi network via the Linux command line interface. The tools are:

- *wpa_supplicant*
- *iw*
- *ip*
- *ping*

iw is the basic tool for WiFi network-related tasks, such as finding the WiFi device name, and scanning access points. *wpa_supplicant* is the wireless tool for connecting to a WPA/WPA2 network. *ip* is used for enabling/disabling devices, and finding out general network interface information.

The steps for connecting to a WPA/WPA2 network are:

1. Find out the wireless device name.

```
$ /sbin/iw dev
phy#0
        Interface wlan0
                ifindex 3
                type managed
```

The above output showed that the system has 1 physical WiFi card, designated as *phy#0*. The device name is *wlan0*. The *type* specifies the operation mode of the wireless device. *managed* means the device is a WiFi station or client that connects to an access point.

2. Check that the wireless device is up.

```
$ ip link show wlan0
3: wlan0: (BROADCAST,MULTICAST) mtu 1500 qdisc noop state DOWN mode DEFAUL
    link/ether 74:e5:43:a1:ce:65 brd ff:ff:ff:ff:ff:ff
```

Look for the word "UP" inside the brackets in the first line of the output.

In the above example, *wlan0* is not UP. Execute the following command to bring it up:

```
$ sudo ip link set wlan0 up
[sudo] password for peter:
```

Note: you need root privilege for the above operation.

If you run the show link command again, you can tell that *wlan0* is now UP.

```
$ ip link show wlan0
3: wlan0: (NO-CARRIER,BROADCAST,MULTICAST,UP) mtu 1500 qdisc mq state DOWN mode DEFAULT qlen 1000
    link/ether 74:e5:43:a1:ce:65 brd ff:ff:ff:ff:ff:ff
```

3. Check the connection status.

```
$ /sbin/iw wlan0 link
Not connected.
```

The above output shows that you are not connected to any network.

4. Scan to find out what WiFi network(s) are detected

```
$ sudo /sbin/iw wlan0 scan
BSS 00:14:d1:9c:1f:c8 (on wlan0)
        ... sniped ...
        freq: 2412
```

```
        SSID: gorilla
        RSN:     * Version: 1
                 * Group cipher: CCMP
                 * Pairwise ciphers: CCMP
                 * Authentication suites: PSK
                 * Capabilities: (0x0000)
        ... sniped ...
```

The 2 important pieces of information from the above are the SSID and the security protocol (WPA/WPA2 vs WEP). The SSID from the above example is *gorilla*. The security protocol is RSN, also commonly referred to as WPA2. The security protocol is important because it determines what tool you use to connect to the network.

5. Connect to WPA/WPA2 WiFi network.

   This is a 2 step process. First, you generate a configuration file for *wpa_supplicant* that contains the pre-shared key ("passphrase") for the WiFi network.

```
$ sudo -s
[sudo] password for peter:
$ wpa_passphrase gorilla >> /etc/wpa_supplicant.conf
...type in the passphrase and hit enter...
```

   wpa_passphrase takes the SSID as the single argument. You must type in the passphrase for the WiFi network *gorilla* after you run the command. Using that information, *wpa_passphrase* will output the necessary configuration statements to the standard output. Those statements are appended to the *wpa_supplicant* configuration file located at */etc/wpa_supplicant.conf*.

   Note: you need root privilege to write to */etc/wpa_supplicant.conf*.

```
$ cat /etc/wpa_supplicant.conf
# reading passphrase from stdin
network={
        ssid="gorilla"
        #psk="testtest"
        psk=4dfe1c985520d26a13e932bf0acb1d4580461dd854ed79ad1a88ec221a802061
}
```

   The second step is to run *wpa_supplicant* with the new configuration file.

```
$ sudo wpa_supplicant -B -D wext -i wlan0 -c /etc/wpa_supplicant.conf
```

   -B means run *wpa_supplicant* in the background.

   -D specifies the wireless driver. *wext* is the generic driver.

   -c specifies the path for the configuration file.

   Use the *iw* command to verify that you are indeed connected to the SSID.

```
$ /sbin/iw wlan0 link
Connected to 00:14:d1:9c:1f:c8 (on wlan0)
        SSID: gorilla
        freq: 2412
        RX: 63825 bytes (471 packets)
        TX: 1344 bytes (12 packets)
        signal: -27 dBm
        tx bitrate: 6.5 MBit/s MCS 0

        bss flags:      short-slot-time
        dtim period:    0
        beacon int:     100
```

6. Obtain IP address by DHCP

```
$ sudo dhclient wlan0
```

   Use the *ip* command to verify the IP address assigned by DHCP. The IP address is 192.168.1.113 from below.

```
$ ip addr show wlan0
3: wlan0:  mtu 1500 qdisc mq state UP qlen 1000
    link/ether 74:e5:43:a1:ce:65 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.113/24 brd 192.168.1.255 scope global wlan0
    inet6 fe80::76e5:43ff:fea1:ce65/64 scope link
       valid_lft forever preferred_lft forever
```

7. Add default routing rule.

   The last configuration step is to make sure that you have the proper routing rules.

```
$ ip route show
192.168.1.0/24 dev wlan0  proto kernel  scope link  src 192.168.1.113
```

   The above routing table contains only 1 rule which redirects all traffic destined for the local subnet (192.168.1.x) to the *wlan0* interface. You may want to add a default routing rule to pass all other traffic through *wlan0* as well.

► November (5)
▼ October (3)
How to connect to a WPA/WPA2 WiFi network using Li...
How to use the screen command to run a program una...
Allow root ssh login with public key authenticatio...
► September (1)
► May (2)
► April (2)
► March (3)
► February (1)
► 2012 (1)
► 2010 (1)
► 2009 (9)
► 2008 (51)
► 2007 (21)

```
$ sudo ip route add default via 192.168.1.254 dev wlan0
$ ip route show
default via 192.168.1.254 dev wlan0
192.168.1.0/24 dev wlan0  proto kernel  scope link  src 192.168.1.113
```

8. ping external ip address to test connectivity

```
$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_req=1 ttl=48 time=135 ms
64 bytes from 8.8.8.8: icmp_req=2 ttl=48 time=135 ms
64 bytes from 8.8.8.8: icmp_req=3 ttl=48 time=134 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 134.575/134.972/135.241/0.414 ms
```

The above series of steps is a very verbose explanation of how to connect a WPA/WPA2 WiFi network. Some steps can be skipped as you connect to the same access point for a second time. For instance, you already know the WiFi device name, and the configuration file is already set up for the network. The process needs to be tailored according to your situation.

Posted by Peter Leung at 9:09 PM

28 comments:

jaspreet singh said...

This comment has been removed by the author.

February 4, 2014 at 8:01 AM

Conor said...

Oh thank goodness you've written this up, step by step instead of just "use wpa_supplicant". WPA was a maze of twisty passages and I didn't know about wpa_passphrase. Thanks a lot for this.

March 11, 2014 at 5:45 AM

Anonymous said...

excelente ... muchas gracias
Francisco

March 15, 2014 at 7:39 PM

Anonymous said...

What to do if I have hidden SSID for WPA2 wifi?

#iw wlan0 scan
....
....
SSID: \x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
....
....

March 16, 2014 at 1:06 PM

Anonymous said...

PS. I know name of hidden SSID

March 16, 2014 at 1:07 PM

Peter Leung said...

For hidden SSIDs, edit wpa_supplicant.conf.

Add 'scan_ssid=1' to the network definition.

Eg,

network={
ssid="gorilla"
psk=4dfe1c985520d26a13e932bf0acb1d4580461dd854ed79ad1a88ec221a802061
scan_ssid=1
}

March 17, 2014 at 10:42 AM

Anonymous said...

Messed up my graphics driver install so the network manager gui wouldn't start. Saved my day! Thanks. Put that in a script now :)

March 23, 2014 at 3:04 PM

jaspreet singh said...

nothing better than the Linux the only draw back with Linux is it is not user friendly and post like this help many to do the task on linux

April 9, 2014 at 5:45 AM

Anonymous said...

you saved be, thank you

April 17, 2014 at 10:45 AM

Anonymous said...

 # wpa_supplicant -B -D wext -i wlan0 -c /etc/wpa_supplicant.conf
Successfully initialized wpa_supplicant
ioctl[SIOCSIWENCODEEXT]: Invalid argument
ioctl[SIOCSIWENCODEEXT]: Invalid argument
 # iw wlan0 link
Not connected.

xubuntu 14.04, Qualcomm Atheros AR9285 Wireless Network Adapter (PCI-Express) (rev 01)

April 25, 2014 at 11:31 AM

---

jaspreet singh said...

Perfect post for slove my problem atleast

red wimax

April 25, 2014 at 11:08 PM

Nick said...

I have almost the same problem as anonymous at April 25 above. But mine doesn't even say successfully initialised before returning the invalid argument message. Any help greatly appreciated. Thanks.

May 17, 2014 at 10:34 PM

Satheesh P'h'p said...

Thank you so much for this great post. You saved me from a big disaster. Thanks a lot. But I'm suffering from last one problem. I've connected to the network in my office. I can view IP from 'ifconfig' command.
But I couldn't connect to this system from a external system at given ip. I've looked upon router configuration page where my device has been connected but ip isn't shown. when I use 'ip route show' this is coming,
192.168.1.0/24 dev wlan0 proto kernel scope link src 192.168.1.12.

Please help on this... :-(

May 22, 2014 at 6:08 AM

Anonymous said...

Thank you SO much for this!

May 31, 2014 at 11:48 PM

Anonymous said...

I used none instead of wext after the -D option and it solved the problem.

June 28, 2014 at 10:59 AM

Anonymous said...

I used none instead of wext and it solved the problem

June 28, 2014 at 11:02 AM

---

jaspreet singh said...

Nice thanks for share

wifi en eventos

August 11, 2014 at 6:55 AM

---

gordon said...

>>Anonymous said...
>>I used none instead of wext after the
>>-D option and it solved the problem.

I get: Unsupported driver 'none'

If remove the -D option, I get Invalid argument errors.

I'm so close on this. Please help.

August 28, 2014 at 11:34 AM

Bo Grimes said...

Thanks. I had a power outage during a distro upgrade. I couldn't get back into the GUI, so I used this to connect at console and finish the upgrade. Plus I know more about networking now.

August 28, 2014 at 2:55 PM

Emerson Prado said...

Many thanks! My Linux Mint Debian Edition powered notebook is Wi-Fi connected now. Great article!
I just had to change the route command in step 7 to:
sudo route add default gw <gateway> <interface>

Also, the route and the wpa_supplicant command aren't persistent. For a permanent change, we have to add them in /etc/network/interfaces (for Debian), in the Wi-Fi interface section. Mine ended up like:

auto <interface>
iface <interface> inet dhcp
wireless-essid <essid>
pre-up wpa_supplicant -B -D wext -i <interface> -c /etc/wpa_supplicant.conf
post-up route add default gw <gateway> <interface>
post-down killall -q wpa_supplicant

Keep up,
Emerson

September 1, 2014 at 1:28 PM

Anonymous said...

Grandiose and verbose gratitude. Excellent step by step!

September 7, 2014 at 6:08 PM

---

arvindh said...

Hey,.
nice post. But i have a WIFI network without any security protocol.

BSS xx:xx:xx:xx:xx:xx (on wlan0)
TSF: 478427021 usec (0d, 00:07:58)
freq: 2437
beacon interval: 100
capability: ESS ShortPreamble ShortSlotTime (0x0421)
signal: -45.00 dBm
SSID: tpay
Supported rates: 1.0* 2.0* 5.5* 11.0*
DS Parameter set: channel 6
ERP:
Extended supported rates: 6.0 9.0 12.0 18.0 24.0 36.0 48.0 54.0
HT capabilities:
Capabilities: 0x12c
HT20
SM Power Save disabled
RX HT20 SGI
RX STBC 1-stream
Max AMSDU length: 7935 bytes
No DSSS/CCK HT40
Maximum RX AMPDU length 65535 bytes (exponent: 0x003)
Minimum RX AMPDU time spacing: No restriction (0x00)
HT RX MCS rate indexes supported: 0-7
HT TX MCS rate indexes are undefined
WMM: * Parameter version 1
* u-APSD
* BE: CW 15-1023, AIFSN 3
* BK: CW 15-1023, AIFSN 7
* VI: CW 7-15, AIFSN 2, TXOP 3008 usec
* VO: CW 3-7, AIFSN 2, TXOP 1504 usec

How can i connect to this network through command line,..???
can any one plz help,..!!

September 24, 2014 at 9:22 AM

Bostek said...

Dude you're awsome!! Thx for that. I had a big problem to configure wireless until i found your step by step solution!

Everything works great now!!
Thx again!

October 17, 2014 at 7:46 AM

Anonymous said...

Thank you so much for that. It solved my problem as well :)

November 23, 2014 at 12:05 AM

Softql said...

very nice blog i am very interseting your blog
RF Post processing

March 19, 2015 at 12:18 AM

Anonymous said...

Thank you so much!

May 9, 2015 at 5:13 PM

---

trayalis29 said...

Thank you verymuch for your step by step procedures and the explanations.

May 26, 2015 at 5:12 PM

Anonymous said...

you are awesome. Thanks for a great tut.

May 27, 2015 at 9:02 PM

Post a Comment

Links to this post

Create a Link

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)