Tor-Proxy einrichten - Praktisches Handbuch

Dimitri Graf und Timo Furrer

April 2014

Inhaltsverzeichnis

1 Einleitung **2**

2 Raspberry Pi **2**
2.1 Hardware 2
2.2 Betriebssystem 3

3 Washalb ein Raspberry Pi **4**

4 Was man braucht **4**
4.1 Betriebssystem installieren 5

5 Vorkonfiguraton **5**
5.1 Betriebssystem starten 5
5.2 Root-Rechte erlangen 7
5.3 System aktualisieren 7

6 WiFi-Adapter testen **8**

7 Access Point aufsetzen **8**
7.1 DHCP-Server konfigurieren 9
7.2 Hostapd konfigurieren 11
7.3 NAT konfigurieren 11

8 Erster Start des Access Points **12**
8.1 Fehlerbehandlung 13
8.2 Verbindung herstellen zum Access Point 14

9 Daemons einrichten **14**

10 Tor aufsetzen **15**
10.1 Protokollierung aktivieren 16

11 IP-Adresse verifizieren **17**

12 Exit-Nodes konfigurieren **18**

13 Abschluss **19**

1 Einleitung

Dieses Handbuch zeigt Schritt für Schritt auf, wie ein Raspberry Pi mit ein wenig Zubehör in einen Tor-Proxy umgewandelt werden kann. Mittels Tor-Proxy kann sich der Anwender schnell und einfach Anonymität verschaffen und so sicherer im Internet unterwegs sein. Die Anonymität wird dabei durch das Tor-Netzwerk zur Verfügung gestellt, in das sich der Anwender über den Proxy einklinkt. Dieses Handbuch macht vor allem dann Sinn, wenn das Prinzip und die Funktionsweise von Tor bekannt und verstanden sind. Folgende Seiten sind eine erste Anlaufstelle für zusätzliche Informationen:

- Homepage des Torprojektes, Englisch: `https://www.torproject.org/`
- Wikipedia-Artikel, Deutsch: `http://de.wikipedia.org/wiki/Tor_%28Netzwerk%29`

2 Raspberry Pi

Ein Raspberry Pi ist ein Einplatinencomputer in der Grösse einer Kreditkarte. Er wurde 2009 von der Raspberry Pi Foundation entwickelt, um Schulen einen kostengünstigen Computer zur Verfügung zu stellen. Dabei ging es vor allem um das Erlernen von Computerwissenschaften, wie Programmierung von Software oder Ansteuerung einfacher Hardware, wie LEDs, Displays und einfachen Motoren. [1] [2] Der Raspberry Pi ist sehr vielseitig einsetzbar und nicht zuletzt deswegen hat sich eine grosse und begeisterte Community gebildet. Als Betriebssystem kommt auf Grund der leistungsfähig begrenzten Hardware meistens eine Linux-basierte Distribution zum Einsatz. Linux arbeitet sehr ressourcenschonend und die Distributionen sind zudem oft speziell auf den Rechner und das jeweilige Einsatzgebiet angepasst. Der Raspberry Pi ist ein Spielzeug für technikbegeisterte Menschen - jung und alt gleichermassen. Die Community ist ein essentieller Bestandteil des Projektes, denn sie erfinden ständig neue Einsatzmöglichkeiten, kreieren interessante Produkte und Projekte und stellen Anleitungen ins Netz. Um dann selbst loszulegen, braucht man oft nicht mehr als den Raspberry Pi selbst, Monitor, Tastatur und Maus und ein wenig Know-How.

2.1 Hardware

Es ist wichtig, für die Inbetriebnahme des Raspberry Pis ein wenig über die Hardware Bescheid zu wissen. Es gibt zur Zeit zwei Modelle des Mini-Rechners: das ursprüngliche Model A und das erweiterte Model B. Model B ist performanter und verfügt über mehr Anschlüsse. Aus diesem Grund ist es Model A in den meisten Fällen vorzuziehen.

Raspberry Pi, Model B verfügt über folgende Anschlüsse:

- Kartenleser für SD/MMC/SDIO (Hauptspeicher, Betriebssystem)
- 2x USB 2.0
- FBAS (Videoausgabe)

[1] `http://www.raspberrypi.org/faqs#introWhatIs`
[2] `http://www.raspberrypi.org/about`

- HDMI (Video & Audio)
- Klinkenstecker, 3.5mm (Audio)
- 10/100-MBit Ethernet Controller (RJ45, Netzwerk)
- 17 GPIO Pins (Externe Hardware)
- 5V Micro-USB (Stromanschluss) [3]

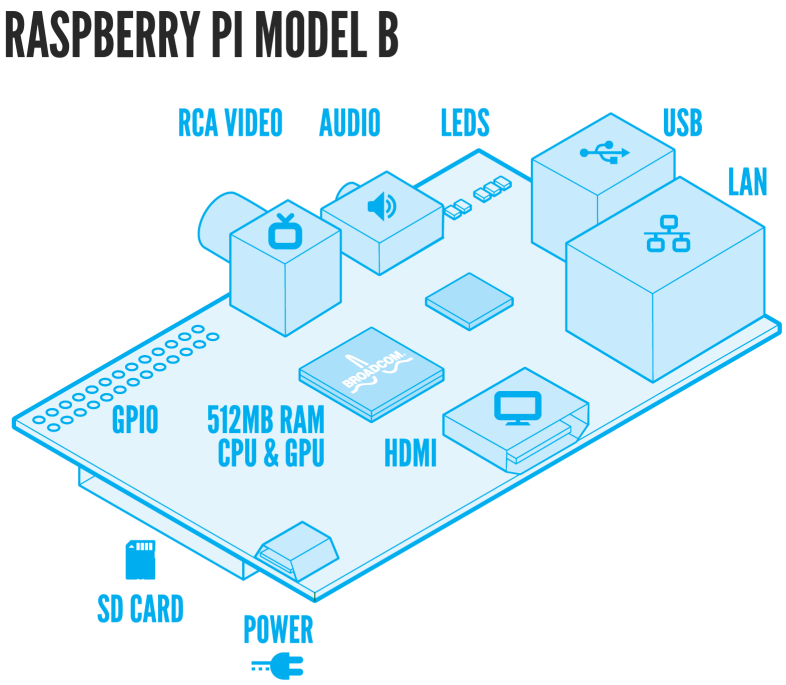

Abbildung 1: Raspberry Pi, Model B

2.2 Betriebssystem

Wie bereits erwähnt, wird in diesem Versuch eine Linux-basierte Distribution eingesetzt, die speziell auf den Raspberry Pi zugeschneidert wurde. Raspbian basiert auf der Linux-Distribution Debian und ist neben den Grundfunktionalitäten mit vielen weiteren Programmen ausgestattet, die *out of the box* genutzt werden können. [4] Wie die meisten anderen Linux-basierten Distributionen, kann Raspbian gratis aus dem Internet heruntergeladen und gebraucht werden. Natürlich ist es nicht das einzige Linux-Betriebssystem, welches auf dem Raspberry Pi eingesetzt werden kann. Eine Liste bekannter Distributionen kann unter `http://www.raspberrypi.org/downloads` gefunden werden.

[3] `http://de.wikipedia.org/wiki/Raspberry_Pi#Spezifikationen`
[4] `http://www.raspbian.org/`

3 Washalb ein Raspberry Pi

Die Frage die sich vielleicht stellt ist, wieso gerade der Raspberry Pi gut als Tor-Proxy geeignet ist. Dies hat zum einen den Grund, dass die Hardware an sich für wenig Geld erhältlich ist und trotzdem die benötigte Leistung bringt. Ein anderer, fast noch wichtigerer Punkt ist der, dass der Raspberry Pi durch seine geringe Grösse sehr mobil ist. Man kann ihn überall hin mitnehmen, ohne dass er gross auffällt oder Schwierigkeiten bereitet. Ist man beispielsweise auf Geschäftsreise und möchte nicht, dass das Hotel mitkriegt, wo man im Internet unterwegs ist, kann man kurzerhand den Raspberry Pi (Tor-Proxy) in Betrieb nehmen und *anonym* lossurfen.

4 Was man braucht

Damit ein Raspberry Pi in einen Tor-Proxy umgewandelt werden kann, benötigt man folgendes:

- Raspberry Pi, Model B
- Micro-USB Kabel für Stromversorgung
- SD Speicherkarte
- RJ45 Netzwerkkabel
- WiFi-Adapter

Es gibt gewisse Voraussetzungen, die man bei der Auswahl der Komponenten beachten muss. Das Micro-USB Kabel für die Stromversorgung muss folgende Spezifikationen erfüllen: mind. 700mA, 5V. Die SD-Karte, auf die später das Betriebssystem geladen wird, sollte mindestens eine Kapazität von 4 Gigabyte aufweisen. Zudem muss man beachten, dass nicht jede verfügbare Karte auch tatsächlich unter dem verwendeten Betriebssystem funktioniert. Deshalb gibt es im Internet eine Liste, auf der eine Reihe kompatibler und nicht kompatibler Karten aufgeführt werden: `http://elinux.org/RPi_SD_cards#Working_.2F_Non-working_SD_cards`. Das Netzwerkkabel wird verwendet, um den Raspberry Pi mit dem Internet zu verbinden. Dies kann beispielsweise über einen Switch, Router oder RJ45-Dose geschehen.
Bei der Wahl des WiFi–Adapters ist besonders Vorsicht geboten. Erstens sind nicht alle im Handel erhältlichen Adapter sind mit dem Raspberry Pi kompatibel. Eine Liste von kompatiblen Adaptern gibt es hier: `http://elinux.org/RPi_USB_Wi-Fi_Adapters`. Zweitens können nicht alle kompatiblen WiFi-Adapter in den Access-Point-Modus versetzt werden. Das für dieses Tutorial eingesetzte Programm hostapd, welches die Access Point-Funktionalitäten zur Verfügung stellt, unterstützt nur gewisse WiFi-Adapter bzw. gewisse Treiber. man sollte sich also gut informieren bevor man sich für einen Adapter entscheidet. Für dieses Tutorial wurde ein WiFi-Adapter von B-Link eingesetzt. Er wird von hostapd unterstützt und kann über adafruit oder Pi-Shop für wenig Geld erworben werden.

Für die Einrichtung selbst wird noch zusätzliche Peripherie gebraucht, die aber in den meisten Haushalten schon vorhanden sein sollte:

- Computer oder Notebook
- SD-Karten-Leser
- Monitor mit HDMI-Anschluss
- HDMI-Kabel
- Tastatur
- Maus

Die aufgelisteten Peripherie braucht man nur für die Einrichtung selbst oder anfallende Wartungsarbeiten. Ist alles erst einmal korrekt eingerichtet, kann diese wieder andersweitig verwendet werden.
Detaillierte Informationen und Tipps zu den Komponenten sowie der Peripherie findet man unter: `http://www.raspberrypi.org/phpBB3/viewtopic.php?t=4277`

4.1 Betriebssystem installieren

Bevor der Computer gestartet und mit der Einrichtung begonnen werden kann, muss noch das Betriebssystem installiert werden. Die SD Speicherkarte kann mit wenigen Anweisungen von jedem gängigen Betriebssystem (Windows, OSX, Linux) aus mit einem Raspbian bestückt werden.

Zuerst gilt es das Betriebssystem selbst aus dem Internet zu laden. [5]. Ist die Datei vollständig heruntergeladen und die SD-Karte angeschlossen, kann das heruntergeladene Image (Datei) auf die Karte geladen werden. Es gibt eine sehr gute Anleitung im Internet, die die einzelnen Schritte in Details beschreibt. Man findet sie unter: `http://elinux.org/RPi_Easy_SD_Card_Setup`.

5 Vorkonfiguraton

Nach erfolgreichem Aufspielen des Betriebssystem auf die SD-Karte, folgt nun dessen Vorkonfiguration. Es müssen gewisse Einstellungen vorgenommen und Komponenten aktualisiert werden, um eine gute Basis für die Einrichtung des Tor-Proxy zu legen.

5.1 Betriebssystem starten

Beim ersten Start des Raspberry öffnet sich nach kurzer Wartezeit ein graues Fenster auf blauem Hintergrund mit dem Titel *Raspberry Pi Software Configuration Tool*.

Dieses Werkzeug dient dazu, gewisse Grundeinstellungen vorzunehmen. Folgende Einstellungen sollten vorgenommen werden:

[5] Das Betriebssystem kann unter `http://www.raspberrypi.org/downloads` gratis bezogen werden

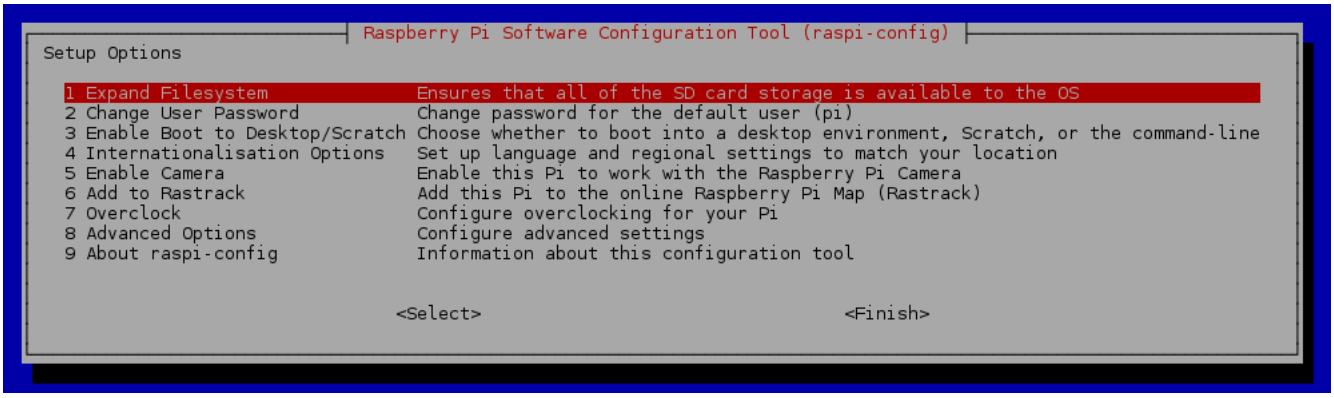

Abbildung 2: Raspberry Pi Konfigurations-Tool

1. Benutzerpasswort ändern (Change User Password)
2. Ausweiten des Dateisystems, um den Speicher der Karte voll auszunutzen (Expand Filesystem)
3. Anpassen des Keyboardlayouts (Internationalisation Options > Change Keyboard Layout)
4. Regionaleinstellungen auf en_US.UTF-8 setzen (Internationalisation Options > Change Locale)
5. Zeitregion setzen (Internationalisation Options > Change Timezone)
6. Raspberry Pi übertakten auf "Medium, 900MHz"(Overclock)
7. Speicherverteilung für die GPU auf 16 (Megabyte, MB) begrenzen (Advanced Options > A3 Memory Split)

Sollte es mit den getroffenen Einstellungen zu Problemen kommen, kann das Konfigurationstool im laufenden Betrieb erneut aufgerufen werden. Folgender Befehl muss dazu in der Konsole eingegeben werden:

```
sudo raspi-config
```

Sind alle Einstellungen vorgenommen, kann das Menü mittels *Finish* verlassen werden. Die Frage, ob neu gestartet werden soll (reboot), mit "Ja"beantworten, worauf das System neu startet und alle zuvor vorgenommenen Einstellungen übernommen werden.

Nach dem Neustart findet man sich in einem konsolenartigen Fenster mit einem blinkenden Cursor wieder. Dies wird für den Rest des Tutorials die Arbeitsumgebung sein, da die grafische Benutzeroberfläche nicht gebraucht wird. Die Performance ist in der Konsole zudem deutlich besser.

5.2 Root-Rechte erlangen

Die meisten der in dieser Anleitung beschriebenen Befehle verlangen erweiterte Rechte. Diese können unter Raspbian ganz einfach erlangt werden mittels:

```
sudo su
```

Grosse Macht bringt auch grosse Verantwortung. Hat man unter Linux Administrator-Rechte (auch Root-Rechte genannt), kann man sehr schnell sehr vieles kaputt machen. Im schlimmsten Fall muss die SD-Karte neu aufgesetzt werden. Es lohnt sich deshalb, ein paar wenige Regeln zum Gebrauch der Konsole zu beachten:

- Ein Befehl wird mittels Drücken der *Enter-Taste* abgesetzt
- Bevor ein neuer Befehl abgsetzt werden, muss der zuvor eingegebene abgeschlossen sein (manchmal ist Geduld gefragt)
- Gross- und Kleinschreibung werden unter Linux unterschieden!
- Immer sicherstellen, dass der Befehl auch wirklich richtig eingegeben wurde

Als Root-User sollte man nur dann unterwegs sein, wenn man die erweiterten Rechte für einen längeren Zeitraum benötigt, wie das in diesem Tutorial der Fall ist. Für einzelne Befehle kann auch *sudo* verwendet werden. Dieses Schlüsselwort wird einfach jedem Befehl vorangestellt, der erweiterte Rechte verlangt.
Ein Beispiel:

```
sudo apt-get update
```

Am Ende dieses Tutorials sollten die Root-Rechte unbedingt wieder abgegeben werden. Um das zu tun, reicht ein einfaches *exit* in der Kommandozeile und schon ist man wieder als normaler User unterwegs.

5.3 System aktualisieren

Um sicherzustellen, dass das System auf dem aktuellsten Stand ist, müssen zuerst alle installierten Pakete aktualisiert werden. Dazu müssen folgende zwei Befehle abgesetzt werden:

```
apt-get update
apt-get upgrade
```

Nach der Eingabe von *apt-get upgrade* fragt das Terminal noch einmal nach, ob man die zur Verfügung stehenden Pakete wirklich installieren will. Standardmässig ist die Antwort auf *Ja* eingestellt, was man an dem grossen Y in *[Y/n]* erkennt. Um fortzufahren reicht ein erneutes Drücken der Enter-Taste. Zukünftige Rückfragen bei abgesetzten Befehlen können auf die gleiche Weise behandelt werden. Es empfiehlt sich dennoch, die angezeigte Meldungen (Informationen, Warnungen) immer durchzulesen und entsprechend zu handeln.

```
pi@raspberrypi ~ $ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be upgraded:
  dpkg dpkg-dev libdpkg-perl libxfont1
4 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 5,029 kB of archives.
After this operation, 600 kB of additional disk space will be used.
Do you want to continue [Y/n]?
```

Abbildung 3: apt-get upgrade

6 WiFi-Adapter testen

Vor dem Aufsetzen des Tor-Proxys sollte zuerst überprüft werden, ob der angeschaffte WiFi-Adapter auf dem aufgespielten Betriebssystem lauffähig ist. Dazu steckt man den Adapter in den USB-Port und ruft nach kurzer Wartezeit folgendes Kommando auf:

```
ifconfig -a
```

Wenn auf der erschienenen Ausgabe ein Eintrag für *wlan0* zu sehen ist, kann mit der Einrichtung des Access Points begonnen werden.

```
pi@raspberrypi ~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:dc:9a:76
          inet addr:192.168.1.103  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:106 errors:0 dropped:1 overruns:0 frame:0
          TX packets:68 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10006 (9.7 KiB)  TX bytes:9056 (8.8 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

wlan0     Link encap:Ethernet  HWaddr 44:33:4c:10:6b:f0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Abbildung 4: ifconfig - wlan0

7 Access Point aufsetzen

Mit dem eingesteckten Netzwerkkabel und dem per USB angeschlossenen WiFi-Adapter stehen nun zwei Netzwerk-Schnittstellen zur Verfügung. Das Netzwerkkabel hat die Aufgabe,

den Raspberry Pi mit dem Internet zu verbinden. Der WiFi-Adapter hingegen soll so eingerichtet werden, dass er ein lokales WLAN (Wireless Local Area Network) aufzieht. Der Raspberry Pi fungiert somit als Wireless Access Point - auf deutsch so viel wie "kabelloser Zugangspunkt". Ein Access Point ermöglicht es Geräten wie Notebooks oder Smartphones, sich über ihn kabellos mit dem Internet zu verbinden. Neben der Software für den Access Point wird ein eigener DHCP-Server benögtigt. DHCP (Dynamic Host Configuration Protocol) ist ein Protokol, das die automatische Einbindung eines Computers in ein bestehendes Netzwerk ermöglicht. Der DHCP-Server sorgt also dafür, dass jeder Computer, der sich mit dem WLAN verbindet, eine valide IP-Adresse erhält und somit ein Teil des Netzes werden kann.
Mit den folgenden befehlen werden die benötigten Softwarekomponenten installiert.

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

7.1 DHCP-Server konfigurieren

Damit der DHCP-Server auch richtig funktioniert, muss er zuerst konfiguriert werden. Dazu öffnet man die Datei */etc/dhcp/dhcpd.conf* mit einem beliebigen Texteditor. In diesem Tutorial wird für diesen und alle nachfolgenden Fälle der vorinstallierte Texteditor *nano* verwendet.
Mit dem folgenden Kommando wird die Textdatei mit *nano* geöffnet:

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

Zuerst müssen die folgenden Zeilen gefunden und mittels einer vorangestellten Raute (#) auskommentiert werden. Auskommentieren bedeutet, dass die Zeile ungültig und somit nicht mehr aktiv ist.

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

Neu sieht das Ganze folgendermassen aus:

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

Als Nächstes muss dem DHCP-Server mitgeteilt werden, dass er der offizielle DHCP-Server des zu erstellenden WLAN-Netzes ist. Einmal eingestellt vergibt er fortan valide IP-Adressen an jeden Computer, der dem Netz beitreten möchte.
Dazu muss folgende Zeile einkommentiert - die vorangestellte Raute entfernt - werden:

```
#authoritative;
```

Die Zeile sollte nun so aussehen:

```
authoritative;
```

Das künftig vom WiFi-Adapter aufzuziehende WLAN muss einen eigenen Bereich zugewiesen bekommen, in dem es wirken kann.
Dazu einfach die folgenden Zeilen ans Ende der Datei anfügen:

```
subnet 192.168.66.0 netmask 255.255.255.0 {
  range 192.168.66.10 192.168.66.50;
  option broadcast-address 192.168.66.255;
  option routers 192.168.66.1;
  default-lease-time 600;
  max-lease-time 7200;
  option domain-name "local";
  option domain-name-servers 8.8.8.8, 8.8.4.4;
}
```

Der DHCP-Server ist fertig konfiguriert und muss an ein Netwerkinterface (Netzwerkschnittstelle) gebunden werden. Das gesuchte Netzwerkinterface ist in diesem Fall *wlan0*, wie man schon zu beginn mittels *ifconfig -a* herausgefunden hat. In der Datei */etc/default/isc-dhcp-server* muss dazu bei der Einstellung *INTERFACES* der Wert *wlan0* eingetragen werden:

```
INTERFACES="wlan0"
```

Dem Interface *wlan0* kann jetzt eine fixe IP-Adresse vergeben werden. Es handelt sich dabei um jene Adresse, die bei der Konfiguration des DHCP-Server bei *option routers* angegeben wurde - in diesem Fall also *192.168.66.1*.
In der Datei */etc/network/interfaces* müssen zuerst folgende Zeilen auskommentiert werden:

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

Anschliessend fügt man folgende Zeilen hinzu:

```
iface wlan0 inet static
  address 192.168.66.1
  netmask 255.255.255.0
```

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

iface wlan0 inet static
  address 192.168.66.1
  netmask 255.255.255.0
```

Abbildung 5: konfigurierte interfaces-Datei

Das Interface *wlan0* läuft zu diesem Zeitpunkt bereits. Um die eben getätigten Einstellungen sofort wirksam zu machen, muss noch folgendes Kommando abgesetzt werden:

```
ifconfig wlan0 192.168.66.1
```

7.2 Hostapd konfigurieren

Nach dem DHCP-Server muss jetzt der Access Point konfiguriert werden. Die Konfigurationsdatei für *hostapd* existiert aber noch nicht. Die neue Datei erstellt man mit:

```
touch /etc/hostapd/hostapd.conf
```

Die Datei sollte jetzt erstellt sein und kann mit einem Texteditor geöffnet werden:

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

Folgende Zeilen müssen hinzugefügt werden:

```
interface=wlan0
driver=rtl871xdrv
ssid=PiProxy
hw_mode=g
channel=6
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=Test1234
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Ein paar der obigen Werte kurz erklärt:

- driver: Der Treiber-Name des Wifi-Adapters
- ssid: Der Name des Netzes, wie man es nach aussen hin sieht
- wpa_passphrase: Passwort für den Zugang zum Netz

hostapd kennt diese neue Konfiguration noch nicht. Um diese dem Programm bekanntzumachen, muss in der Datei */etc/default/hostapd* nach *DAEMON_CONF* gesucht und die Zeile wie folgt geändert werden:

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

Der Raspberry Pi (Tor-Proxy) kann nun auf der einen Seite ein WLAN aufziehen und sich auf der anderen Seite mit dem Internet verbinden. Was jetzt noch fehlt ist die Verbindung dazwischen.

7.3 NAT konfigurieren

NAT (Network Address Translation) wird verwendet, um die mit dem WLAN verbundenen Geräte ins Internet weiterzuleiten. Der Datei */etc/sysctl.conf* muss dazu folgende Zeile angefügt werden:

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

Folgender Befehl macht die Einstellung umgehend aktiv:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Zusätzlich muss die Firewall so eingestellt werden, dass NAT die eingerichtete Weiterleitung durchführen kann:

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

iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j
    ACCEPT

iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
```

Damit man diese Befehle nicht bei jedem Neustart des Raspberry Pi eingeben muss, können sie permanent gespeichert werden:

```
iptables-save > /etc/iptables.ipv4.nat
echo "up iptables-restore < /etc/iptables.ipv4.nat" >> /etc/network/interfaces
```

8 Erster Start des Access Points

Der Access Point ist an diesem Punkt fast vollständig aufgesetzt. Um zu überprüfen, ob alles richtig gemacht wurde, muss man das Programm zusammen mit der Konfigurationsdatei aufrufen:

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

Entspricht die Ausgabe der unten Abgebildeten, waren die bisherigen Schritte erfolgreich und das nachfolgende Kapitel kann übersprungen werden.

```
root@raspberrypi:/home/pi# /usr/sbin/hostapd /etc/hostapd/hostapd.conf
Configuration file: /etc/hostapd/hostapd.conf
drv->ifindex=3
l2_sock_recv==l2_sock_xmit=0x0x162f638
+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 44:33:4c:10:6b:f0 and ssid 'PiProxy'
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 ignore_broadcast_ssid:0, PiProxy,7
rtl871x_set_acl
```

Abbildung 6: hostapd- Ausgabe vom Erststart

8.1 Fehlerbehandlung

Tritt ein Fehler auf, liegt das möglicherweise an einer Inkompatibilität zwischen *hostpad* und dem WiFi-Adapter bzw. dessen Treiber. Für den in diesem Tutorial verwendeten WiFi-Adapter kann das Problem gelöst werden, indem man *hostapd* selber kompilliert. Dazu müssen mehrere Schritte befolgt werden.
Achtung: Die Dateinamen und Verzeichnisse können sich je nach Fall von den hier Verwendeten unterscheiden und müssen dementsprechend angepasst werden.

Zuerst lädt man von der Hersteller-Seite den Linux-Treiber herunter. Dazu geht man auf `http://realtek.com` und navigiert zu *Downloads > Communications Network ICs > Wireless LAN ICs > WLAN NIC > IEEE 802.11b/g/n Single-Chip > Software*. Dort lädt man den Linux-Treiber für den vom WiFi-Adapter verwendeten Chipsatz herunter (hier RTL8192CU). Den heruntergeladenen Treiber (Zip-Datei) kopiert man auf einen USB-Stick und von dort in das Home-Verzeichnis des Raspberry Pi. Ist der USB-Stick am Pi angeschlossen, müssen dazu folgende Befehle abgesetzt werden:

```
mount /dev/sda1 /mnt
cp /mnt/RTL8188C_8192C_USB_linux_v4.0.2_9000.20130911.zip /home/pi
umount /dev/sda1
```

Anschliessend entpackt man die Zip-Datei:

```
cd /home/pi
unzip RTL8188C_8192C_USB_linux_v4.0.2_9000.20130911.zip
```

Danach wechselt man in das richtige Verzeichnis und entpackt dort das Archiv, welches *hostapd* enthält:

```
cd wpa_supplicant_hostapd
tar -xzvf wpa_supplicant_hostapd-0.8_rtw_r7475.20130812.tar.gz
```

Danach wechselt man wiederum in das soeben entpackte Verzeichnis, wechselt in den Ordner namens *hostapd* und startet den Kompiliervorgang:

```
cd wpa_supplicant_hostapd-0.8_rtw_r7475.20130812/hostapd
make
```

Nach ein paar Minuten Wartezeit steht die frisch kompilierte, binäre Datei bereit. Diese wird in Zukunft die zu Beginn des Tutorials installierte Version ersetzen. Zuvor sollte man aber sicherheitshalber die Original-Datei sichern:

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

Anschliessend kann man die neue *hostapd*-Binärdatei kopieren:

```
cp hostapd /usr/sbin
```

Jetzt ist es an der Zeit, die ersten Schritte des Hauptkapitels *Erster Start des Access Points* noch einmal auszuführen. Funktioniert es dieses Mal, kann mit dem nächsten Kapitel fortgefahren werden. Kommt es erneut zu Fehlern, sollten alle vorherigen Schritte genau überprüft und allenfalls die Internet-Community zu Rate gezogen werden.

8.2 Verbindung herstellen zum Access Point

Bevor man sich mit dem Access Point verbinden kann, müssen die beiden Komponenten *hostapd* und *isc-dhcp-server* gestartet werden:

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

Das WLAN sollte jetzt aktiv und für jedes WiFi-fähige Geräte erkennbar sein.

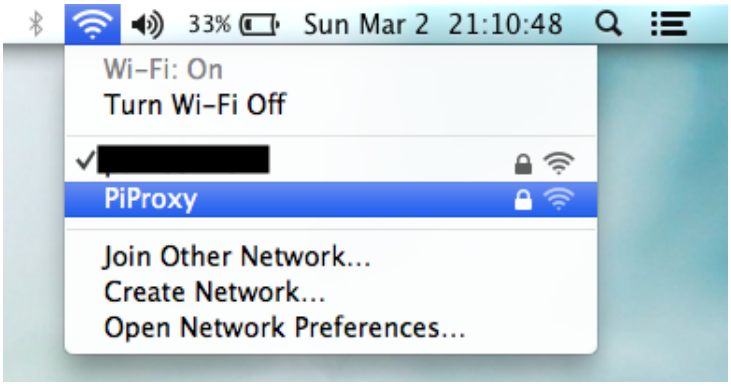

Abbildung 7: Der erstellte Access Point

9 Daemons einrichten

Beide Komponenten sollten als sogenannter *Daemon* eingerichtet werden. Daemons sind Programme, die bei jedem Computerstart ebenfalls gestartet werden und fortan im Hintergrund laufen. Man will den Access Point schliesslich nicht jedes Mal manuell über die Konsole starten müssen.

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

Um die Daemons auf korrekte Funktionsweise zu überprüfen, muss der Tor-Proxy neu gestartet werden:

```
shutdown -r now
```

Ist der Computer wieder hochgefahren, fragt man die Stati der beiden Daemons ab:

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

Die Ausgabe sollte wie folgt aussehen:

```
root@raspberrypi:/home/pi# service hostapd status
[ ok ] hostapd is running.
```

Abbildung 8: Status Access Point

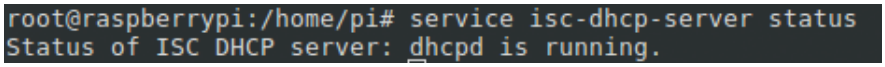

Abbildung 9: Status DHCP-Server

Verhält sich der Access Point nicht wie gewünscht oder läuft gar nicht erst, empfiehlt es sich, alle vorherigen Schritte noch einmal zu studieren und die Konfigurationen zu überprüfen. Für schwer zu lösende Probleme kann man sich auch immer an die Internet-Gemeinschaft wenden. Waren alle bisherigen Schritte erfolgreich, kann jetzt damit begonnen werden, den eigentlichen Tor-Proxy aufzusetzen.

10 Tor aufsetzen

An dieser Stelle ist es möglich, sich mittels Access Point mit dem Internet zu verbinden und loszusurfen. Das Ziel ist es aber, die Verbindung ins Internet über das Tor-Netzwerk zu leiten, um ein gewisses Mass an Anonymität zu gewährleisten. Dafür muss als erstes die entsprechende Softwarekomponente installiert werden. Tor kann man - wie die zuvor verwendeten Programme auch - über die Paketverwaltung installieren:

```
apt-get install tor
```

Nach der Installation muss Tor natürlich noch richtig konfiguriert werden:

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

Direkt nach dem einleitenden Kommentarblock müssen folgende Zeilen hinzugefügt werden:

```
Log notice file /var/log/tor/notices.log
VirtualAddrNetwork 10.192.0.0/10
AutomapHostsSuffixes .onion,.exit
AutomapHostsOnResolve 1
TransPort 9040
TransListenAddress 192.168.66.1
DNSPort 53
DNSListenAddress 192.168.66.1
```

Die Datei sollte nun folgendermassen aussehen:

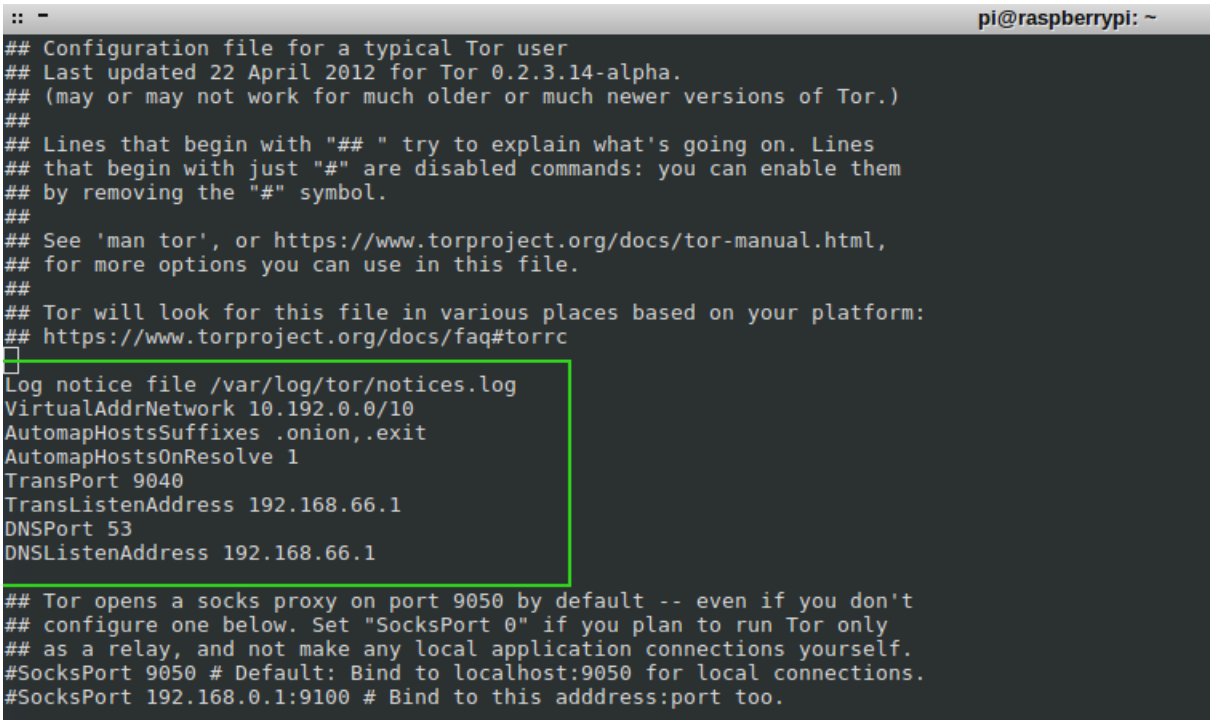

Abbildung 10: torrc-Datei

Der Firewall muss jetzt noch beigebracht werden, dass jegliche Kommunikation vom WLAN ins Internet über das Tor-Netzwerk geleitet werden soll:

```
iptables -F

iptables -t nat -F

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

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

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

iptables-save > /etc/iptables.ipv4.nat
```

10.1 Protokollierung aktivieren

In der *torrc*-Konfigurationsdatei wurde bei *Log notice file* eine Datei angegeben, die in Zukunft alle Logs enthält. Mit *Logs* sind Protokolleinträge gemeint, die von der Tor-Software erstellt werden und Auskunft über den Status und allfällige Probleme geben. Diese Datei muss erstellt und mit den entsprechenden Rechten versehen werden:

```
touch /var/log/tor/notices.log
chown debian-tor /var/log/tor/notices.log
```

```
chmod 644 /var/log/tor/notices.log
```

Nun muss man den Tor-Service neu starten, damit die vorgenommenen Einstellungen aktiv werden:

```
service tor restart
```

Um zu prüfen, ob Tor erfolgreich gestartet werden konnte, kann man folgenden Befehl ausführen:

```
service tor status
```

Die Ausgaben sollten wie die unten Abgebildeten aussehen.

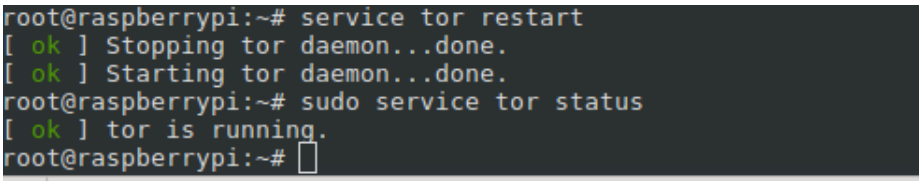

Abbildung 11: Tor - Neustart und Status

Auch Tor soll bei jedem Start des Systems automatisch laufen. Um Tor als Daemon zu konfigurieren, muss man analog zu vorhin folgenden Befehl absetzen:

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

Der Access Point und Tor sind an diesem Punkt fertig installiert und eingerichtet. Gratuliere! Jetzt fehlt nur noch ein abschliessender Test, um sicherzustellen, dass alles so funktioniert, wie es soll.

11 IP-Adresse verifizieren

Mittels `http://wieistmeineip.ch` kann die IP-Adresse ermittelt werden, mit der man sich im Internet bewegt. Diese wird einem normalerweise vom Provider zugeteilt und ändert sich eher selten. Mit einem funktionierenden Tor-Setup ändert sich dieses Verhalten jedoch. Da je Anfrage ins Internet drei verschiedene Tor-Server passiert, ändert sich die Absender-IP-Adresse des Paketes jedes Mal. Die angefragte Internetseite sieht somit nie die IP-Adresse, die einem vom Provider zugeteilt wurde, sondern immer die des zuletzt passierten Tor-Servers - auch Exit-Node genannt. Um die korrekte Funktionsweise des Tor-Proxys zu verifizieren, muss man lediglich von einem Computer im normalen Netz und einem Gerät, das mittels Tor-Proxy mit dem Internet verbunden is, die Seite *wieistmeineip* aufrufen. Die IP-Adressen und auch die sonstigen angezeigten Informationen dürfen sich bei einem funktionierenden Tor-Proxy nicht decken.

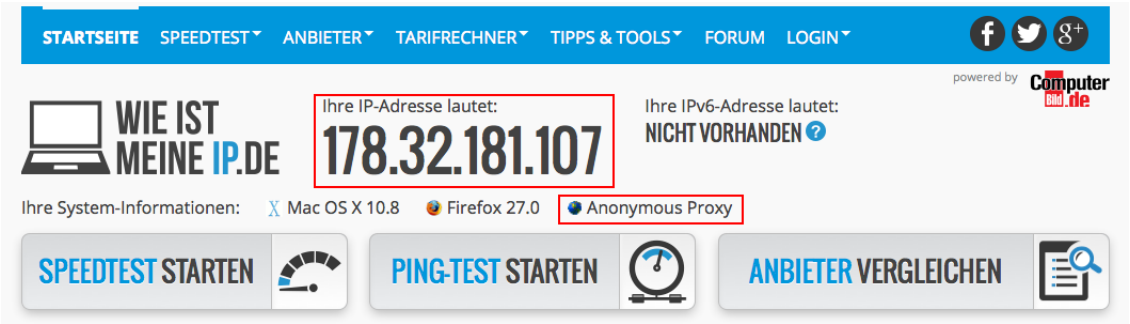

Abbildung 12: wieistmeineip.ch mit funktionierendem Tor-Proxy

Zusätzlich dazu kann die Seite `https://check.torproject.org/` besucht werden. Diese zeigt einem an, ob man mittels Tor im Internet unterwegs ist und liefert gleich noch ein paar hilfreiche Links.

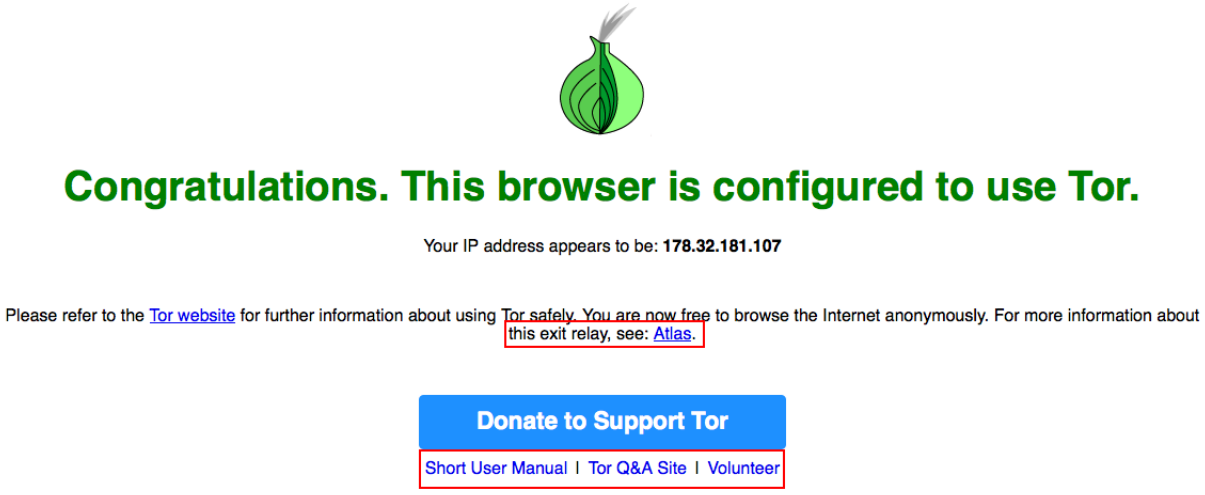

Abbildung 13: Tor-Check

12 Exit-Nodes konfigurieren

Will man den Tor-Proxy bzw. deren Verwendung sicherer gestalten, lohnt es sich, die zugelassenen Exit-Nodes festzulegen. Der letzte Tor-Server auf dem Weg zum Ziel ist bekanntlich der Exit-Node. Er übernimmt das versendete Paket vom vorherigen Tor-Server, entpackt es und leitet es schliesslich an die Zieladresse weiter. Entpacken bedeutet, dass er je nach verwendetem Protokoll - http oder https - alle Details des Pakets einsehen kann. Da jeder einen solchen Exit-Node betreiben kann - gut gesinnte Privatpersonen oder Organisationen, Geheimdienste oder sonstige Kriminelle - kann man nie genau wissen, wer jetzt einen sogenannten *Bad-Exit-Node* betreibt. Glücklicherweise kann dieses Risiko minimiert werden. In der Tor-Konfigurationsdatei *torrc* kann man festlegen, welche Exit-Nodes verwendet werden

sollen und/oder nur Exit-Nodes aus gewissen Ländern zulassen. Es gibt diverse Organisationen und Verbünde, die solche Tor-Exit-Nodes betreiben und zum Gebrauch anbieten. Eine davon ist die Swiss Privacy Foundation: `http://privacyfoundation.ch/`.
Um die Exit-Nodes einzuschränken bzw. nach seinen persönlichen Wünschen zu selektieren, gibts es zwei Möglichkeiten, die sich mitunter sogar kombinieren lassen:

1. Angabe des *Fingerprints* (eindeutige ID) eines spezifischen Exit-Nodes
2. Angabe eines Ländercodes, wodurch nur Exit-Nodes von diesem Land verwendet werden

Die Einstellungen werden in der *torrc*-Konfigurationsdatei vorgenommen. Diese muss man zuerst mittels Texteditor öffnen:

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

Die folgenden Zeilen werden nach *DNSListenAddress 192.168.66.1* angefügt:

```
StrictNodes 1
ExitNodes Fingerprint1,Fingerprint2,{Laendercode1},{Laendercode2},{
    Laendercode3}
```

Beim Fingerprint handelt es sich um eine lange Kombination aus Zahlen und Buchstaben mit einem vorangestellten $-Zeichen. Beispielsweise hat ein von der *Swiss Privacy Foundation* bereitgestellter Exit-Node den Fingerprint $944224E9413705EEAFCBAC98BF57C475EB1960C5.
Zwischen den geschweiften Klammern kann ein Ländercode angegeben werden. Für Exit-Nodes aus der Schweiz müsste man {ch} verwenden. Man kann dabei so viele Länder angeben wie man will.
Wichtig ist, dass alle angegebenen Optionen (Ländercodes und Fingerprints) stets mittels Komma getrennt werden und keine Leerzeichen dazwischenstehen.

Ein weiteres Beispiel soll ein bisschen Klarheit schaffen:

```
ExitNodes $944224E9413705EEAFCBAC98BF57C475EB1960C5,{de},{at}
```

Mit dieser Einstellung werden nur deutsche und österreichische Exit-Nodes verwendet inklusive einem zu Bginn einzeln definierten Exit-Node.

13 Abschluss

Tor bietet noch viele andere Möglichkeiten, den Dienst den eigenen Ansprüchen anzupassen und sicherer zu machen. Details zu den unterschiedlichen Konfigurationen und Modi findet man hier: `https://www.torproject.org/docs/tor-manual.html.en`. Die Seite ist zur Zeit leider nur auf Englisch verfügbar. Es gibt zudem gewisse Verhaltensregeln, die für einen sinnvollen und sicheren Gebrauch von Tor befolgt werden sollten:

- Verwendung vom verschlüsselten https-Protokoll an Stelle von http wenn immer möglich!
- Keine Soziale Netzwerke besuchen!

- Sich nirgends einloggen oder registrieren - zumindest nicht mit Daten, die Rückschlüsse auf die Identität zulassen!
- Cookies weitgehend deaktivieren, JavaScript deaktivieren

Achtung: Ein falscher und verantwortungsloser Gebrauch von Tor birgt mehr Gefahren, als wenn man ganz auf den Dienst verzichten würde. Deshalb sollte man sich vor Gebrauch von Tor gut über die verschiedenen Funktionen und Risiken informieren. Im Internet findet man allerhand an Material und Informationen, um sattelfest auf dem Gebiet zu werden. Ist man einmal sattelfest, steht einem sinnvollen Gebrauch von Tor nichts mehr im Weg. Viel Spass!

Abbildungsverzeichnis

1 Raspberry Pi, Model B 3
2 Raspberry Pi Konfigurations-Tool 6
3 apt-get upgrade 8
4 ifconfig - wlan0 8
5 konfigurierte interfaces-Datei 10
6 hostapd- Ausgabe vom Erststart 12
7 Der erstellte Access Point 14
8 Status Access Point 14
9 Status DHCP-Server 15
10 torrc-Datei 16
11 Tor - Neustart und Status 17
12 wieistmeineip.ch mit funktionierendem Tor-Proxy 18
13 Tor-Check 18