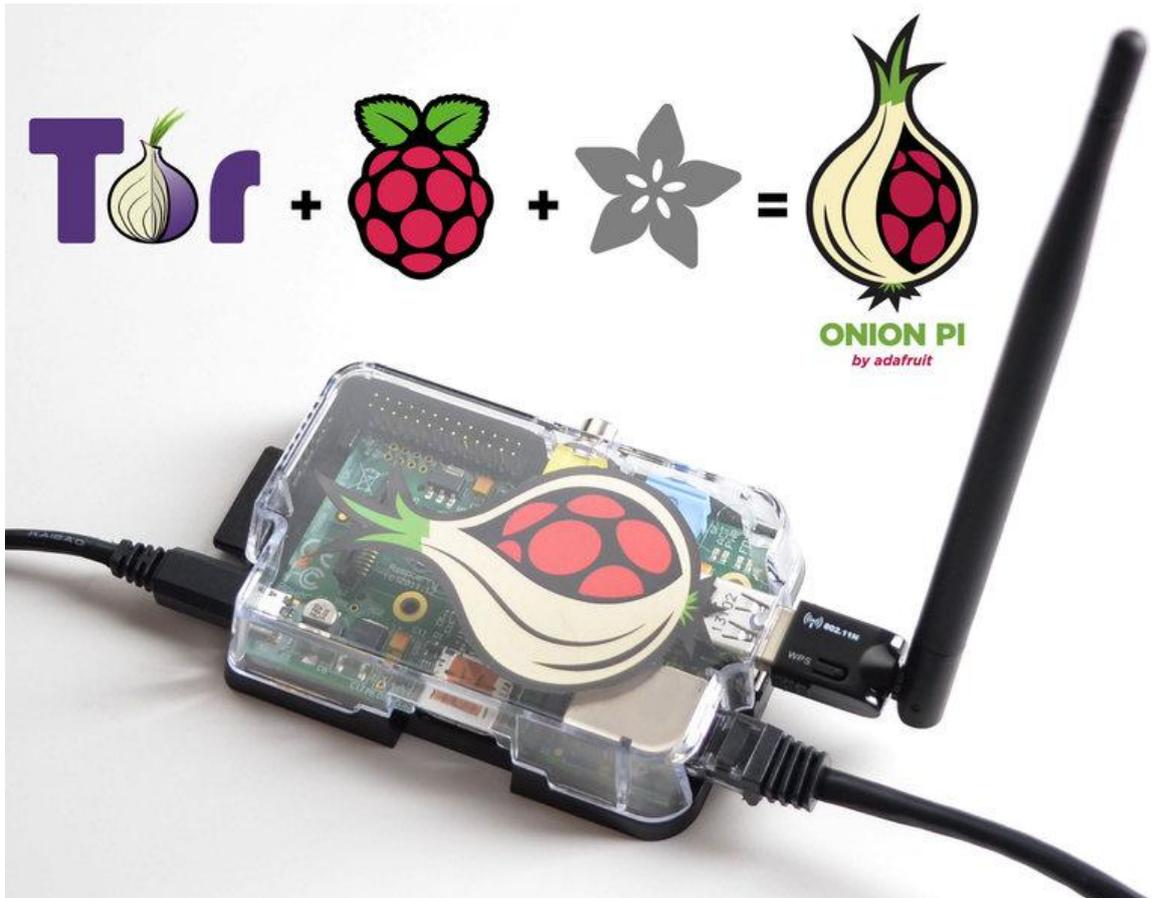




# Onion Pi

Created by lady ada



<https://learn.adafruit.com/onion-pi>

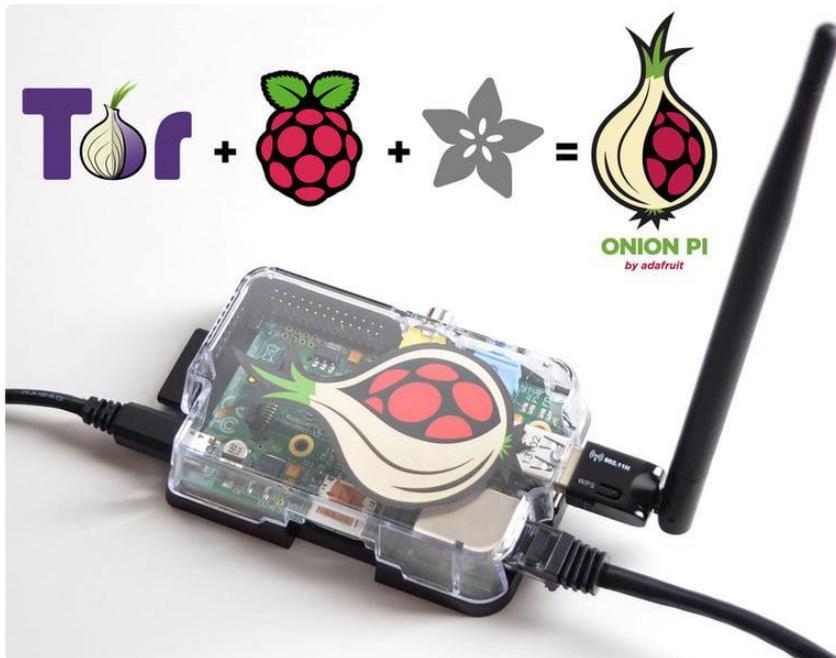
Last updated on 2023-08-29 02:21:03 PM EDT

# Table of Contents

Overview	3
<ul style="list-style-type: none"><li>• Who is this good for?</li><li>• What is Tor?</li></ul>	
What you'll need	5
Preparation	5
Install Tor	8
Test It!	12
Do more...	13
<ul style="list-style-type: none"><li>• Set up as a Wifi-to-Wifi Tor middlebox</li><li>• Configure Tor to make your Exit Node in a particular country only</li><li>• Set up as a Tor Relay or Exit Node</li><li>• Donate to the Tor Project</li></ul>	

---

## Overview



Feel like someone is snooping on you? Browse anonymously anywhere you go with the Onion Pi Tor proxy. This is fun weekend project that uses a Raspberry Pi, a USB WiFi adapter and Ethernet cable to create a small, low-power and portable privacy Pi.

Using it is easy-as-pie. First, plug the Ethernet cable into any Internet provider in your home, work, hotel or conference/event. Next, power up the Pi with the micro USB cable to your laptop or to the wall adapter. The Pi will boot up and create a new secure wireless access point called Onion Pi. Connecting to that access point will automatically route any web browsing from your computer through the anonymizing Tor network.

## Who is this good for?

If you want to browse anonymously on a netbook, tablet, phone, or other mobile or console device that cannot run Tor and does not have an Ethernet connection. If you do not want to or cannot install Tor on your work laptop or loan computer. If you have a guest or friend who wants to use Tor but doesn't have the ability or time to run Tor on their computer, this gift will make the first step much easier.

## What is Tor?

Tor is an onion routing service - every internet packet goes through 3 layers of relays before going to your destination. This makes it much harder for the server you are

accessing (or anyone snooping on your Internet use) to figure out who you are and where you are coming from. It is an excellent way to allow people who are blocked from accessing websites to get around those restrictions.

### [According to the Tor website: \(\)](#)

Journalists use Tor to communicate more safely with whistleblowers and dissidents. Non-governmental organizations (NGOs) use Tor to allow their workers to connect to their home website while they're in a foreign country, without notifying everybody nearby that they're working with that organization.

Groups such as Indymedia recommend Tor for safeguarding their members' online privacy and security. Activist groups like the Electronic Frontier Foundation (EFF) recommend Tor as a mechanism for maintaining civil liberties online. Corporations use Tor as a safe way to conduct competitive analysis, and to protect sensitive procurement patterns from eavesdroppers. They also use it to replace traditional VPNs, which reveal the exact amount and timing of communication. Which locations have employees working late? Which locations have employees consulting job-hunting websites? Which research divisions are communicating with the company's patent lawyers?

A branch of the U.S. Navy uses Tor for open source intelligence gathering, and one of its teams used Tor while deployed in the Middle East recently. Law enforcement uses Tor for visiting or surveilling web sites without leaving government IP addresses in their web logs, and for security during sting operations.

**BEFORE YOU START USING YOUR PROXY - remember that there are a lot of ways to identify you, even if your IP address is 'randomized'. Delete & block your browser cache, history and cookies - some browsers allow "anonymous sessions". Do not log into existing accounts with personally identifying information (unless you're sure that's what you want to do). Use SSL whenever available to end-to-end encrypt your communication. And read <https://www.torproject.org/> for a lot more information on how to use Tor in a smart and safe way**

**This tutorial is a great way to make something fun and useful with your Raspberry Pi, but it is a work in progress. We can't guarantee that it is 100% anonymous and secure! Be smart & paranoid about your Tor usage.**

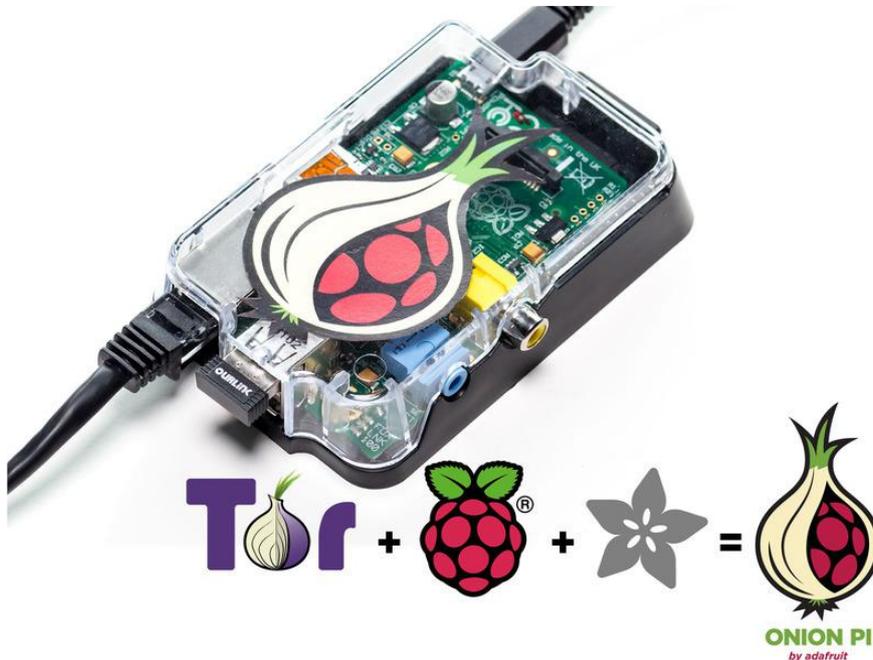
---

# What you'll need

You'll need a few things to run this tutorial:

- [Raspberry Pi model B+](http://adafru.it/1914) (<http://adafru.it/1914>) (or B) - Ethernet is required
- [Ethernet cable](http://adafru.it/730) (<http://adafru.it/730>)
- [WiFi adapter](http://adafru.it/814) (<http://adafru.it/814>) - Not all WiFi adapters work, we know for sure it works with the ones in the Adafruit shop!
- [SD Card \(4GB or greater\)](http://adafru.it/102) (<http://adafru.it/102>) with Raspbian on it. You can either copy the Raspbian image onto it or [buy a ready-made Raspbian card](http://adafru.it/1121) (<http://adafru.it/1121>)
- [Power supply for your Pi](http://adafru.it/1995) (<http://adafru.it/1995>)
- [USB Console cable \(optional\)](http://adafru.it/954) - this makes it a little easier to debug the system (<http://adafru.it/954>)
- [Case for your Pi \(optional\)](http://adafru.it/2258) (<http://adafru.it/2258>)
- [A SD or MicroSD card reader](http://adafru.it/939) (<http://adafru.it/939>) (optional)

Chances are you've got a couple of these items already. If not, our [Onion Pi starter pack](http://adafru.it/1410) (<http://adafru.it/1410>) has everything you need



---

## Preparation

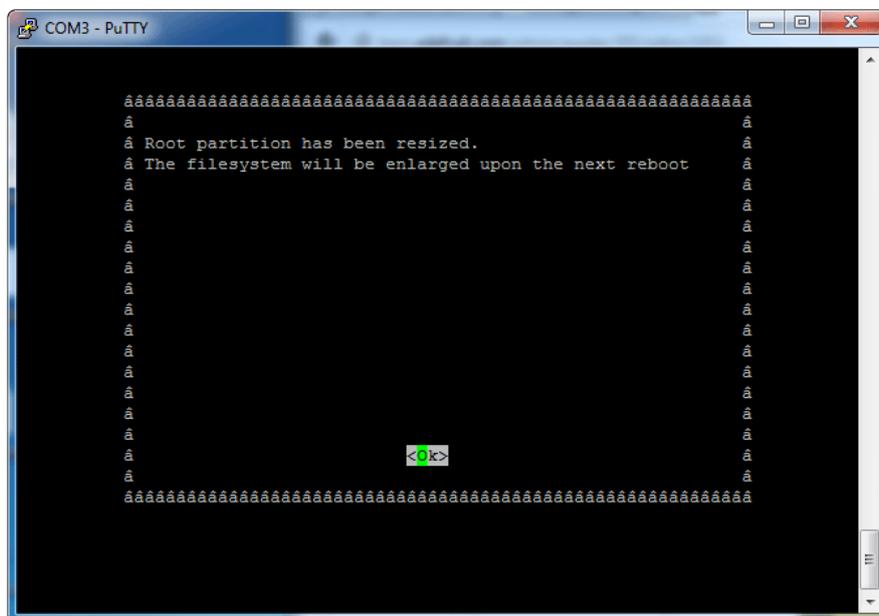
This tutorial assumes you have your Pi mostly set up and have followed our "Raspberry Pi as Wifi Access Point" tutorial

Please follow these tutorials in order to

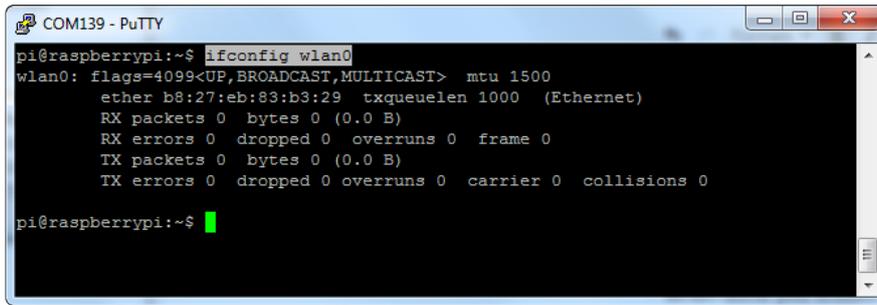
- [Install the OS onto your SD card \(\)](#)  
If you bought an SD card with Raspbian pre-burned on you can skip this step
- [Boot the Pi and log into it \(\)](#)  
Don't forget to change the default password for the 'pi' account!!!

Make sure to expand the filesystem to the entire disk or you may run out of space (this is done by default now on Raspbian when you boot it)

However, don't configure WiFi - you can log in over Ethernet or Serial console



- [Set up and test the Ethernet connection - in general this means just plug into Ethernet before booting \(\)](#)  
Check that you can ssh to, or ping from the Raspberry Pi
- For WiFi, you do not need to configure anything. and that your Wifi adapter is recognized and shows up as wlan0 when you run ifconfig wlan0



```
COM139 - PuTTY
pi@raspberrypi:~$ ifconfig wlan0
wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether b8:27:eb:83:b3:29 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@raspberrypi:~$
```

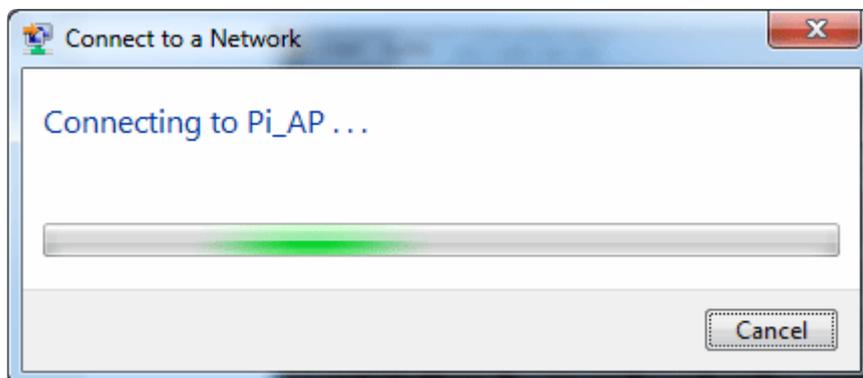
- [Connect with a USB console cable \(optional\) \(\)](#)

Handy for debugging especially when connecting to the access point hosted by the Pi

When done you should have a Pi that is booting Raspbian, has working Ethernet, and you can connect to with a USB console cable and log into the Pi via the command line interface.

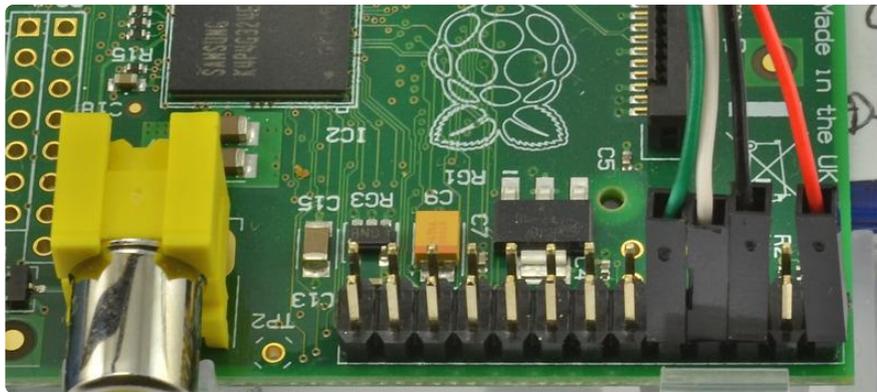
- Then [follow our Pi-as-Access-Point tutorial \(\)](#) to set up the Pi as a wifi access point router.

When done you should be able to connect to the Pi as a WiFi access point and connect to the internet through it.



It is possible to do this tutorial via ssh on the Ethernet port or using a console cable.

If using a console cable, even though the diagram on the last step shows powering the Pi via the USB console cable (red wire) we suggest not connecting the red wire and instead powering from the wall adapter. Keep the black, white and green cables connected as is.



## Install Tor

[Essentially, this tutorial just follows the tor "anonymizing middlebox" writeup here. \(\)](#)

If you hate typing a lot, [this script from break will do it all \(\)](#) for you! Make sure to read through the script to make sure you don't want to change anything! ([More about how to use it here! \(\)](#)) We do suggest going step by step so you can have the experience of all the upkeep tasks.

We'll begin by installing tor - the onion routing software.

Log into your pi by Ethernet or console cable and run

```
sudo apt-get update
sudo apt-get install tor
```

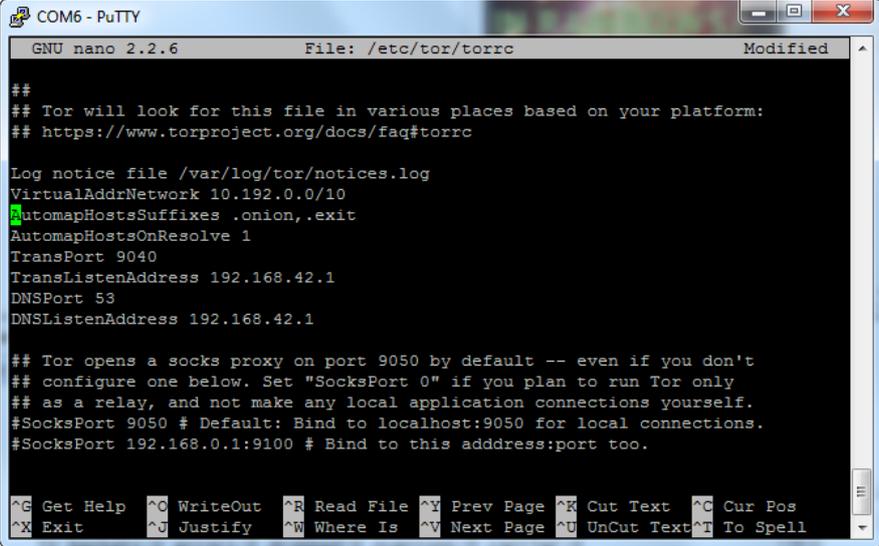
```
COM6 - PuTTY
pi@raspberrypi:~$ sudo apt-get install tor
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  tor-geoipdb torsocks
Suggested packages:
  mixmaster xul-ext-torbutton socat tor-arm polipo privoxy apparmor-utils
The following NEW packages will be installed:
  tor tor-geoipdb torsocks
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 2,694 kB of archives.
After this operation, 7,386 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 tor armhf 0.2.3.25-1 [1,168 kB]
Get:2 http://mirrordirector.raspbian.org/raspbian/ wheezy/main torsocks armhf 1.2-3 [75.0 kB]
Get:3 http://mirrordirector.raspbian.org/raspbian/ wheezy/main tor-geoipdb all 0.2.3.25-1 [1,452 kB]
Fetched 2,694 kB in 7s (344 kB/s)
Selecting previously unselected package tor.
(Reading database ... 62340 files and directories currently installed.)
Unpacking tor (from .../tor_0.2.3.25-1_armhf.deb) ...
```

Edit the tor config file by running

```
sudo nano /etc/tor/torrc
```

and copy and paste the text into the top of the file, right below the the FAQ notice.

```
Log notice file /var/log/tor/notices.log
VirtualAddrNetwork 10.192.0.0/10
AutomapHostsSuffixes .onion,.exit
AutomapHostsOnResolve 1
TransPort 9040
TransListenAddress 192.168.42.1
DNSPort 53
DNSListenAddress 192.168.42.1
```

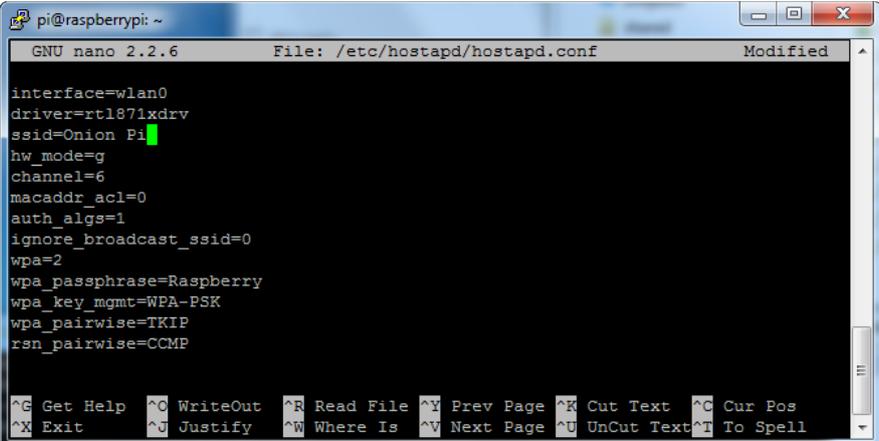


```
COM6 - PuTTY
GNU nano 2.2.6 File: /etc/tor/torrc Modified
##
## Tor will look for this file in various places based on your platform:
## https://www.torproject.org/docs/faq#torrc
Log notice file /var/log/tor/notices.log
VirtualAddrNetwork 10.192.0.0/10
AutomapHostsSuffixes .onion,.exit
AutomapHostsOnResolve 1
TransPort 9040
TransListenAddress 192.168.42.1
DNSPort 53
DNSListenAddress 192.168.42.1
## Tor opens a socks proxy on port 9050 by default -- even if you don't
## configure one below. Set "SocksPort 0" if you plan to run Tor only
## as a relay, and not make any local application connections yourself.
#SocksPort 9050 # Default: Bind to localhost:9050 for local connections.
#SocksPort 192.168.0.1:9100 # Bind to this address:port too.
^G Get Help ^C 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
```

Let's edit the host access point so it is called something memorable like Onion Pi - don't forget to set a good password, don't use the default here!

```
sudo nano /etc/hostapd/hostapd.conf
```

(Don't forget to do the AP setup step in "Preparation" before this!)



```
pi@raspberrypi: ~
GNU nano 2.2.6 File: /etc/hostapd/hostapd.conf Modified
interface=wlan0
driver=rtl871xdrv
ssid=Onion Pi
hw_mode=g
channel=6
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=Raspberr
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
^G Get Help ^C 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
```

Time to change our ip routing tables so that connections via the wifi interface (wlan0) will be routed through the tor software.

Type the following to flush the old rules from the ip NAT table

```
sudo iptables -F
sudo iptables -t nat -F
```

If you want to be able to ssh to your Pi after this, you'll need to add an exception for port 22 like this (not shown in the screenshot below)

```
sudo iptables -t nat -A PREROUTING -i wlan0 -p tcp --dport 22 -j REDIRECT
--to-ports 22
```

Type the following to route all DNS (UDP port 53) from interface wlan0 to internal port 53 (DNSPort in our torrc)

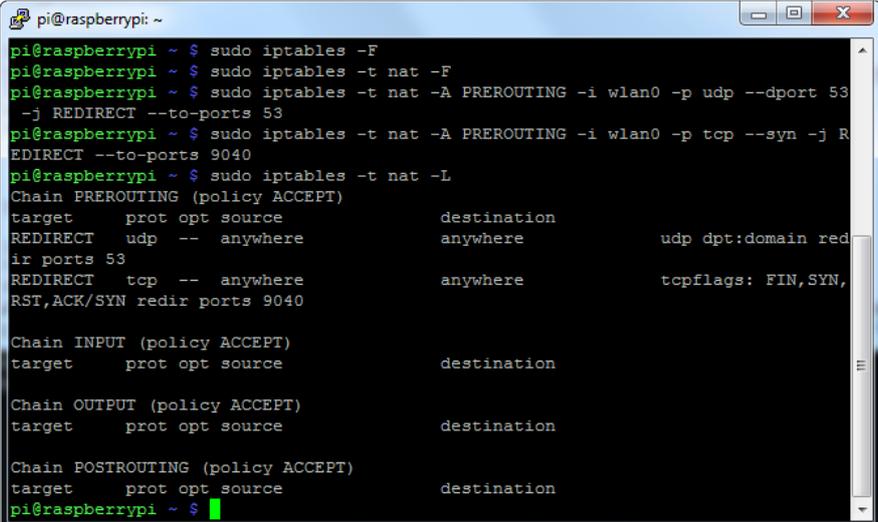
```
sudo iptables -t nat -A PREROUTING -i wlan0 -p udp --dport 53 -j
REDIRECT --to-ports 53
```

Type the following to route all TCP traffic from interface wlan0 to port 9040 (TransPort in our torrc)

```
sudo iptables -t nat -A PREROUTING -i wlan0 -p tcp --syn -j REDIRECT --to-
ports 9040
```

Next you can check that the ip tables are right with

```
sudo iptables -t nat -L
```



```
pi@raspberrypi: ~
pi@raspberrypi ~ $ sudo iptables -F
pi@raspberrypi ~ $ sudo iptables -t nat -F
pi@raspberrypi ~ $ sudo iptables -t nat -A PREROUTING -i wlan0 -p udp --dport 53
-j REDIRECT --to-ports 53
pi@raspberrypi ~ $ sudo iptables -t nat -A PREROUTING -i wlan0 -p tcp --syn -j R
EDIRECT --to-ports 9040
pi@raspberrypi ~ $ sudo iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source                destination
REDIRECT  udp  --  anywhere              anywhere          udp dpt:domain red
ir ports 53
REDIRECT  tcp  --  anywhere              anywhere          tcpflags: FIN,SYN,
RST,ACK/SYN redir ports 9040

Chain INPUT (policy ACCEPT)
target     prot opt source                destination

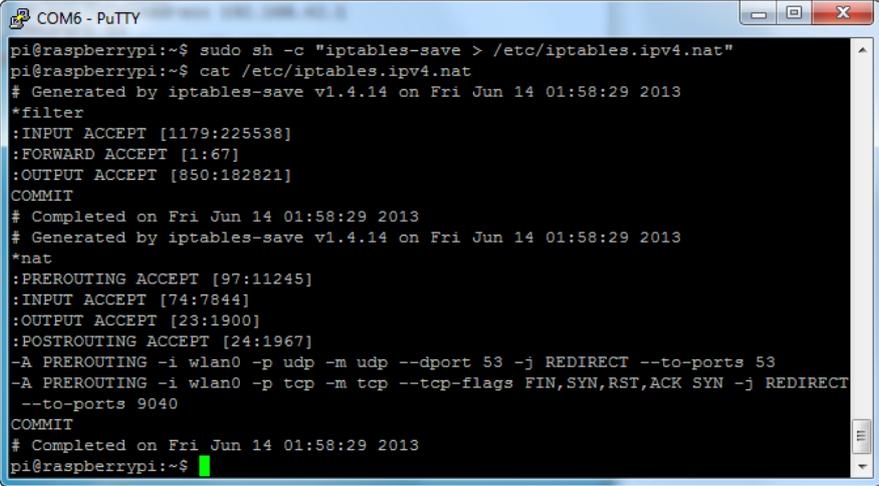
Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source                destination
pi@raspberrypi ~ $
```

If all is good, we'll save it to our old NAT save file

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

It will automatically get loaded when the networking is set up on reboot (as we did in the last tutorial on making a Pi access point)



```
COM6 - PuTTY
pi@raspberrypi:~$ sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
pi@raspberrypi:~$ cat /etc/iptables.ipv4.nat
# Generated by iptables-save v1.4.14 on Fri Jun 14 01:58:29 2013
*filter
:INPUT ACCEPT [1179:225538]
:FORWARD ACCEPT [1:67]
:OUTPUT ACCEPT [850:182821]
COMMIT
# Completed on Fri Jun 14 01:58:29 2013
# Generated by iptables-save v1.4.14 on Fri Jun 14 01:58:29 2013
*nat
:PREROUTING ACCEPT [97:11245]
:INPUT ACCEPT [74:7844]
:OUTPUT ACCEPT [23:1900]
:POSTROUTING ACCEPT [24:1967]
-A PREROUTING -i wlan0 -p udp -m udp --dport 53 -j REDIRECT --to-ports 53
-A PREROUTING -i wlan0 -p tcp -m tcp --tcp-flags FIN,SYN,RST,ACK SYN -j REDIRECT
--to-ports 9040
COMMIT
# Completed on Fri Jun 14 01:58:29 2013
pi@raspberrypi:~$
```

Next we'll create our log file (handy for debugging) with

```
sudo touch /var/log/tor/notices.log
sudo chown debian-tor /var/log/tor/notices.log
sudo chmod 644 /var/log/tor/notices.log
```

Check it with

```
ls -l /var/log/tor
```

Start the tor service manually

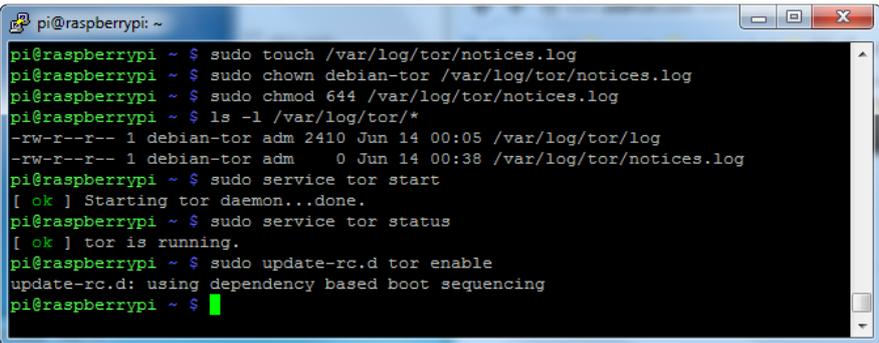
```
sudo service tor start
```

Check its really running (you can run this whenever you're not sure, if something is wrong you'll see a big FAIL notice)

```
sudo service tor status
```

Finally, make it start on boot

```
sudo update-rc.d tor enable
```



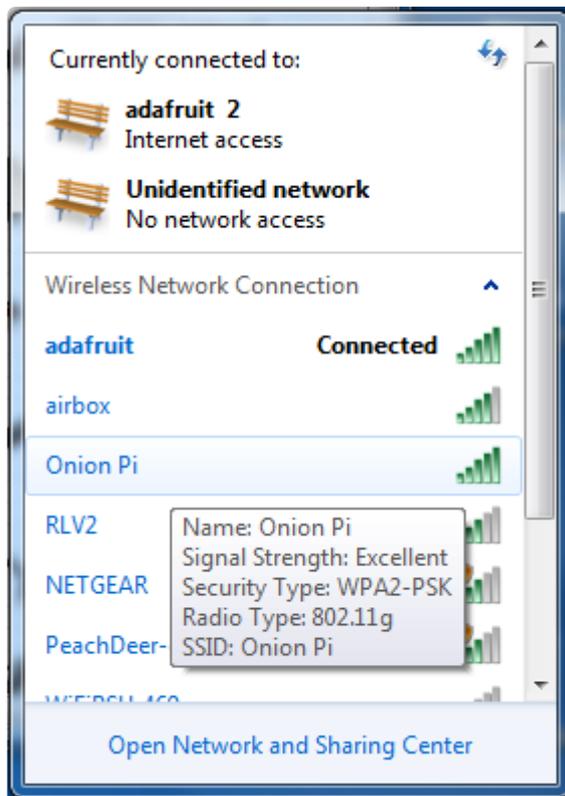
```
pi@raspberrypi: ~
pi@raspberrypi ~ $ sudo touch /var/log/tor/notices.log
pi@raspberrypi ~ $ sudo chown debian-tor /var/log/tor/notices.log
pi@raspberrypi ~ $ sudo chmod 644 /var/log/tor/notices.log
pi@raspberrypi ~ $ ls -l /var/log/tor/*
-rw-r--r-- 1 debian-tor adm 2410 Jun 14 00:05 /var/log/tor/log
-rw-r--r-- 1 debian-tor adm    0 Jun 14 00:38 /var/log/tor/notices.log
pi@raspberrypi ~ $ sudo service tor start
[ ok ] Starting tor daemon...done.
pi@raspberrypi ~ $ sudo service tor status
[ ok ] tor is running.
pi@raspberrypi ~ $ sudo update-rc.d tor enable
update-rc.d: using dependency based boot sequencing
pi@raspberrypi ~ $
```

That's it, now you're ready to test in the next step.

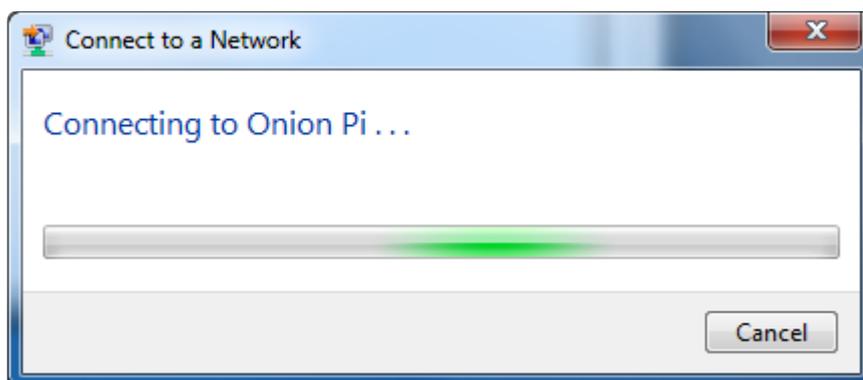
---

## Test It!

OK now the fun part! It's time to test your TOR anonymizing proxy. On a computer, check out the available wifi networks, you should see the Onion Pi network



Connect to it using the password you entered into the hostapd configuration file



You can open up a Terminal or command prompt and ping 192.168.42.1 to check that your connection to the Pi is working. However you won't be able to ping outside of it because ping's are not translated through the proxy

```
C:\Windows\system32\cmd.exe
^C
C:\Users\ladyada>ping 192.168.42.1
Pinging 192.168.42.1 with 32 bytes of data:
Reply from 192.168.42.1: bytes=32 time=29ms TTL=64
Reply from 192.168.42.1: bytes=32 time=49ms TTL=64
Reply from 192.168.42.1: bytes=32 time=11ms TTL=64
Reply from 192.168.42.1: bytes=32 time=317ms TTL=64
Ping statistics for 192.168.42.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 11ms, Maximum = 317ms, Average = 101ms
C:\Users\ladyada>
```

To check that the proxy is working, visit a website like <http://www.ipchicken.com> () which will display your IP address as it sees it and also the matching domain name if available. The IP address should not be from your internet provider - in fact, if you reload the page it should change!



Your web browsing traffic is now anonymized!

BEFORE YOU START USING YOUR PROXY - remember that there are a lot of ways to identify you, even if your IP address is 'randomized'. Delete your browser cache, history and cookies (some browsers allow "anonymous sessions") and read <https://www.torproject.org/> for a lot more information on how to use TOR in a smart and safe way

## Do more...

Now that you have this project set up, you can do more...

## Set up as a Wifi-to-Wifi Tor middlebox

We use Ethernet because it requires no configuration or passwords, just click the cable to get DHCP but if you want, its possible to set it up as a WiFi-to-WiFi proxy. You'll need two WiFi adapter, then edit /etc/networks/interfaces to add wlan1 and [enter in the SSID/password for your Internet provider using our WiFi tutorial \(\)](#) We don't have a tutorial for this project

## Configure Tor to make your Exit Node in a particular country only

Its very easy to configure tor to give you a presence in any country of your choice. For example here's my torrc that makes me 'present' in Great Britain.

Replace aaa.bbb.ccc.ddd by the IP address of your RPi and GB by the country code of your choice.

Configure your browser to uses a Socks 5 proxy on aaa.bbb.ccc.ddd, port 9050

```
Log notice file /var/log/tor/notices.log
SocksListenAddress aaa.bbb.ccc.ddd
ExitNodes {GB}
StrictNodes 1
```

## Set up as a Tor Relay or Exit Node

If you like using Tor, help make it faster by joining as a relay, or increasing the anonymity by becoming an exit node. [Check out the Tor project website for how to edit your \(\)torrc \(\) to turn your Pi into either \(\)](#).

## Donate to the Tor Project

If you like using Tor, but can't run a relay or exit node - [consider donating to the project which helps pay for developers, servers and more. \(\)](#) Your donation is tax-deductable if you are in the US