Wi-Fi

This page describes how to set up a wireless network connection with WPA encryption.

You may want to use Network setup-scripts to configure your wifi quickly. If it does not work or does not meet your requirement, please follow the steps outlined on this page.

NetworkManager provides automatic detection and configuration for systems to connect to networks. Both iwd and `wpa_supplicant (https://pkgs.alpinelinux.org/packages?name=wpa_supplicant&branch=edge&repo=&arch=x86_64&maintainer=)` can be used with NetworkManager.

Tip: For wireless daemon choose between `iwd (https://pkgs.alpinelinux.org/packages?name=iwd&branch=edge&repo=&arch=x86_64&maintainer=)` and `wpa_supplicant (https://pkgs.alpinelinux.org/packages?name=wpa_supplicant&branch=edge&repo=&arch=x86_64&maintainer=)`. You can't use both.

Prerequisites

Working wireless drivers.

Note: in most cases installing `linux-firmware (https://pkgs.alpinelinux.org/packages?name=linux-firmware&branch=edge&repo=&arch=x86_64&maintainer=)` should get you the required drivers. Installation of this package can produce errors on diskless or data disk modes. If such errors occur, switch to system disk mode.

If you are using a **Broadcom chipset**, see the Broadcom Wi-Fi section.

iwd

Main article: iwd

Iwd (iNet wireless daemon) is a wireless daemon written by Intel and aiming at replacing `wpa_supplicant (https://pkgs.alpinelinux.org/packages?name=wpa_supplicant&branch=edge&repo=&arch=&maintainer=)`. The core goal of the project is to optimize resource utilization by not depending on any external libraries and instead utilizing features provided by the Linux Kernel to the maximum extent possible. See the iwd for more information. `iwd (https://pkgs.alpinelinux.org/packages?name=iwd&branch=edge&repo=&arch=&maintainer=)` is supported since Alpine Linux 3.10 (https://alpinelinux.org/posts/Alpine-3.10.0-released.html).

wpa_supplicant

To get started install `wpa_supplicant (https://pkgs.alpinelinux.org/packages?name=wpa_supplicant&branch=edge&repo=&arch=x86_64&maintainer=)`

```
# apk add wpa_supplicant
```

To list your available network interfaces:

Note: Refer to #Troubleshooting section to see if you don't see any wireless interfaces (e.g. `wlan0`). You probably need to load and/or install drivers/firmware.

```
ip link
```

or

```
ip a
```

Bring up the desired interface:

```
# ip link set wlan0 up
```

Note: If this errors with `ioctl 0x8914 failed: No error information`, that's `busybox ip`'s way of saying your wireless radio is rfkill'd. See the Rfkill section for information on how to unblock your wireless radio.

Use this command to add your Wi-Fi network to wpa_supplicant:

```
# wpa_passphrase 'ExampleWifiSSID' 'ExampleWifiPassword' > /etc/wpa_supplicant/wpa_
```

(Access point not broadcasting its SSID requires additional line `scan_ssid=1` in the file `wpa_supplicant.conf`)

Note: the Wi-Fi SSID and password are case sensitive and the single quote before and after the SSID and password need to be there

Start wpa_supplicant in the foreground to make sure the connection succeeds.

```
# wpa_supplicant -i wlan0 -c /etc/wpa_supplicant/wpa_supplicant.conf
```

If all is well, run it as a daemon in the background by setting the `-B` option.

```
# wpa_supplicant -B -i wlan0 -c /etc/wpa_supplicant/wpa_supplicant.conf
```

Configure the interface with an IP address.

```
# udhcpc -i wlan0
```

Sanity check: the interface should have a `inet` address.

```
$ ip addr show wlan0
```

Automatic Configuration on System Boot

Add a entry for the desired interface (e.g. `wlan0`):

Contents of `/etc/network/interfaces`

```
auto wlan0
iface wlan0 inet dhcp
```

Note: Dont remove or comment out the **auto lo** entry

Sanity check: Make sure `/etc/wpa_supplicant/wpa_supplicant.conf` is the correct configuration for the wireless access point you want to connect to.

Bring the interface down.

```
# ip link set wlan0 down
```

Manually restart (or **start**) **networking**.

```
# rc-service networking --quiet restart &
```

If all is well (feel free to confirm with the sanity checks),

Configure wpa_supplicant to start automatically on boot:

```
# rc-update add wpa_supplicant boot
```

Also make sure **networking** is set to automatically start on boot:

```
# rc-update add networking boot
```

Optional security precaution:

By default `wpa_supplicant (https://pkgs.alpinelinux.org/packages?name=wpa_supplicant&branch=edge&repo=&arch=x86_64&maintainer=)` will store your Wi-Fi password in plain text:

Contents of `(Example) /etc/wpa_supplicant/wpa_supplicant.conf`

```
network={
    ssid="<YourSSIDShouldBeHere>"
        #psk="<YourPasswordShouldBeHereInPlainText>"
    psk=<RandomLettersAndNumbersShouldBeHere>
}
```

this is not necessary and `wpa_supplicant (https://pkgs.alpinelinux.org/packages?name=wpa_supplicant&branch=edge&repo=&arch=x86_64&maintainer=)` should funtion just fine without it, if you dont want your stored password in plain text just delete the line with `#psk="<YourPasswordShouldBeHereInPlainText>"` on it.

Launching udhcpc through wpa_cli actions

With the above configuration, udhcpc will only run once at boot. If the Wifi isn't available then, or the network changes after booting, udhcpc needs to be notified. You can automatically notify udhcpc of network changes by using a wpa_cli action file, such as the one installed by default at `/etc/wpa_supplicant/wpa_cli.sh`.

To manually start a wpa_cli daemon with an action file, use the `-a` option:

```
# wpa_cli -a /etc/wpa_supplicant/wpa_cli.sh
```

To do this automatically, use the `wpa_cli` service included in `wpa_supplicant-openrc (https://pkgs.alpinelinux.org/packages?name=wpa_supplicant-openrc&branch=edge&repo=&arch=x86_64&maintainer=)`:

Contents of `/etc/conf.d/wpa_cli`

```
WPACLI_OPTS="-a /etc/wpa_supplicant/wpa_cli.sh"
```

```
# rc-update add wpa_cli boot
```

Troubleshooting

checking network cards

```
$ cat /proc/net/dev
```

lists the network interfaces that are detected. If the expected interfaces are not available, Check what network hardware chip you have using lspci or lsusb:

```
$ lspci -nn
```

```
$ lsusb
```

Refer hardware management, if the default outputs shown above lacks sufficient information.

Check what driver the card uses and modprobe it. Check that the card is in master mode. Check what driver you need on the b43 compatibility page (https://wireless.wiki.kernel.org/en/users/Drivers/b43#list_of_hardware)

Broadcom Wi-Fi Chipset Users

The Broadcom chipset is quite popular among older computers. The b43 driver is included in the linux-lts or linux-edge kernel packages. However, you might need to compile the firmware manually for this chipset as it is not included in linux-firmware for some cards.

To check what broadcom chip you have using lspci:

```
$ lspci -nn -d 14e4:
```

B43

Download firmware cutter.

```
$ apk add b43-fwcutter
```

Now we have everything to download the proprietary driver and extract the firmware from it:

```
# export FIRMWARE_INSTALL_DIR="/lib/firmware"
$ wget http://www.lwfinger.com/b43-firmware/broadcom-wl-5.100.138.tar.bz2
$ tar xjf broadcom-wl-5.100.138.tar.bz2
$ b43-fwcutter -w "$FIRMWARE_INSTALL_DIR" broadcom-wl-5.100.138/linux/wl_apsta.o
```

More information can be found here (http://linuxwireless.sipsolutions.net/en/users/Drivers/b43/#Other_distributions_not_mentioned_above).

Now you need to use modprobe so the device will show up:

```
# modprobe b43
```

Now continue with the normal instructions.

wl

First install the software we need to build a driver^ (https://unix.stackexchange.com/questions/606073/how-to-build-kernel-modules-in-alpine-3-12):

```
apk add git alpine-sdk linux-headers
```

Then install the driver build repo (this is archived, however it's legacy so that doesn't matter):

```
git clone https://github.com/antoineco/broadcom-wl
cd broadcom-wl
```

Then follow the build instructions listed in the git repo:

```
make
doas make install
doas depmod -A
doas modprobe wl
```

reboot and you have a working wl driver!

Now continue with the normal instructions.

Check dmesg

Run dmesg and check for errors related to the wireless interface. Usually, dmesg gives maximum information related to network.

```
# dmesg
```

Rfkill

Many laptops have a hardware button (or switch) to turn off wireless card, however, the card can also be blocked by kernel. This can be changed using rfkill. To show the current of your Wi-Fi:

```
~$ rfkill list
0: phy0: wlan
    Soft blocked: no
    Hard blocked: no
```

If the card is hard-blocked, use the hardware button or switch to unblock it. If the card is not hard-blocked but soft-blocked, use the following command:

```
# rfkill unblock wifi
```

See Also

- Networking
- iwd
- NetworkManager
- PostmarketOS Wiki (https://wiki.postmarketos.org/wiki/WiFi)
- Alpine setup scripts
- Archwiki (https://wiki.archlinux.org/title/Network_configuration/Wireless)
- Gentoo Wiki (https://wiki.gentoo.org/wiki/Wi-Fi)