



Using WPA_Supplicant to Connect to WPA2 Wi-fi from Terminal on Ubuntu 16.04 Server

 Last Updated: October 29th, 2021  Xiao Guoan (Admin)  41 Comments
 Command Line

In this tutorial, we are going to learn how to connect to Wi-fi network from the command line on Ubuntu 16.04 server and desktop using `wpa_supplicant`.

In a modern home wireless network, communications are protected with WPA-PSK (pre-shared key) as opposed to WPA-Enterprise, which is designed for enterprise networks. WPA-PSK is also known as WPA-Personal. **wpa_supplicant** is an implementation of the WPA supplicant component. A supplicant in wireless LAN is a client software installed on end-user's computer that needs to be authenticated in order to join a network.

Please note that you will need to install the `wpa_supplicant` software before connecting to Wi-Fi, so you need to connect to Wired Ethernet first, which is done for just one time. If you don't like this method, please don't be mad at me. Maybe someday Ubuntu will ship `wpa_supplicant` on a clean install.

Step 1: Enable wireless interface

First, make sure your wireless card is enabled. You can use **rfkill**.

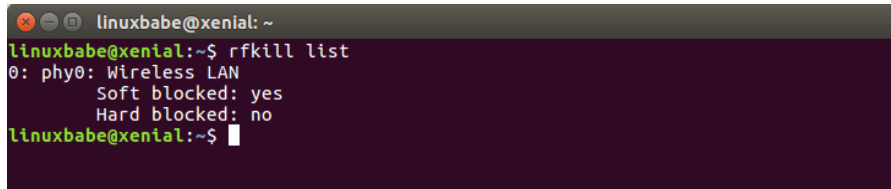
```
sudo apt install rfkill
```

To check the status of wireless card, run



```
rfkill list
```

Sample output:

A terminal window with a dark purple background. The prompt is 'linuxbabe@xenial: ~'. The command 'rfkill list' has been executed, showing the output: '0: phy0: Wireless LAN', 'Soft blocked: yes', and 'Hard blocked: no'. The prompt is now 'linuxbabe@xenial:~\$' with a cursor.

```
linuxbabe@xenial:~$ rfkill list
0: phy0: Wireless LAN
    Soft blocked: yes
    Hard blocked: no
linuxbabe@xenial:~$
```

As you can see, my wireless card is blocked by software. To unblock it, use the following command:

```
rfkill unblock wifi
```

If you are using the desktop version of Ubuntu, then you also need to stop Network Manager with the following command, otherwise it will cause connection problem when using wpa_supplicant.

```
sudo systemctl stop NetworkManager
```

You also need to disable NetworkManager start at boot time by executing the following command.

```
sudo systemctl disable NetworkManager
```

Step 2: Find your wireless interface name and wireless network name

Run `iwconfig` to find the name of your wireless interface.

```
iwconfig
```

`wlan0` is a common name, but my wireless interface is called `wlp3s0`. You can also see that it's not associated with any access point right now.

```
linuxbabe@xenial:~$ iwconfig
enp12s0    no wireless extensions.

lo         no wireless extensions.

wlp3s0     IEEE 802.11  ESSID:off/any
Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
Retry short limit:7   RTS thr=2347 B   Fragment thr:off
Power Management:off

linuxbabe@xenial:~$
```

If your wireless interface isn't shown, perhaps you need to bring it up with the following command.

```
sudo ifconfig wlp3s0 up
```

Then find your wireless network name by scanning nearby networks with the command below. Replace wlp3s0 with your own interface name. ESSID means network name.

```
sudo iwlist wlp3s0 scan | grep ESSID
```

```
linuxbabe@xenial:~$ sudo iwlist wlp3s0 scan | grep ESSID
ESSID:"C8B5"
ESSID:"MERCURY_5322"
ESSID:""
ESSID:"Zhu"
ESSID:"ChinaNet-4xT9"
ESSID:"Xiaomi_\xE5\xAE\xA2\xE5\x8E\x85"
ESSID:"Laura12"
ESSID:"LinuxBabe.Com office network"
ESSID:"icongo-b5e281"
```

Step 3: Connect to Wi-fi network using wpa_supplicant

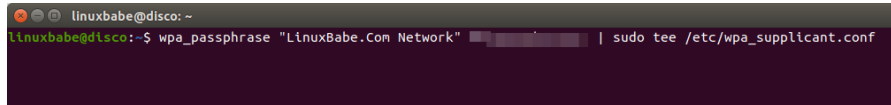
Now install wpa_supplicant on Ubuntu 16.04.

```
sudo apt install wpasupplicant
```

We need to create a file named wpa_supplicant.conf using the wpa_passphrase utility. wpa_supplicant.conf is the configuration file describing all networks that the user wants the computer to connect to. Run the following command to create this file. Replace ESSID and Wi-fi passphrase with your own.

```
wpa_passphrase your-ESSID your-passphrase |
```

```
sudo tee /etc/wpa_supplicant.conf
```



```
linuxbabe@disco: ~  
linuxbabe@disco:~$ wpa_passphrase "LinuxBabe.Com Network" | sudo tee /etc/wpa_supplicant.conf
```

Note that in the above screenshot, I wrapped my ESSID with double-quotes, because my ESSID contains a whitespace.

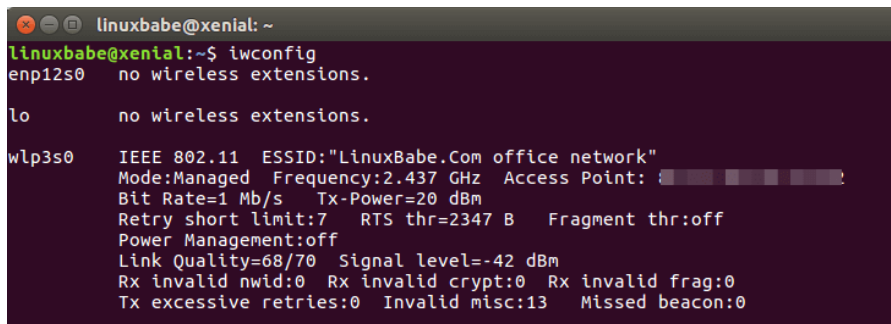
The output will be piped to `tee` which then write to `/etc/wpa_supplicant.conf` file. Now use the following command to connect your wireless card to wireless access point.

```
sudo wpa_supplicant -c /etc/wpa_supplicant.conf -i wlp3s0
```

By default, `wpa_supplicant` runs in the foreground. If the connection is completed, then open up another terminal window and run

```
iwconfig
```

You can see that the wireless interface is now associated with an access point.



```
linuxbabe@xenial: ~  
linuxbabe@xenial:~$ iwconfig  
enp12s0    no wireless extensions.  
  
lo         no wireless extensions.  
  
wlp3s0     IEEE 802.11  ESSID:"LinuxBabe.Com office network"  
           Mode:Managed  Frequency:2.437 GHz  Access Point: [REDACTED]  
           Bit Rate=1 Mb/s   Tx-Power=20 dBm  
           Retry short limit:7   RTS thr=2347 B   Fragment thr:off  
           Power Management:off  
           Link Quality=68/70  Signal level=-42 dBm  
           Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0  
           Tx excessive retries:0  Invalid misc:13  Missed beacon:0
```

You can press `CTRL+C` to stop the current `wpa_supplicant` process and run it in the background by adding `-B` option.

```
sudo wpa_supplicant -B -c /etc/wpa_supplicant.conf -i wlp3s0
```

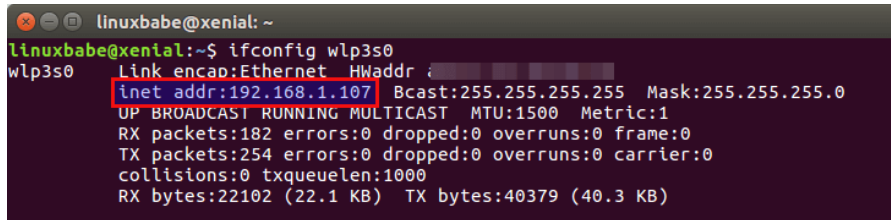
Although we're authenticated and connected to wireless network, but we don't have an IP address yet. To obtain a private IP address from DHCP server, use the following command:



```
sudo dhclient wlp3s0
```

Now your wireless interface has a private IP address, which can be shown with:

```
ifconfig wlp3s0
```

A terminal window titled 'linuxbabe@xenial: ~' showing the output of the 'ifconfig wlp3s0' command. The output displays network details for the wlp3s0 interface, including link encoding, hardware address, IP address (192.168.1.107), broadcast address, netmask, and various statistics like RX/TX packets and bytes. The IP address '192.168.1.107' is highlighted with a red box.

```
linuxbabe@xenial:~$ ifconfig wlp3s0
wlp3s0:  Link encap:Ethernet  HWaddr 88:8e:6d:48:4d:4d
          inet addr:192.168.1.107  Bcast:255.255.255.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:182 errors:0 dropped:0 overruns:0 frame:0
          TX packets:254 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:22102 (22.1 KB)  TX bytes:40379 (40.3 KB)
```

Now you can access the Internet. To release the private IP address, run

```
sudo dhclient wlp3s0 -r
```

Connecting to Hidden Wireless Network

If your wireless doesn't broadcast SSID, then you need to add the following line in `/etc/wpa_supplicant.conf` file.

```
scan_ssid=1
```

Like below:

```
network={
    ssid="LinuxBabe.Com office network"
    #psk="12345qwert"
    psk=68add4c5fee7dc3d0dac810f89b805d
    6d147c01e281f07f475a3e0195
    scan_ssid=1
}
```

Auto Connect on Startup

To automatically connect to wireless network at boot time, we need to edit the `wpa_supplicant.service` file. It's a good idea to copy the file from `/lib/systemd/system/` directory to `/etc/systemd/system/` directory, then edit it because we don't want newer version of `wpa_supplicant` to override our modifications.

```
sudo cp /lib/systemd/system/wpa_supplicant.  
service /etc/systemd/system/wpa_supplicant.  
service
```

```
sudo nano /etc/systemd/system/wpa_supplican  
t.service
```

Find the following line.

```
ExecStart=/sbin/wpa_supplicant -u -s -O /ru  
n/wpa_supplicant
```

Change it to the following. Obviously you need to change `wlp3s0` if that isn't your interface name.

```
ExecStart=/sbin/wpa_supplicant -u -s -c /et  
c/wpa_supplicant.conf -i wlp3s0
```

It's recommended to always try to restart `wpa_supplicant` when failure is detected. Add the following right below the `ExecStart` line.

```
Restart=always
```

If you can find the following line in this file, comment it out (Add the `#` character at the beginning of the line).

```
Alias=dbus-fi.w1.wpa_supplicant1.service
```

Save and close the file. Then enable wpa_supplicant service to start at boot time.

```
sudo systemctl enable wpa_supplicant.service
```

We also need to start dhclient at boot time to obtain a private IP address from DHCP server. This can be achieved by creating a systemd service unit for dhclient.

```
sudo nano /etc/systemd/system/dhclient.service
```

Put the following text into the file.

```
[Unit]
Description= DHCP Client
Before=network.target

[Service]
Type=forking
ExecStart=/sbin/dhclient wlp3s0 -v
ExecStop=/sbin/dhclient wlp3s0 -r
Restart=always

[Install]
WantedBy=multi-user.target
```

Save and close the file. Then enable this service.

```
sudo systemctl enable dhclient.service
```

How to Obtain a Static IP Address

Edit the dhclient configuration file.



```
sudo nano /etc/dhcp/dhclient.conf
```

Add the following line in the file. Replace the IP address with your preferred IP address. This makes dhclient request a static IP address from router.

```
interface "wlp3s0" {  
    send dhcp-requested-address 192.168.0.  
122;  
}
```

Save and close the file. Then restart dhclient service.

```
sudo systemctl restart dhclient
```

I found that this only works for the first time, or when the Ubuntu system boots up. Perhaps it's because of my router setup. If you run the `/sbin/dhclient wlp4s0` command multiple times (via the dhclient service), the router might refuse dhclient's request. Also note that iOS's hotspot doesn't support client requesting a particular IP address.

The better way to make sure your Ubuntu system have a static IP address is probably by logging into your router's management interface and assigning a static IP to the MAC address of your wireless card, if your router supports this feature.

Using a Hostname to Access Services on Ubuntu

Actually, you don't have to obtain a static IP address for your Ubuntu box. Ubuntu can use mDNS (Multicast DNS) to announce its hostname to the local network and clients can access services on your Ubuntu box with that hostname. This hostname can always be resolved to the IP address of your Ubuntu box, even if the IP address changes.

In order to use mDNS, you need to install avahi-daemon, which is an open-source implementation of mDNS/DNS-SD.

```
sudo apt install avahi-daemon
```

Start the service.

```
sudo systemctl start avahi-daemon
```

Enable auto-start at boot time.

```
sudo systemctl enable avahi-daemon
```

Avahi-daemon listens on UDP 5353, so you need to open this port in the firewall. If you use UFW, then run the following command.

```
sudo ufw allow 5353/udp
```

Then you should set a unique hostname for your Ubuntu box with the `hostnamectl` command. Replace `ubuntubox` with your preferred hostname, which should not be already taken by other devices in the local network.

```
sudo hostnamectl set-hostname ubuntubox
```

Now restart avahi-daemon.

```
sudo systemctl restart avahi-daemon
```

If you check the status with

```
systemctl status avahi-daemon
```

you can see the mDNS hostname, which ends with the `.local` domain.



```
avahi-daemon[5613]: New relevant interface enp5s0.IPv6 for mDNS.  
avahi-daemon[5613]: Joining mDNS multicast group on interface enp5s0.IPv4 with address 192.168.0.101.  
avahi-daemon[5613]: New relevant interface enp5s0.IPv4 for mDNS.  
avahi-daemon[5613]: Joining mDNS multicast group on interface lo.IPv4 with address 127.0.0.1.  
avahi-daemon[5613]: New relevant interface lo.IPv4 for mDNS.  
avahi-daemon[5613]: Network interface enumeration completed.  
avahi-daemon[5613]: Registering new address record for fe80::4d94:3401:e928:6f3d on enp5s0.*.  
avahi-daemon[5613]: Registering new address record for 192.168.0.101 on enp5s0.IPv4.  
avahi-daemon[5613]: Registering new address record for 127.0.0.1 on lo.IPv4.  
avahi-daemon[5613]: Server startup complete. Host name is ubuntu.local. Local service cookie is 958440470.
```

On the client computer, you also need to install an mDNS/DNS-SD software.

- Linux users should install `avahi-daemon`.
- Windows users need to enable the Bonjour service by either installing the [Bonjour print service](#) or installing [iTunes](#).
- On macOS, Bonjour is pre-installed.

Now you can access services by using the `ubuntubox.local` hostname, eliminating the need to check and type IP address.

Unblock Wifi on Raspberry Pi

The Ubuntu ARM OS for Raspberry Pi blocks wireless interface by default. You need to unblock it with:

```
sudo rfkill unblock wifi
```

To unblock it at boot time, edit the `wpa_supplicant.service` systemd unit.

```
sudo nano /etc/systemd/system/wpa_supplicant.service
```

Change the `ExecStart` parameter to the following.

```
ExecStart=/bin/bash -c '/usr/sbin/rfkill unblock wifi && /sbin/wpa_supplicant -u -s -c /etc/wpa_supplicant.conf -i wlan0'
```

Save and close the file.

If that doesn't work, you can mask the `systemd-rfkill.socket` system unit.

```
sudo systemctl mask systemd-rfkill.socket
```

Also, create the `rc.local` file.

```
sudo nano /etc/rc.local
```

Add the following lines in this file.

```
#!/bin/bash
/usr/sbin/rfkill unblock wifi
exit 0
```

Make it executable.

```
sudo chmod +x /etc/rc.local
```

In my test, the `rfkill unblock` command in the `systemd` unit doesn't work, but it works in the `/etc/rc.local` file. Perhaps the timing is important. It could be that the `rfkill unblock` command should run after the `wpa_supplicant` command.

Note that if you have installed a desktop environment, there's probably a network manager running that can interfere with connection. You need to disable it. For example, I use the lightweight LXQT desktop environment on Raspberry Pi and need to disable `connman.service`.

```
sudo systemctl disable connman.service
```

Recommended Reading

- [How to Use Systemd on Linux – Manage Services, Run Levels and Logs](#)

Multiple Wi-Fi Networks

The `/etc/wpa_supplicant.conf` configuration file can include multiple Wi-Fi networks. `wpa_supplicant` will automatically select the best network based on the order of network blocks in the configuration file, network security level, and signal strength.

To add a second Wi-Fi network, run


```
wpa_passphrase your-ESSID your-wifi-passphrase | sudo tee -a /etc/wpa_supplicant.conf
```

Note that you need to use the `-a` option with the `tee` command, which will append, instead of deleting the original content, the new Wifi-network to the file.





Wrapping Up

I hope this tutorial helped you set up Ubuntu server 16.04 wifi from the command line. As always, if you found this post useful, then [subscribe to our free newsletter](#). You can also follow us on [Google+](#), [Twitter](#) or [like our Facebook page](#).

Rate this tutorial

 [Total: 123 Average: 4.3]

You may also like:

 2 Ways to...	 Set Up ParseD...	 Connect to Wi-F...	 How to Install...
