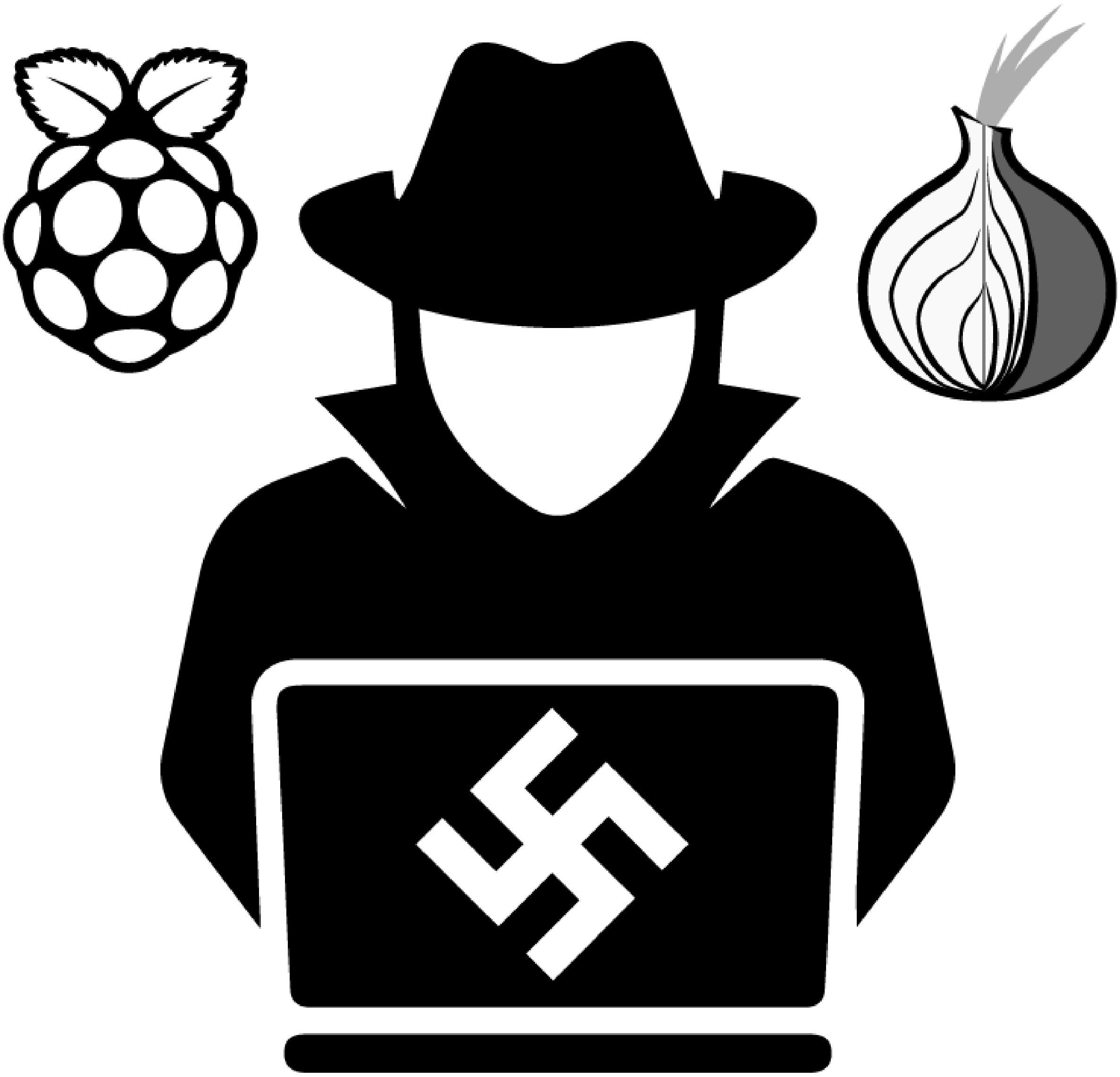
BUILD IT: TOR PI

A STEP-BY-STEP GUIDE



BY: OPSECGOY

DATE: APRIL, 2020

TABLE OF CONTENTS

WHAT & WHY

WHAT IN THE HELL IS A TOR PI?

Welcome to my step-by-step guide on building a Tor Pi. You may be wondering what in the hell a Tor Pi is and why you would need one. Let me explain… A Tor Pi, for lack of a better term, is a Raspberry Pi that acts like a wireless access point. When you connect your **[redacted]** devices to this wireless access point it will route all traffic over the Tor network. This can come in handy for a variety of reasons, but in particular, if you want to connect your **[redacted]** phone to the internet, but you are not particularly fond of having that device showing up on your normie router or associated in any way with your “normie” internet traffic.

WHAT YOU WILL NEED

BE PREPARED FOR THE BUILD

Building a Tor Pi does not require a lot to get started. Here is a list of items you will need to successfully complete this project.

1. Raspberry Pi 3 (Model B+)
2. Micro USB Power Supply (5V DC @ 2.5A+)
3. microSD card (at least 8GB)
4. Raspberry Pi Case ****Optional****
5. Ethernet cable
6. Laptop/Desktop
7. microSD Adapter/Slot in Laptop/Computer

INITIALIZE PI

SETTING IT UP

In this section I will show you how to setup your Raspberry Pi for the first time. If you already have a Raspberry Pi setup and ready to go you can safely skip this section.

Alright, let’s get started! The first step is to gather all of the files we will need to complete the project.

DOWNLOAD RASPBIAN

To get Raspbian, the official operating system of the Raspberry Pi, check out the Raspbian download page. As of the time of writing the latest release is called “Raspbian Buster”. When the page loads you should see three different options, “Raspbian w/ Desktop” is the one I would choose. Raspian Lite is nice because it is very small and comes with almost nothing.

It also can be a pain in the ass for the exact same reason. The decision is ultimately up to you if you feel like messing around with it and you are comfortable enough in the terminal.

DOWNLOAD ETCHER

Once you have downloaded Raspbian it is time to download and install Etcher. This program will take your image (Raspbian) and burn it to your microSD card. You can get Etcher here for just about any OS.

PREPARE THE MICRO SD

Now it is time to insert the microSD card into the drive so you can burn the image to the drive. So pop the card in and fire up Etcher.

BURN THE IMAGE TO THE DRIVE

Now it is time to insert the microSD card into the drive so you can burn the image to the drive. So pop the card in and fire up Etcher.Select the image you downloaded (Raspbian) and the drive you anttoput it on (microSD) and Burn it on there.

ENABLE SSH

If Etcher automatically unmounted your microSD make sure you re-mount it and then find the path to it. From there create an empty file in the root of the microSD called **ssh**. To do this in Linux open a terminal and run the command, `touch /path/to/microSD/ssh` (replace with the proper path).

SETUP THE PI

Now that you have the Raspbian image burnt onto the microSD safely eject the microSD and put it in the microSD port on the bottom of your Raspberry Pi. Connect the Pi to your network via Ethernet cable and you are ready to power it up!

POWER THE PI

Get your 5VDC 2.5+A power supply for the Pi and plug it in. Now your Pi should boot up. Give it 90 seconds before you proceed.

CONNECT TO THE PI

You will need to find out what your Pi’s ip address is within your network. Logging in to your router can usually give you the quickest and easiest result. Once you have the IP for your Pi you connect via SSH through the terminal like so…

Assuming the Pi’s IP is 192.168.1.10 it would look like this:

```
ssh pi@192.168.1.10
```

You will then be prompted for a password which is ‘**raspberry**’ by default.

BEGIN THE BUILD

SETTING UP WIRELESS ACCESS POINT

Now that we have completed the initial setup we are ready to begin the exciting part, turning the Pi into a Tor Pi. Let’s begin.

First, connect to the Pi via SSH again as before. We need to make sure everything is up-to-date. To do that enter these commands:

```
sudo apt upgrade -y
sudo apt update -y
```

Once everything is updated we need to install two new programs.

```
sudo apt install hostapd dnsmasq -y
```

We need to immediately stop these packages from running so that we can configure them.

Here is the command for that…

```
sudo systemctl stop hostapd
sudo systemctl stop dnsmasq
```

Now that hostapd and dnsmasq is stopped we need to modify our dhcpd configuration so that we can control the wlan0 interface.

```
sudo nano /etc/dhcpcd.conf
```

Now at the bottom of this file we will set up the wlan0 interface by adding these lines…

```
interface wlan0

    static ip_address=192.168.220.1/24
    nohook wpa_supplicant
```

Now close the file by doing CTRL+x then y and finally [Enter]

Time to restart the dhcpd service so it can load in our new configuration.

```
sudo systemctl restart dhcpcd
```

Moving on to the hostapd configuration. Run this in the terminal…

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

Paste this in your file as follows and adjust for the **ssid** and the **wpa_passphrase**. The **ssid** is what you want to name the network you are setting up and the passphrase… well that should be obvious.

```
interface=wlan0
driver=nl80211
hw_mode=g
channel=6
ieee80211n=1
wmm_enabled=0
macaddr_acl=0
ignore_broadcast_ssid=0
auth_algs=1
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
# This is the name of the network
ssid=torpi
# The network passphrase
wpa_passphrase=passwordhere
```

Again, exit the file with **Ctrl+X** then **Y** then **[Enter]**.

Now it is time to let the machine know where our configuration file is located. To do that start by running this command…

```
sudo nano /etc/default/hostapd
```

Now look for this line

```
#DAEMON_CONF=""
```

And replace it with this

```
#DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

Next we need to edit another file.

```
sudo nano /etc/init.d/hostapd
```

Again, find this line

```
DAEMON_CONF=
```

and replace with this

```
DAEMON_CONF=/etc/hostapd/hostapd.conf
```

SETUP DNSMASQ

Now that we have completed setting up hostapd it is time to set up dnsmasq. Start by saving a backup of the original configuration file.

```
sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
```

Now we make our own configuration file.

```
sudo nano /etc/dnsmasq.conf
```

Here is what the configuration should look like.

```
# Use interface wlan0
interface=wlan0
# Use Cloudflare DNS
server=1.1.1.1
# IP range and lease time
dhcp-range=192.168.220.50,192.168.220.150,12h
```

Once complete save and close.

Note: This configuration uses CloudFlare’s DNS however, you may wish to use a different DNS provider from the list below…

Name	**IPV4 Address**	**IPV6 Address**
Alternate DNS	198.101.242.72 23.253.163.53	
BlockAid Public DNS (or PeerDNS)	205.204.88.60 178.21.23.150	
Censurfridns	91.239.100.100 89.233.43.71	2001:67c:28a4:: 2002:d596:2a92:1:71:53::
Christoph Hochstätter	209.59.210.167 85.214.117.11	
ClaraNet	212.82.225.7 212.82.226.212	
Comodo Secure DNS	8.26.56.26 8.20.247.20	
DNS.Watch	84.200.69.80 84.200.70.40	2001:1608:10:25::1c04:b12f 2001:1608:10:25::9249:d69b

DNSReactor	104.236.210.29 45.55.155.25	
Dyn	216.146.35.35 216.146.36.36	
FDN	80.67.169.12	2001:910:800::12
FoeBud	85.214.73.63	
FoolDNS	87.118.111.215 213.187.11.62	
FreeDNS	37.235.1.174 37.235.1.177	
Freenom World	80.80.80.80 80.80.81.81	
German Privacy Foundation e.V.	87.118.100.175 94.75.228.29 85.25.251.254 62.141.58.13	
GreenTeamDNS	81.218.119.11 209.88.198.133	
Hurricane Electric	74.82.42.42	2001:470:20::2
Level3	209.244.0.3 209.244.0.4	
Neustar DNS Advantage	156.154.70.1 156.154.71.1	
New Nations	5.45.96.220 185.82.22.133	
Norton DNS	198.153.192.1 198.153.194.1	
OpenDNS	208.67.222.222 208.67.220.220	2620:0:ccc::2 2620:0:ccd::2
OpenNIC	58.6.115.42 58.6.115.43 119.31.230.42 200.252.98.162 217.79.186.148 81.89.98.6 78.159.101.37 203.167.220.153 82.229.244.191 216.87.84.211 66.244.95.20 207.192.69.155	2001:470:8388:2:20e:2eff:fe63:d4a9 2001:470:1f07:38b::1 2001:470:1f10:c6::2001

	72.14.189.120	
PowerNS	194.145.226.26 77.220.232.44	
Quad9	9.9.9.9	2620:fe::fe
SafeDNS	195.46.39.39 195.46.39.40	
SkyDNS	193.58.251.251	
SmartViper Public DNS	208.76.50.50 208.76.51.51	
ValiDOM	78.46.89.147 88.198.75.145	
Verisign	64.6.64.6 64.6.65.6	2620:74:1b::1:1 2620:74:1c::2:2
Xiala.net	77.109.148.136 77.109.148.137	2001:1620:2078:136:: 2001:1620:2078:137::
Yandex.DNS	77.88.8.88 77.88.8.2	2a02:6b8::feed:bad 2a02:6b8:0:1::feed:bad
puntCAT	109.69.8.51	2a00:1508:0:4::9

Now we need to tell the Pi to forward all the traffic from the wlan0 connection through the ethernet connection.

```
sudo nano /etc/sysctl.conf
```

Find this commented line

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

and uncomment the line like this…

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

Activate it now so you do not have to wait for reboot.

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

Now that Ipv4 forwarding is enabled we can setup a NAT between wlan0 and eth0.

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j 
MASQUERADE
```

Every time the Pi is rebooted the iptables rules will be flushed so we have to store the rules somewhere so that they can be loaded back in on every boot.

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

Time to make the rules re-apply after boot each time. The most simple way to do this is by editing rc.local file.

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

Now that we are editing the rc.local file we need to make sure we enter our command above the line that says `exit 0` Anything below that line will not be executed. Once you find that line add this above it.

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

Save and close the file.

Now all we have left to do before we have a functional wireless access point is to start and enable the services we just configured.

```
sudo systemctl unmask hostapd
sudo systemctl enable hostapd
sudo systemctl start hostapd
sudo service dnsmasq start
```

Check and see if you can access this wireless access point. If it is appearing and you can connect and browse the web with it then congratulations you now have a functioning wireless access point! But that is not the point of this guide. The point is to route all of our traffic over the Tor network via this wireless access point. To do that we will need to do a few additional steps.

ADDING TOR

WIRELESS ACCESS POINT + TOR

We have come a long way. Now that we have a working wireless access point it’s time to add Tor to complete the build. First we will need to install Tor.

```
sudo apt install tor -y
```

Once Tor is installed it is time to edit the configuration file.

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

Under the top commented out block that ends with…

```
## Tor will look for this file in various places based on your platform:
## https://www.torproject.org/docs/faq#torrc
```

Add these lines:

```
Log notice file /var/log/tor/notices.log
VirtualAddrNetwork 10.192.0.0/10
AutomapHostsSuffixes .onion,.exit
AutomapHostsOnResolve 1
TransPort 9040
TransListenAddress 192.168.220.1
DNSPort 53
DNSListenAddress 192.168.220.1
```

Save and exit the torrc file.

Flush the iptables and add new rules.

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

Now update iptables with these rules:

```
sudo iptables -t nat -A PREROUTING -i wlan0 -p tcp --dport 22 -j REDIRECT --to-ports 22
sudo iptables -t nat -A PREROUTING -i wlan0 -p udp --dport 53 -j REDIRECT --to-ports 53
sudo iptables -t nat -A PREROUTING -i wlan0 -p tcp --syn -j REDIRECT --to-ports 9040
```

Just as before when we saved the iptables rules due to flushing at every boot, we will need to do this again.

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

As you may have noticed in the torrc file we gave Tor a place to store logs. That file does not yet exist so now we will create that file and set the proper permissions.

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

Time to start the Tor service!

```
sudo service tor start
```

Make sure the service is running without issue.

```
sudo service tor status
```

We need to make sure Tor starts on boot every time.

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

Finally give the Pi a reboot to make sure everything is going to run smoothly! Remember to give your new TorPi 90 seconds before attempting to SSH or connect to the wireless access point. If all is working, Congratulations!

THANK YOU

Thank you for reading this guide! If you found this **free** guide particularly helpful, please consider donating a few shekels our way!

Pro-tip: Before sending any donation to the address listed, make sure you **double-check that the PGP signature** included in the .zip with the document is **valid.**

You can always email me first at OpSecGoy@protonmail.com to confirm.

Thanks again for your help,
“From war to victory!”
-OpSecGoy

卐卐卐

Privacy & Security Goy Hidden Service:

goysec74znsyewq3nu2i3kmwozxptc3lx22jg67km6r2we37ejiaz5yd.onion

Telegram Channel: @PrivSecGoys

Telegram Chat: @PrivSecChat

Monero

44afqsvYK6qeqPsNftcqWeJNSqYjP8jXtFcX2A8AQDKzCR1pusDSUehXNJBCqjmf4oVwd7VsRr2NZVMdEEe6i78ESzYcXWp