

Install software



Next up we install the software onto the Pi that will act as the 'hostap' (host access point) **You need internet access for this step so make sure that Ethernet connection is up!**

```
sudo apt-get update  
sudo apt-get install hostapd isc-dhcp-server
```

(You may need to **sudo apt-get update** if the Pi can't seem to get to the apt-get repositories)

A screenshot of a PuTTY terminal window titled "COM3 - PuTTY". The terminal shows the output of the command "sudo apt-get install hostapd udhcpd". It starts with a notice about software configuration, followed by the command execution. The output shows that busybox is an extra package to be installed along with the requested packages. It then lists the new packages (busybox, hostapd, udhcpd) and the disk space requirements. The user is prompted to continue, and they respond with 'Y'. The terminal then shows the download progress for each package from the Raspbian mirror, totaling 878 kB fetched in 7 seconds.

```
COM3 - PuTTY  
  
NOTICE: the software on this Raspberry Pi has not been fully configured. Please  
run 'sudo raspi-config'  
  
pi@raspberrypi:~$ sudo apt-get install hostapd udhcpd  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  busybox  
The following NEW packages will be installed:  
  busybox hostapd udhcpd  
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.  
Need to get 878 kB of archives.  
After this operation, 1,751 kB of additional disk space will be used.  
Do you want to continue [Y/n]? Y  
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main busybox armhf 1:1  
.20.0-7 [438 kB]  
Get:2 http://mirrordirector.raspbian.org/raspbian/ wheezy/main hostapd armhf 1:1  
.0-3 [419 kB]  
Get:3 http://mirrordirector.raspbian.org/raspbian/ wheezy/main udhcpd armhf 1:1.  
20.0-7 [20.9 kB]  
Fetched 878 kB in 7s (111 kB/s)
```

(text above shows udhcpd but that doesnt work as well as isc-dhcp-server, still, the output should look similar)

Set up DHCP server

Next we will edit /etc/dhcp/dhcpd.conf, a file that sets up our DHCP server - this allows wifi connections to automatically get IP addresses, DNS, etc.

Run this command to edit the file

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

Find the lines that say

```
option domain-name "example.org";  
option domain-name-servers ns1.example.org, ns2.example.org;
```

and change them to add a # in the beginning so they say

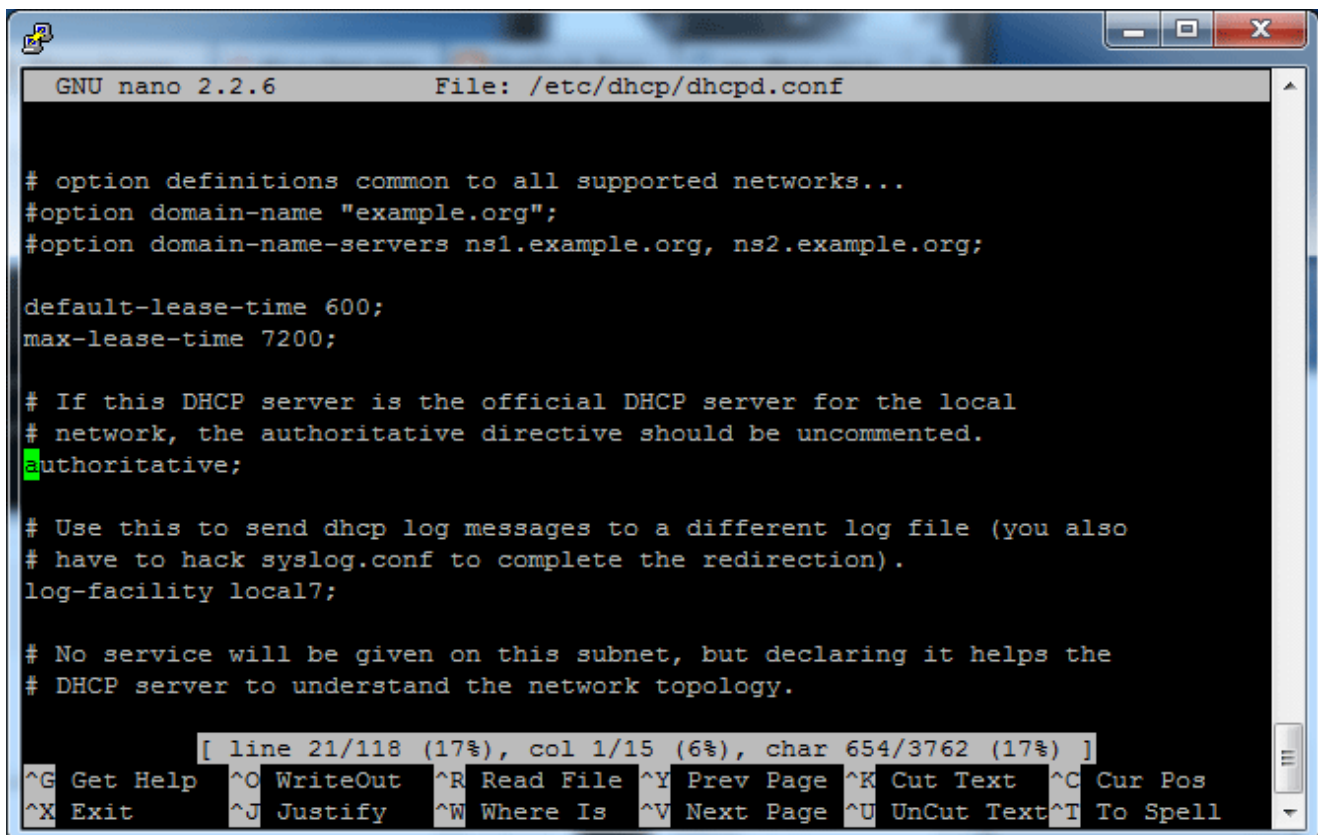
```
#option domain-name "example.org";  
#option domain-name-servers ns1.example.org, ns2.example.org;
```

Find the lines that say

```
# If this DHCP server is the official DHCP server for the local  
# network, the authoritative directive should be uncommented.  
#authoritative;
```

and remove the # so it says

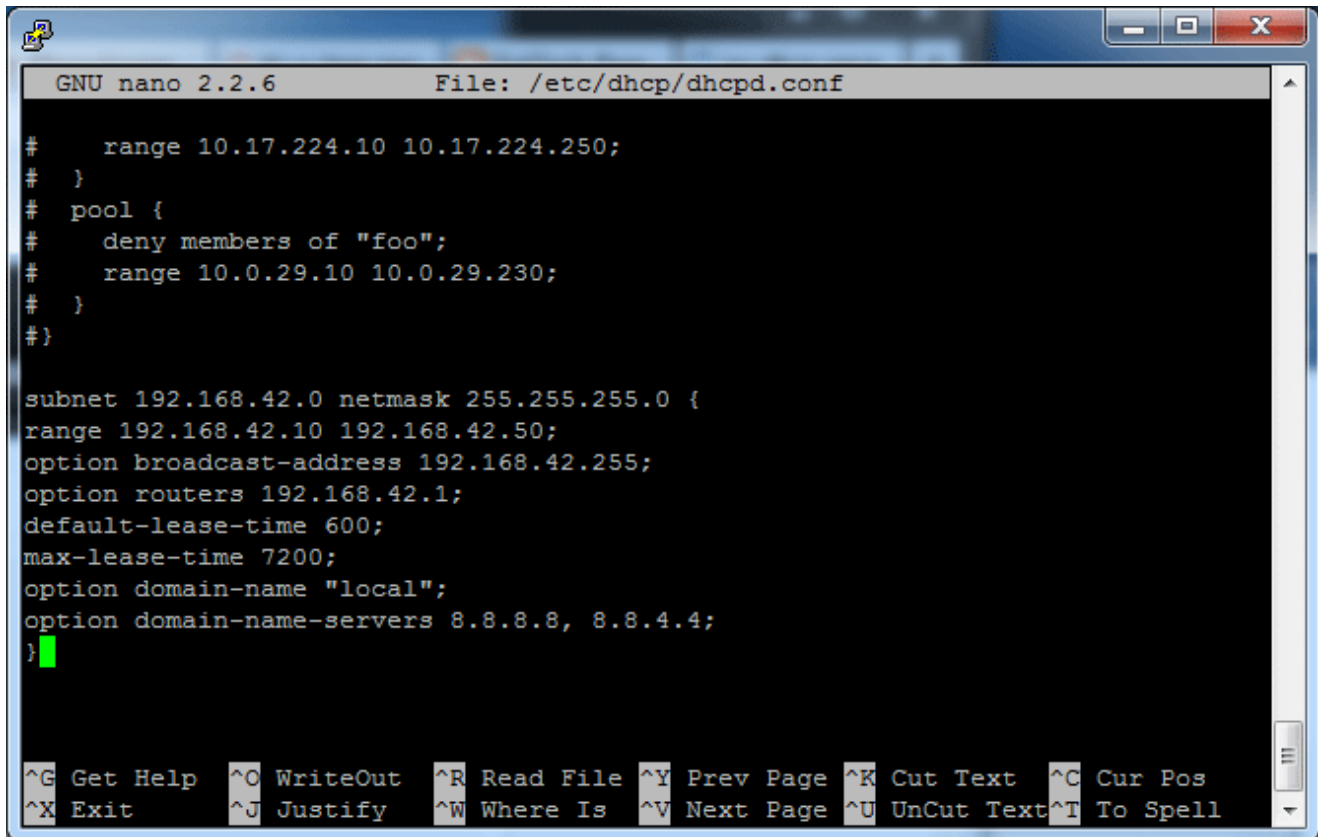
```
# If this DHCP server is the official DHCP server for the local  
# network, the authoritative directive should be uncommented.  
authoritative;
```



```
GNU nano 2.2.6      File: /etc/dhcp/dhcpd.conf  
  
# option definitions common to all supported networks...  
#option domain-name "example.org";  
#option domain-name-servers ns1.example.org, ns2.example.org;  
  
default-lease-time 600;  
max-lease-time 7200;  
  
# If this DHCP server is the official DHCP server for the local  
# network, the authoritative directive should be uncommented.  
#authoritative;  
  
# Use this to send dhcp log messages to a different log file (you also  
# have to hack syslog.conf to complete the redirection).  
log-facility local7;  
  
# No service will be given on this subnet, but declaring it helps the  
# DHCP server to understand the network topology.  
  
[ line 21/118 (17%), col 1/15 (6%), char 654/3762 (17%) ]  
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos  
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text ^T To Spell
```

Then scroll down to the bottom and add the following lines

```
subnet 192.168.42.0 netmask 255.255.255.0 {  
    range 192.168.42.10 192.168.42.50;  
    option broadcast-address 192.168.42.255;  
    option routers 192.168.42.1;  
    default-lease-time 600;  
    max-lease-time 7200;  
    option domain-name "local";  
    option domain-name-servers 8.8.8.8, 8.8.4.4;  
}
```



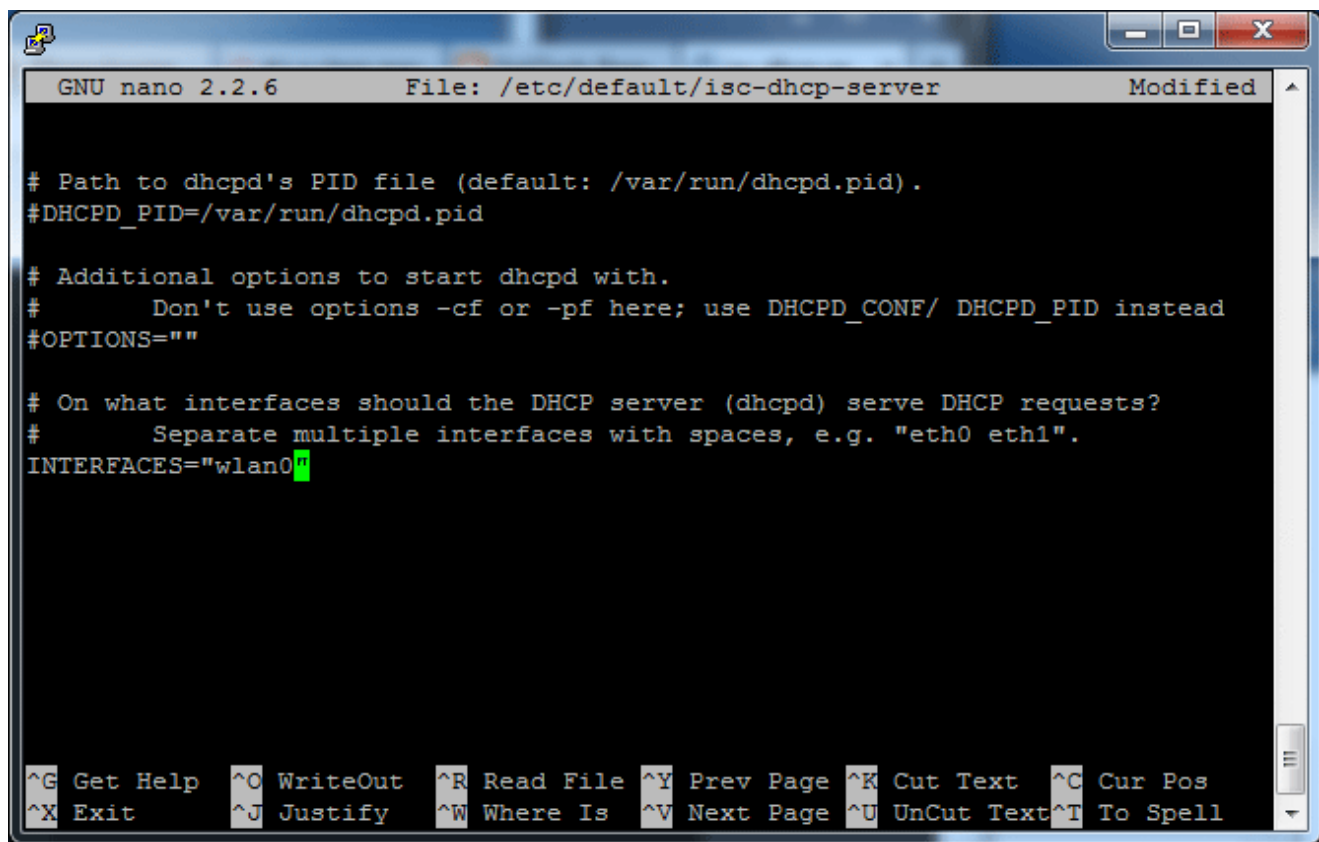
```
GNU nano 2.2.6      File: /etc/dhcp/dhcpd.conf  
  
#       range 10.17.224.10 10.17.224.250;  
#   }  
#   pool {  
#       deny members of "foo";  
#       range 10.0.29.10 10.0.29.230;  
#   }  
#}  
  
subnet 192.168.42.0 netmask 255.255.255.0 {  
range 192.168.42.10 192.168.42.50;  
option broadcast-address 192.168.42.255;  
option routers 192.168.42.1;  
default-lease-time 600;  
max-lease-time 7200;  
option domain-name "local";  
option domain-name-servers 8.8.8.8, 8.8.4.4;  
}  
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos  
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text ^T To Spell
```

Save the file by typing in **Control-X** then **Y** then **return**

Run

```
sudo nano /etc/default/isc-dhcp-server
```

and scroll down to **INTERFACES=""** and update it to say **INTERFACES="wlan0"**



```
GNU nano 2.2.6      File: /etc/default/isc-dhcp-server      Modified ^
# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPD_PID=/var/run/dhcpd.pid

# Additional options to start dhcpd with.
#       Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES="wlan0"
```

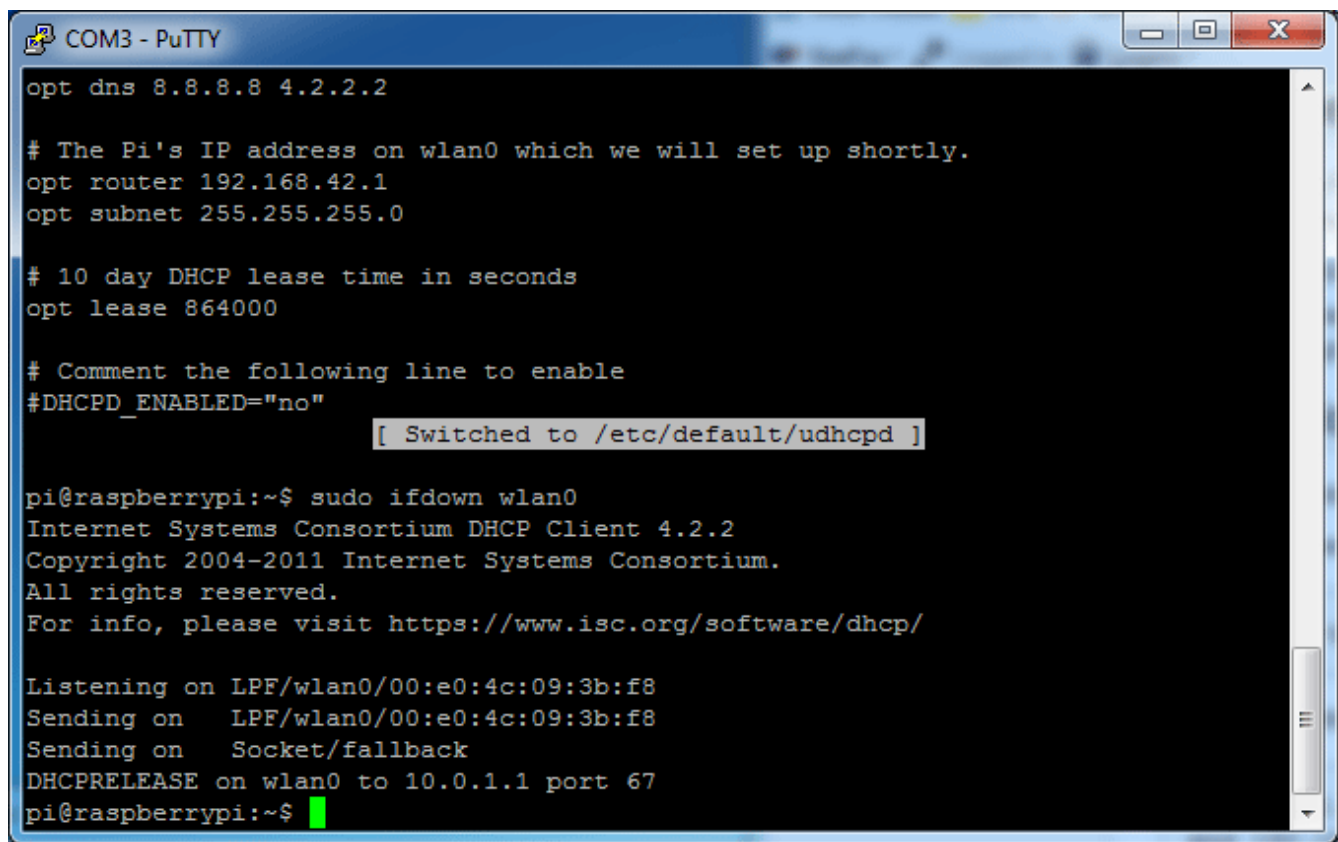
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

close and save the file

Set up wlan0 for static IP

If you happen to have wlan0 active because you set it up, run **sudo ifdown wlan0**

There's no harm in running it if you're not sure



```
COM3 - PuTTY
opt dns 8.8.8.8 4.2.2.2

# The Pi's IP address on wlan0 which we will set up shortly.
opt router 192.168.42.1
opt subnet 255.255.255.0

# 10 day DHCP lease time in seconds
opt lease 864000

# Comment the following line to enable
#DHCPD_ENABLED="no"
[ Switched to /etc/default/udhcpd ]

pi@raspberrypi:~$ sudo ifdown wlan0
Internet Systems Consortium DHCP Client 4.2.2
Copyright 2004-2011 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/wlan0/00:e0:4c:09:3b:f8
Sending on   LPF/wlan0/00:e0:4c:09:3b:f8
Sending on   Socket/fallback
DHCPRELEASE on wlan0 to 10.0.1.1 port 67
pi@raspberrypi:~$
```

Next we will set up the **wlan0** connection to be static and incoming. run **sudo nano /etc/network/interfaces** to edit the file

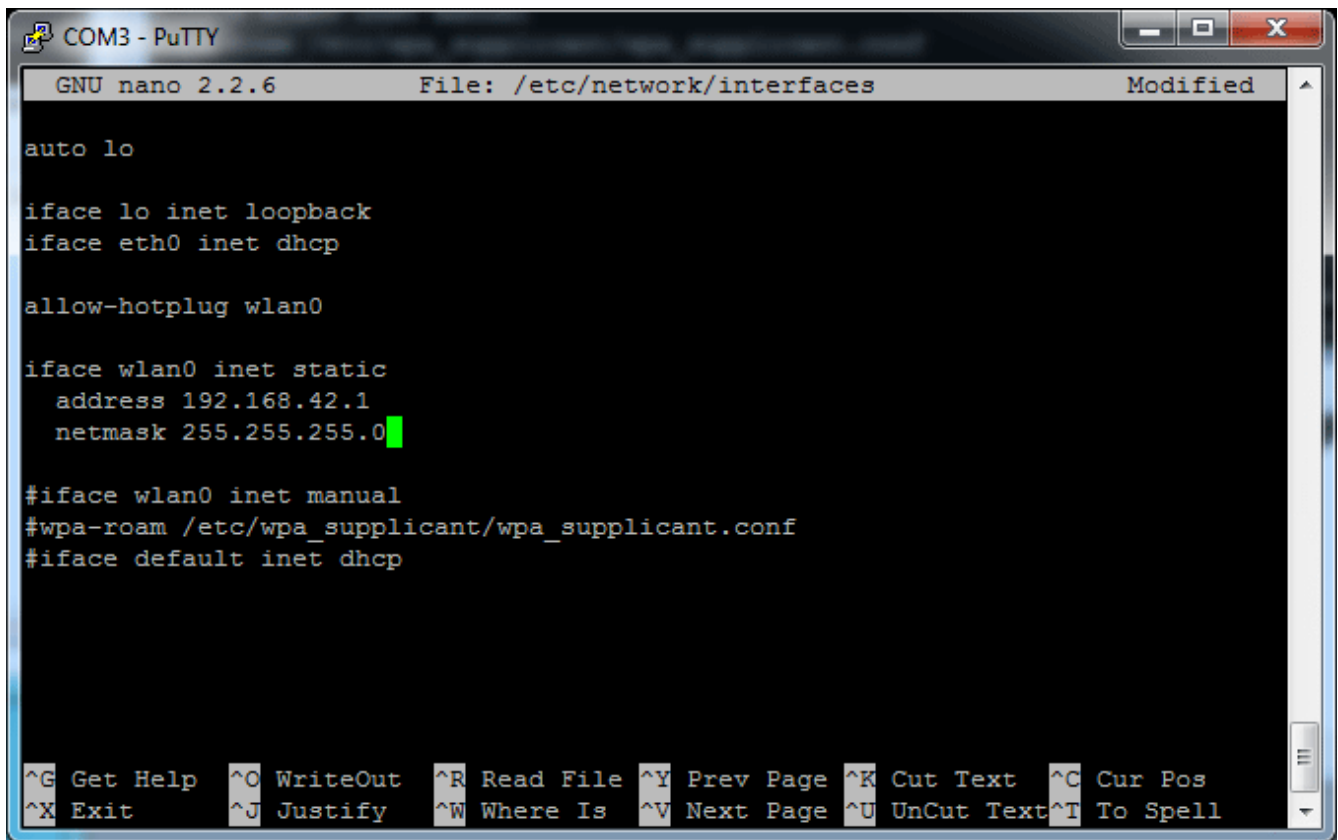
Find the line **auto wlan0** and add a **#** in front of the line, and in front of every line afterwards. If you don't have that line, just make sure it looks like the screenshot below in the end! Basically just remove any old **wlan0** configuration settings, we'll be changing them up

Depending on your existing setup/distribution there might be more or less text and it may vary a little bit

Add the lines

```
iface wlan0 inet static
    address 192.168.42.1
    netmask 255.255.255.0
```

After **allow-hotplug wlan0** - see below for an example of what it should look like. Any other lines afterwards should have a **#** in front to disable them



```
COM3 - PuTTY
GNU nano 2.2.6      File: /etc/network/interfaces      Modified
auto lo

iface lo inet loopback
iface eth0 inet dhcp

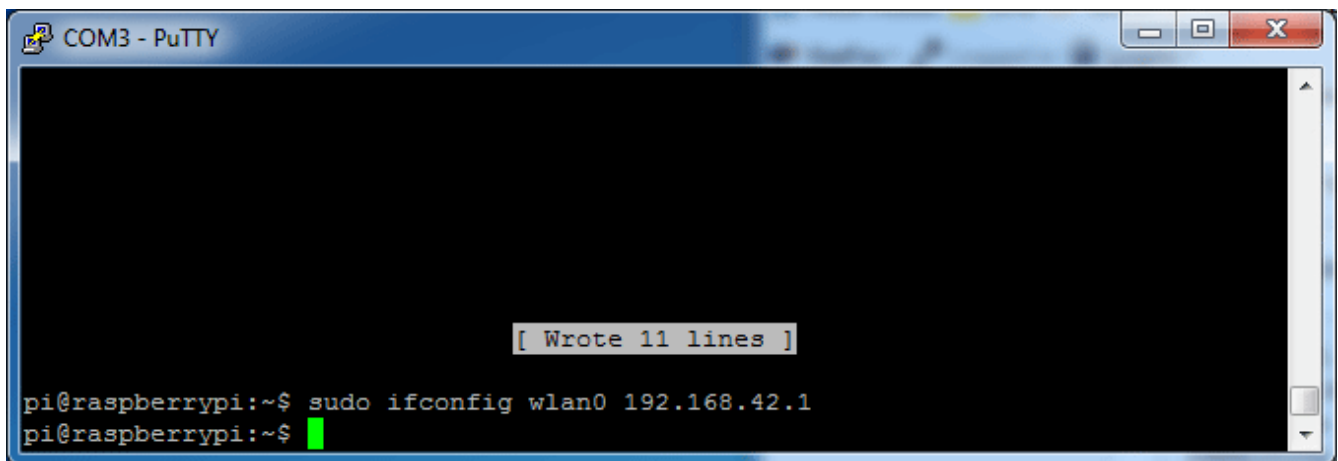
allow-hotplug wlan0

iface wlan0 inet static
    address 192.168.42.1
    netmask 255.255.255.0
#iface wlan0 inet manual
#wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
#iface default inet dhcp

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

Save the file (Control-X Y <return>)

Assign a static IP address to the wifi adapter by running
sudo ifconfig wlan0 192.168.42.1



```
COM3 - PuTTY
[ Wrote 11 lines ]
pi@raspberrypi:~$ sudo ifconfig wlan0 192.168.42.1
pi@raspberrypi:~$
```

Configure Access Point

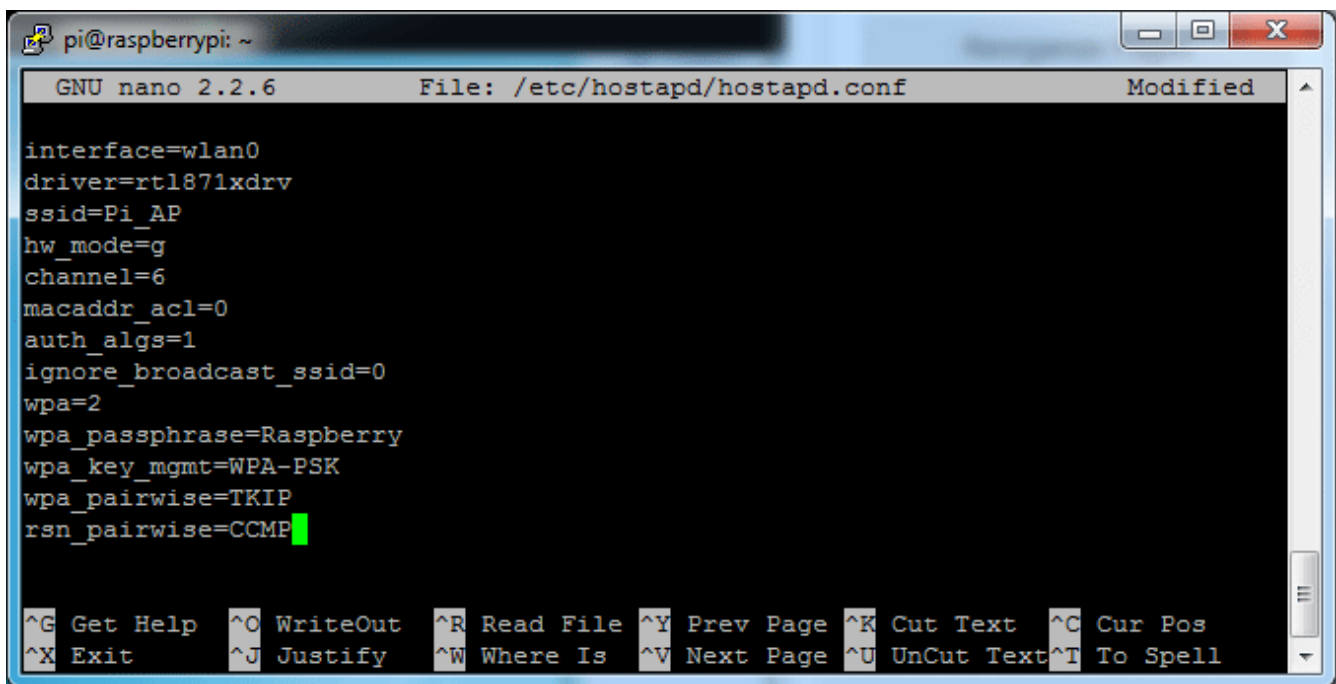
Now we can configure the access point details. We will set up a password-protected network so only people with the password can connect.

Create a new file by running **sudo nano /etc/hostapd/hostapd.conf**

Paste the following in, you can change the text after **ssid=** to another name, that will be the network broadcast name. The password can be changed with the text after **wpa_passphrase=**

```
interface=wlan0
driver=rtl871xdrv
ssid=Pi_AP
hw_mode=g
channel=6
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=Raspberry
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

If you are not using the Adafruit wifi adapters, you may have to change the **driver=rtl871xdrv** to say **driver=nl80211** or something, we don't have tutorial support for that tho, YMMV!



```
pi@raspberrypi: ~
GNU nano 2.2.6      File: /etc/hostapd/hostapd.conf      Modified

interface=wlan0
driver=rtl871xdrv
ssid=Pi_AP
hw_mode=g
channel=6
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=Raspberry
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text ^T To Spell
```

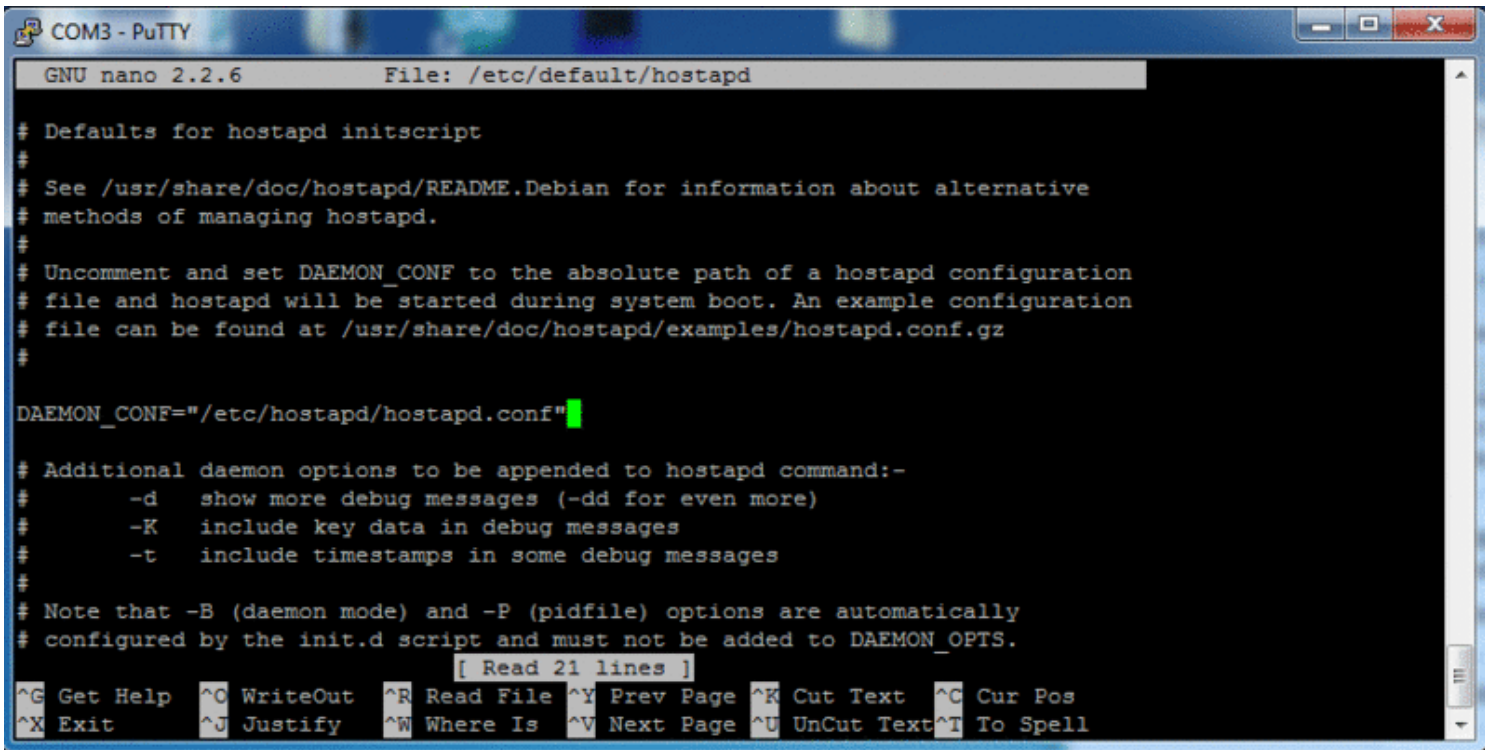
Save as usual. Make sure each line has no extra spaces or tabs at the end or beginning - this file is pretty picky!

Now we will tell the Pi where to find this configuration file. Run **sudo nano /etc/default/hostapd**

Find the line **#DAEMON_CONF=""** and edit it so it says **DAEMON_CONF="/etc/hostapd/hostapd.conf"**

Don't forget to remove the **#** in front to activate it!

Then save the file



```
COM3 - PuTTY
GNU nano 2.2.6      File: /etc/default/hostapd

# Defaults for hostapd initscript
#
# See /usr/share/doc/hostapd/README.Debian for information about alternative
# methods of managing hostapd.
#
# Uncomment and set DAEMON_CONF to the absolute path of a hostapd configuration
# file and hostapd will be started during system boot. An example configuration
# file can be found at /usr/share/doc/hostapd/examples/hostapd.conf.gz
#
DAEMON_CONF="/etc/hostapd/hostapd.conf"

# Additional daemon options to be appended to hostapd command:-
#
#   -d   show more debug messages (-dd for even more)
#   -K   include key data in debug messages
#   -t   include timestamps in some debug messages
#
# Note that -B (daemon mode) and -P (pidfile) options are automatically
# configured by the init.d script and must not be added to DAEMON_OPTS.
[ Read 21 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text ^T To Spell
```

Configure Network Address Translation

Setting up NAT will allow multiple clients to connect to the WiFi and have all the data 'tunneled' through the single Ethernet IP. (But you should do it even if only one client is going to connect)

Run **sudo nano /etc/sysctl.conf**

Scroll to the bottom and add

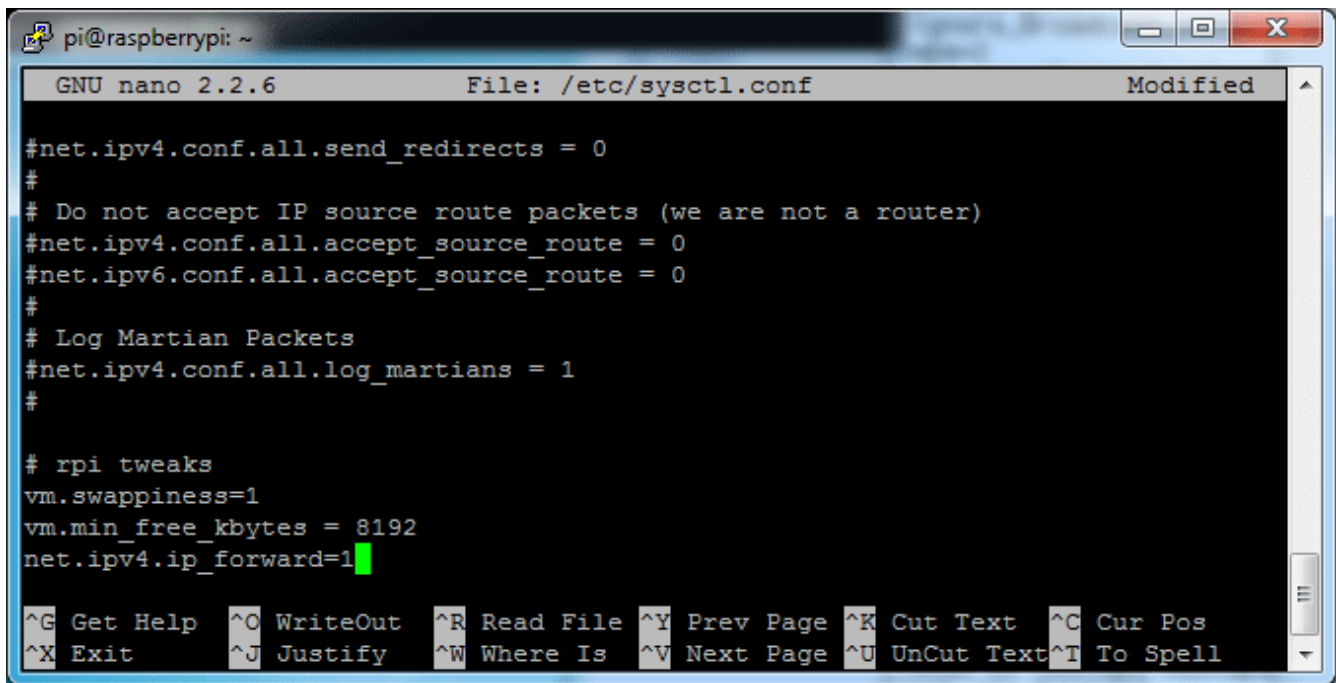
```
net.ipv4.ip_forward=1
```

on a new line. Save the file. This will start IP forwarding on boot up

Also run

```
sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
```

to activate it immediately

A screenshot of a terminal window on a Raspberry Pi. The window title is 'pi@raspberrypi: ~'. The terminal shows the GNU nano 2.2.6 editor editing the file /etc/sysctl.conf. The file content includes network settings: #net.ipv4.conf.all.send_redirects = 0, # Do not accept IP source route packets (we are not a router), #net.ipv4.conf.all.accept_source_route = 0, #net.ipv6.conf.all.accept_source_route = 0, # Log Martian Packets, #net.ipv4.conf.all.log_martians = 1, # rpi tweaks, vm.swappiness=1, vm.min_free_kbytes = 8192, and net.ipv4.ip_forward=1. The bottom status bar shows various keyboard shortcuts like ^G Get Help, ^O WriteOut, ^R Read File, etc.

```
pi@raspberrypi: ~
GNU nano 2.2.6      File: /etc/sysctl.conf      Modified

#net.ipv4.conf.all.send_redirects = 0
#
# Do not accept IP source route packets (we are not a router)
#net.ipv4.conf.all.accept_source_route = 0
#net.ipv6.conf.all.accept_source_route = 0
#
# Log Martian Packets
#net.ipv4.conf.all.log_martians = 1
#
# rpi tweaks
vm.swappiness=1
vm.min_free_kbytes = 8192
net.ipv4.ip_forward=1

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text ^T To Spell
```

Run the following commands to create the network translation between the ethernet port **eth0** and the wifi port **wlan0**

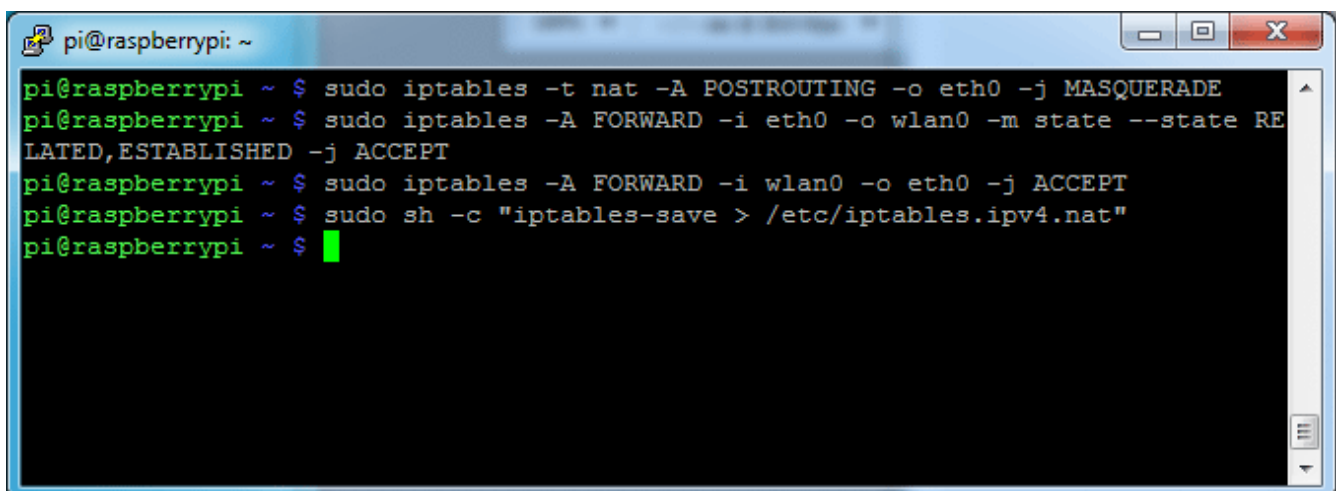
```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
```

You can check to see whats in the tables with

```
sudo iptables -t nat -S
sudo iptables -S
```

To make this happen on reboot (so you don't have to type it every time) run

```
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

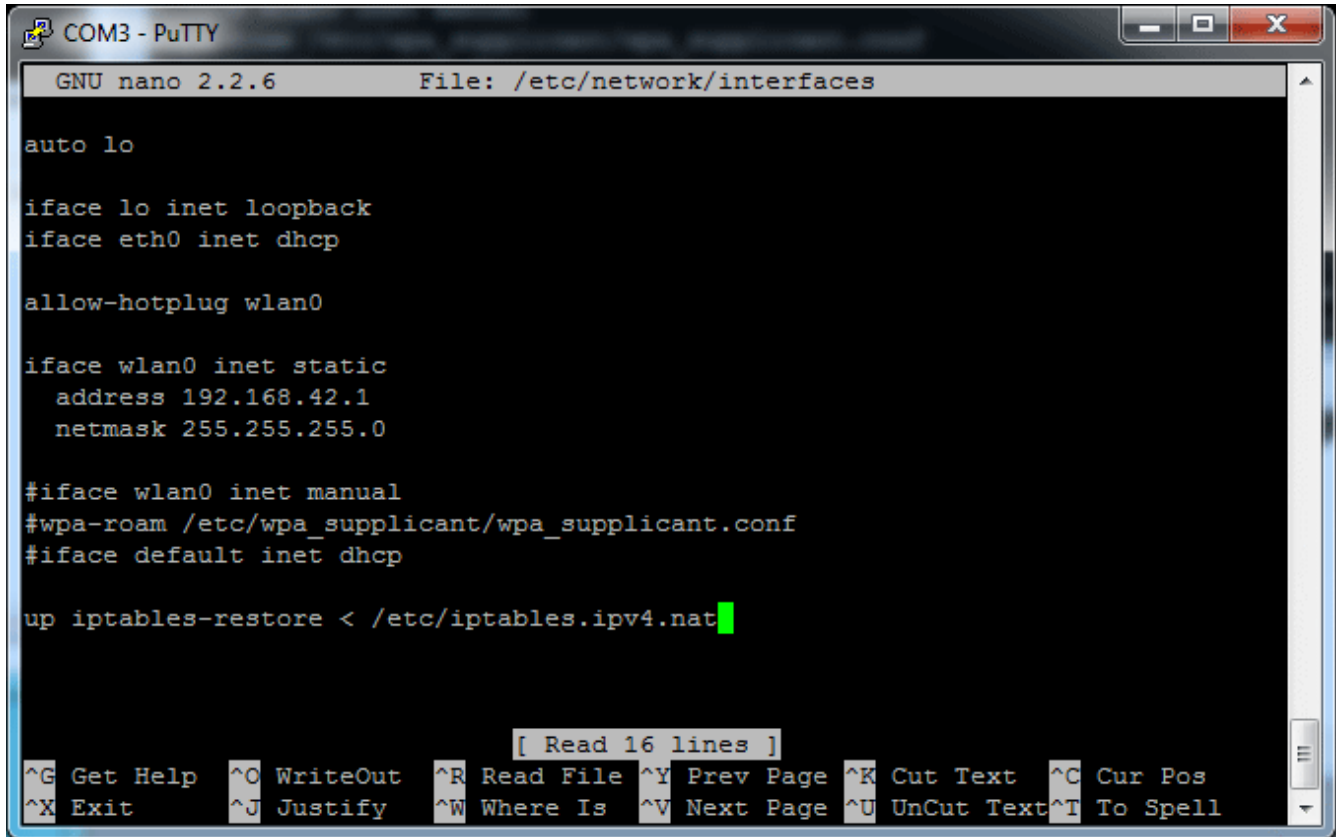
A screenshot of a terminal window on a Raspberry Pi. The window title is 'pi@raspberrypi: ~'. The terminal shows the execution of four commands: 'sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE', 'sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT', 'sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT', and 'sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"'. The prompt 'pi@raspberrypi ~ \$' is visible before each command.

```
pi@raspberrypi ~ $ sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
pi@raspberrypi ~ $ sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state RE
RELATED,ESTABLISHED -j ACCEPT
pi@raspberrypi ~ $ sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
pi@raspberrypi ~ $ sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
pi@raspberrypi ~ $
```

run **sudo nano /etc/network/interfaces** and add

```
up iptables-restore < /etc/iptables.ipv4.nat
```

to the very end



```
COM3 - PuTTY
GNU nano 2.2.6      File: /etc/network/interfaces

auto lo

iface lo inet loopback
iface eth0 inet dhcp

allow-hotplug wlan0

iface wlan0 inet static
    address 192.168.42.1
    netmask 255.255.255.0

#iface wlan0 inet manual
#wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
#iface default inet dhcp

up iptables-restore < /etc/iptables.ipv4.nat
```

Update hostapd

Before we can run the access point software, we have to update it to a version that supports the WiFi adapter. First get the new version by typing in

```
wget http://adafruit-download.s3.amazonaws.com/adafruit_hostapd_14128.zip
```

to download the new version (check the next section for how to compile your own updated **hostapd**) then

```
unzip adafruit_hostapd_14128.zip
```

to uncompress it. Move the old version out of the way with

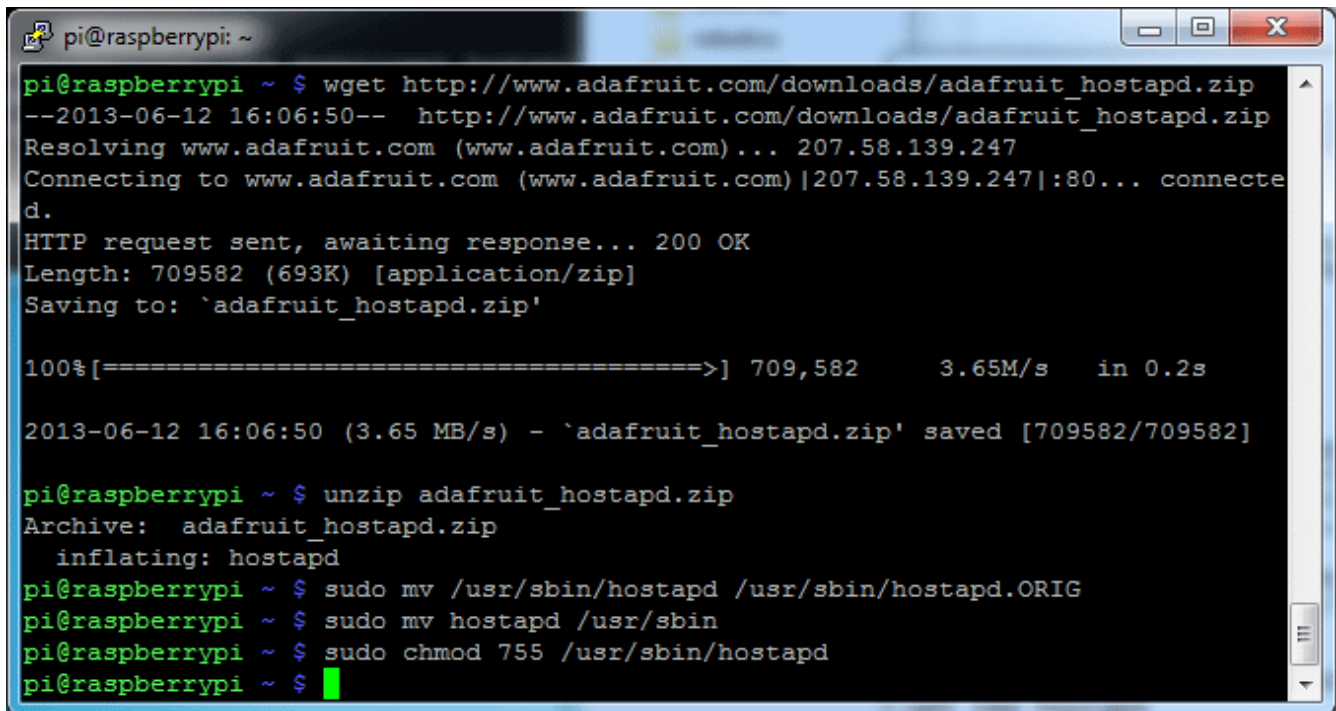
```
sudo mv /usr/sbin/hostapd /usr/sbin/hostapd.ORIG
```

And move the new version back with

```
sudo mv hostapd /usr/sbin
```

set it up so its valid to run with

```
sudo chmod 755 /usr/sbin/hostapd
```

A terminal window titled 'pi@raspberrypi: ~' showing the process of downloading and installing hostapd. The user runs 'wget' to download 'adafruit_hostapd.zip' from 'http://www.adafruit.com/downloads/adafruit_hostapd.zip'. The terminal shows the download progress at 100% with a speed of 3.65 MB/s. After downloading, the user runs 'unzip' to extract the files, showing 'inflating: hostapd'. Finally, the user runs 'mv' to move 'hostapd' to '/usr/sbin' and 'chmod 755' to set permissions.

```
pi@raspberrypi ~ $ wget http://www.adafruit.com/downloads/adafruit_hostapd.zip
--2013-06-12 16:06:50--  http://www.adafruit.com/downloads/adafruit_hostapd.zip
Resolving www.adafruit.com (www.adafruit.com)... 207.58.139.247
Connecting to www.adafruit.com (www.adafruit.com)|207.58.139.247|:80... connecte
d.
HTTP request sent, awaiting response... 200 OK
Length: 709582 (693K) [application/zip]
Saving to: `adafruit_hostapd.zip'

100%[=====>] 709,582      3.65M/s   in 0.2s

2013-06-12 16:06:50 (3.65 MB/s) - `adafruit_hostapd.zip' saved [709582/709582]

pi@raspberrypi ~ $ unzip adafruit_hostapd.zip
Archive:  adafruit_hostapd.zip
  inflating: hostapd
pi@raspberrypi ~ $ sudo mv /usr/sbin/hostapd /usr/sbin/hostapd.ORIG
pi@raspberrypi ~ $ sudo mv hostapd /usr/sbin
pi@raspberrypi ~ $ sudo chmod 755 /usr/sbin/hostapd
pi@raspberrypi ~ $
```

First test!

Finally we can test the access point host! Run

```
sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf
```

To manually run **hostapd** with our configuration file. You should see it set up and use **wlan0** then you can check with another wifi computer that you see your SSID show up. If so, you have successfully set up the access point.

```
pi@raspberrypi: ~  
pi@raspberrypi ~ $ sudo mv /usr/sbin/hostapd /usr/sbin/hostapd.ORIG  
pi@raspberrypi ~ $ sudo mv hostapd /usr/sbin  
pi@raspberrypi ~ $ sudo chmod 755 /usr/sbin/hostapd  
pi@raspberrypi ~ $ sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf  
Configuration file: /etc/hostapd/hostapd.conf  
drv->ifindex=3  
l2_sock_recv==l2_sock_xmit=0x0x1fb638  
+rtl871x_sta_deauth_ops, ff:ff:ff:ff:ff:ff is deauth, reason=2  
rtl871x_set_key_ops  
rtl871x_set_key_ops  
rtl871x_set_key_ops  
rtl871x_set_key_ops  
Using interface wlan0 with hwaddr 00:e0:4c:09:3b:f8 and ssid 'Pi_AP'  
rtl871x_set_wps_assoc_resp_ie  
rtl871x_set_wps_beacon_ie  
rtl871x_set_wps_probe_resp_ie  
rtl871x_set_key_ops  
rtl871x_set_beacon_ops  
rtl871x_set_hidden_ssid_ops
```



You can try connecting and disconnecting from the Pi_AP with the password you set before (probably **Raspberry** if you copied our hostapd config), debug text will display on the Pi console but you won't be able to connect through to the Ethernet connection yet.

Cancel the test by typing **Control-C** in the Pi console to get back to the Pi command line

Finishing up!

OK now that we know it works, time to set it up as a 'daemon' - a program that will start when the Pi boots. Run the following commands

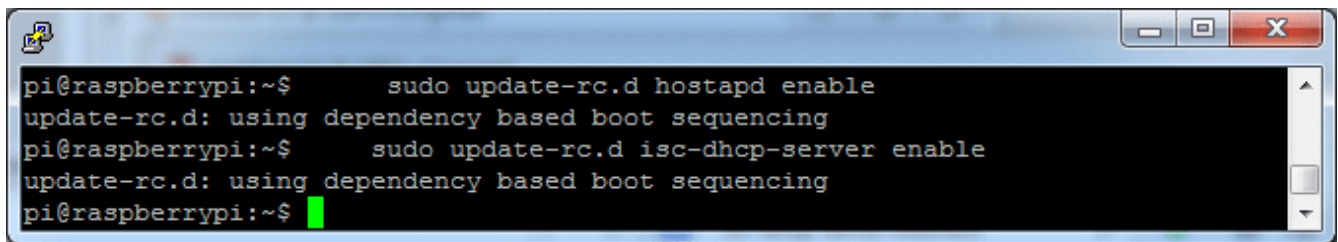
```
sudo service hostapd start
sudo service isc-dhcp-server start
```

you can always check the status of the host AP server and the DHCP server with

```
sudo service hostapd status
sudo service isc-dhcp-server status
```

To start the daemon services. Verify that they both start successfully (no 'failure' or 'errors')
Then to make it so it runs every time on boot

```
sudo update-rc.d hostapd enable
sudo update-rc.d isc-dhcp-server enable
```

A terminal window with a blue title bar and standard window controls. The terminal text shows the user 'pi' at the 'raspberrypi' machine in the '~' directory. They run 'sudo update-rc.d hostapd enable', which outputs 'update-rc.d: using dependency based boot sequencing'. They then run 'sudo update-rc.d isc-dhcp-server enable', which also outputs 'update-rc.d: using dependency based boot sequencing'. The prompt returns to 'pi@raspberrypi:~\$' with a green cursor.

```
pi@raspberrypi:~$ sudo update-rc.d hostapd enable
update-rc.d: using dependency based boot sequencing
pi@raspberrypi:~$ sudo update-rc.d isc-dhcp-server enable
update-rc.d: using dependency based boot sequencing
pi@raspberrypi:~$
```

Extra: Removing WPA-Supplicant

Depending on your distro, you *may* need to remove WPASupplicant. Do so by running this command:

```
sudo mv /usr/share/dbus-1/system-services/fi.epitest.hostap.WPASupplicant.service ~/
```

and then rebooting (**sudo reboot**)