Point d'accès wifi sur Tor avec Raspberry

- Objet : création d'un point d'accès wifi Tor avec un Raspberry
- Niveau requis : moyen
- Débutant, à savoir : Utiliser GNU/Linux en ligne de commande, tout commence là !.
- Suivi :
  à-placer
  - Création par vakuy 02/04/2018
  - Testé par <...> le <...> Fix Me!
- Commentaires sur le forum : Lien vers le forum [1)]

Remarque préliminaire

les instructions ci-dessous sont pour leur majeur partie tirées de ces tutos (en anglais) :

- https://learn.adafruit.com/setting-up-a-raspberry-pi-as-a-wifi-access-point
- https://learn.adafruit.com/onion-pi/overview

Introduction

Le but de ce tutorial est de créer, à l'aide d'un raspberry, un point d'accès wifi (“hotspot”) qui route l'ensemble du traffic des appareils qui s'y connectent par Tor.

À quoi ça sert

Lorsque vous connectez un appareil à ce hotspot, l'ensemble du trafic est redirigé sur Tor. Cela peut être utile dans les cas de figure suivants:

- Vous cherchez un moyen facile de faire transiter l'ensemble de votre trafic internet sur Tor.
- Vous souhaitez isoler des appareils connectés au net auxquels vous ne faites pas confiance (ex. frigos, balances, etc.)
- Vous avez envie de réaliser un projet avec votre Raspberry qui traîne

Ce n'est probablement pas pour vous si

- Vous cherchez un moyen très sécurisé de vous connecter à Tor et/ou vous avez besoin d'un niveau d'anonymat et de sécurité maxium. Dans ce cas il vaut certainement mieux vous tourner vers des solutions comme Tails (https://tails.boum.org/) ou Whonix (https://www.whonix.org/).
- Vous comptez vous y connecter avec un smartphone ou tout autre appareil où vous êtes déjà identifiés à des services (ex. apps diverses, gmail, etc.).

Installation

Matériel requis

- Une connexion internet (sans blague…)
- Un Raspberry (2 ou 3 ou Zero) avec Raspbian installé
- Un ou deux adaptateurs Wifi (j'utilise un Asus USB-N13, mais il y a bien sûr beaucoup d'autres choix)
- Un câble ethernet (optionnel, pas nécessaire si vous avez deux adaptateurs wifi, ou alors un Raspberry 3 ou Zero avec carte wifi et un adaptateur)
- Optionnel, mais recommandé: une connexion ssh à votre raspberry pour le configurer à distance.

Installation et configuration du point d'accès wifi

Remarque : *j'utilise ici une configuration avec un Raspberry 3 connecté en wifi avec sa puce native (wlan0) et avec une deuxième interface wifi (wlan1) correspondant à l'adaptateur wifi.*

1. Premiers pas

On commence par une petite mise à jour du système, cela ne peut pas faire de mal :

```
sudo apt update && sudo apt dist-upgrade
```

On vérifie que les deux interfaces désirées sont reconnues :

```
ifconfig -a
```

```
eth0      Link encap:Ethernet  HWaddr XXX
          inet6 addr: XXX Scope:Link
          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)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:16514 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16514 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
```

```
          RX bytes:35719859 (34.0 MiB)  TX bytes:35719859 (34.0 MiB)

wlan0     Link encap:Ethernet  HWaddr XXX
          inet addr:192.168.1.100  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::93d8:59fb:5c7a:1a86/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:378004 errors:0 dropped:0 overruns:0 frame:0
          TX packets:206942 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:205328121 (195.8 MiB)  TX bytes:101559626 (96.8 MiB)

wlan1     Link encap:Ethernet  HWaddr XXX
          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:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

comme on le voit ici, l'interface wlan1, qui correspond à l'adaptateur wifi, est reconnue mais pas encore configurée. C'est elle qui servira de point d'accès wifi

2. installation des paquets

- hostapd
- isc-dhcpd-server et
- iptables-persistent

1. Les paquets hostapd et isc-dhcpd-server vont nous permettre de créer un accès wifi avec notre Raspberry.

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

2. Le paquet iptables-persistent servira à sauvegarder les règles iptables au redémarrage du Raspberry.

   ```
   sudo apt install iptables-persistent
   ```

Pendant l'installation, on vous demandera si vous souhaitez sauvegarder les règles actuelles pour ipv4 et ipv6. Dites oui aux deux.

3. Configuration du serveur DHCP

Il s'agit maintenant d'éditer le fichier `/etc/dhcp/dhcpd.conf`, pour permettre à notre futur point d'accès d'attribuer automatiquement les plages d'adresses IP locales et de gérer les DNS.

Commencez par faire un backup du fichier :

```
sudo cp /etc/dhcp/dhcpd.conf /etc/dhcp dhcpd.conf.back
```

Ensuite remplacez l'ensemble du contenu du fichier `/etc/dhcp/dhcpd.conf` par les lignes suivantes :

```
authoritative;
default-lease-time 600;
max-lease-time 7200;
subnet 192.168.42.0 netmask 255.255.255.0
{
        option subnet-mask 255.255.255.0;
        option broadcast-address 192.168.42.255;
        option routers 192.168.42.1;
        option domain-name-servers 192.168.42.1;
        range 192.168.42.1 192.168.42.100;
}
```

Ensuite, éditez le fichier `/etc/default/isc-dhcp-server`

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

Et ajoutez remplacez `INTERFACES=“”` par `INTERFACES=“wlan1”` ainsi :

```
# Defaults for isc-dhcp-server initscript
# sourced by /etc/init.d/isc-dhcp-server
# installed at /etc/default/isc-dhcp-server by the maintainer scripts

#
# This is a POSIX shell fragment
#

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPD_CONF=/etc/dhcp/dhcpd.conf

# 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="wlan1"
```

Important : si vous n'avez pas d'adaptateur wifi et que vous utilisez votre puce wifi (Raspberry 3 ou Zero) comme point d'accès, indiquez “wlan0” au lieu de “wlan1”. L'interface doit être celle de votre futur point d'accès.

4. Configuration de l'interface wifi du point d'accès

1. Allez dans le fichier /etc/network/interfaces :

```
sudo nano /etc/network/interfaces
```

2. Supprimez ou commentez toutes références à votre interface wifi qui servira de point d'accès et ajoutez les lignes suivantes :

```
iface wlan1 inet static
address 192.168.42.1
netmask 255.255.255.0
```

Important : même chose que précédemment, remplacez wlan1 par wlan0 si vous utilisez votre puce wifi comme point d'accès.

Après avoir sauvegardé vos modifications, attribuez l'adresse statique 192.168.42.1 à votre point d'accès :

```
sudo ifconfig wlan1 192.168.42.1
```

ou wlan0...

5. Configuration du point d'accès

Nous allons maintenant configurer le point d'accès, et lui donner un mot de passe.
Créer un nouveau fichier `hostapd.conf` en entrant la commande suivante :

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

et entrez les lignes suivantes :

```
interface=wlan1
ssid=Onion
country_code=FR
hw_mode=g
channel=11
macaddr_acl=0
#auth_algs=1
ignore_broadcast_ssid=0
wpa=1
wpa_passphrase=raspberry
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
wpa_group_rekey=86400
ieee80211n=1
```

```
wme_enabled=1
```

Important : *libre à vous de modifier le nom du réseau et le mot de passe. J'ai ici choisi “Onion” comme nom de réseau (ssid) et “raspberry” comme mot de passe. Remplacez wlan1 par wlan0 en fonction de votre configuration (même remarque que précédemment).*

Une fois le fichier sauvegardez, éditez le fichier /etc/default/hostapd:

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

et modifiez la ligne :

/etc/default/hostapd

```
#DAEMON_CONF=""
```

ainsi :

/etc/default/hostap

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

```
# 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.
#
#DAEMON_OPTS=""
```

Même chose avec le fichier /etc/init.d/hostapd, remplacez :

/etc/init.d/hostapd

```
DAEMON_CONF=
```

par :

/etc/init.d/hostapd

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

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

```
#!/bin/sh

### BEGIN INIT INFO
# Provides:     hostapd
# Required-Start:   $remote_fs
# Required-Stop:    $remote_fs
# Should-Start:     $network
# Should-Stop:
# Default-Start:    2 3 4 5
# Default-Stop:     0 1 6
# Short-Description:    Advanced IEEE 802.11 management daemon
# Description:      Userspace IEEE 802.11 AP and IEEE 802.1X/WPA/WPA2/EAP
#           Authenticator
### END INIT INFO

PATH=/sbin:/bin:/usr/sbin:/usr/bin
DAEMON_SBIN=/usr/sbin/hostapd
DAEMON_DEFS=/etc/default/hostapd
DAEMON_CONF=/etc/hostapd/hostapd.conf
NAME=hostapd
DESC="advanced IEEE 802.11 management"
PIDFILE=/run/hostapd.pid

[ -x "$DAEMON_SBIN" ] || exit 0
[ -s "$DAEMON_DEFS" ] && . /etc/default/hostapd
[ -n "$DAEMON_CONF" ] || exit 0

DAEMON_OPTS="-B -P $PIDFILE $DAEMON_OPTS $DAEMON_CONF"

. /lib/lsb/init-functions

case "$1" in
  start)
    log_daemon_msg "Starting $DESC" "$NAME"
```

```
    start-stop-daemon --start --oknodo --quiet --exec "$DAEMON_SBIN" \
        --pidfile "$PIDFILE" -- $DAEMON_OPTS >/dev/null
    log_end_msg "$?"
    ;;
  stop)
    log_daemon_msg "Stopping $DESC" "$NAME"
    start-stop-daemon --stop --oknodo --quiet --exec "$DAEMON_SBIN" \
        --pidfile "$PIDFILE"
    log_end_msg "$?"
    ;;
  reload)
    log_daemon_msg "Reloading $DESC" "$NAME"
    start-stop-daemon --stop --signal HUP --exec "$DAEMON_SBIN" \
        --pidfile "$PIDFILE"
    log_end_msg "$?"
    ;;
  restart|force-reload)
    $0 stop
    sleep 8
    $0 start
    ;;
  status)
    status_of_proc "$DAEMON_SBIN" "$NAME"
    exit $?
    ;;
  *)
    N=/etc/init.d/$NAME
    echo "Usage: $N {start|stop|restart|force-reload|reload|status}" >&2
    exit 1
    ;;
esac

exit 0
```

6. Installation et configuration de Tor

Installez le paquet Tor :

```
sudo apt install tor
```

Ajoutez les lignes suivantes à la fin du fichier /etc/tor/torrc:

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

```
VirtualAddrNetwork 10.192.0.0/10
AutomapHostsSuffixes .onion,.exit
AutomapHostsOnResolve 1
TransPort 192.168.42.1:9040
TransListenAddress 192.168.42.1
```

```
DNSPort 192.168.42.1:53
DNSListenAddress 192.168.42.1
```

Note: si vous utilisez une version antérieure à Debian Buster, ajoutez les lignes suivantes à la fin du fichier /etc/tor/torrc:

```
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
```

7. Configuration du NAT

Enclenchez l'ip forwarding au démarrage de la machine en ajoutant `net.ipv4.ip_forward=1` sur une nouvelle ligne tout à la fin du fichier /etc/sysctl.conf :

```
sudo nano /etc/systctl.conf
```

```
# Do not send ICMP redirects (we are not a router)
#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
#
net.ipv4.ip_forward=1
```

Puis entrez la commande suivante pour activer l'ip forwarding immédiatement :

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

Maintenant il s'agit d'entrer les règles iptables qui vont permettre de router les connexions au point d'accès (wlan1 chez moi) par Tor.
On commence par supprimer les règles existantes, au cas où :

```
sudo iptables -F
```

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

Ensuite on entre les deux commandes suivantes :

```
sudo iptables -t nat -A PREROUTING -i wlan1 -p udp --dport 53 -j
```

```
REDIRECT --to-ports 53
```

```
sudo iptables -t nat -A PREROUTING -i wlan1 -p tcp --syn -j REDIRECT --to-ports 9040
```

Remarque: remplacer wlan1 par wlan0 si besoin...

Enfin, on sauvegarde ces règles pour qu'elles persistent au redémarrage du Raspberry :

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

Utilisation

Premier test

Testez manuellement votre point d'accès en entrant la commande suivante :

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

Si tout fonctionne, vous devriez voir apparaître un nouveau réseau wifi appelé `Onion`[2].
Essayez de vous y connecter et testez votre adresse IP dans un navigateur, en vous rendant sur une page qui vous renseigne sur votre adresse IP et votre navigateur (il en existe des centaines), comme `whoer.net`.

Si vous voyez que vous avez une adresse qui ne correspond pas à la vôtre, d'un pays étranger, bravo, tout fonctionne comme prévu !

Activer le service de manière permanente

Entrez les commandes suivantes pour activer votre point d'accès au démarrage de la machine :

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

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

Remarques finales

Il y a des chances que vous n'y arriviez pas du premier coup.

1. En cas d'erreurs, vérifiez bien tous les fichiers que vous avez modifiés, la moindre erreur de

syntaxe étant fatale.
2. Même lorsqu'il fonctionne, le système n'est pas d'une fiabilité totale et peut s'interrompre régulièrement.

Les commandes suivantes :

```
sudo service hostapd status
```

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

vous permettent d'inspecter l'état des services du point d'accès.
Cela peut vous donner des indications précieuses en cas d'erreur.

Vous pouvez également vérifier que Tor fonctionne :

```
sudo service tor status
```

Enfin, si vous n'y arrivez vraiment pas, je vous suggère de tout reprendre depuis le début en vous référant à la documentation à l'origine d'adafruit :

- https://learn.adafruit.com/setting-up-a-raspberry-pi-as-a-wifi-access-point
- https://learn.adafruit.com/onion-pi/overview

Peut-être qu'il vaut mieux d'abord s'assurer que vous parvenez à faire fonctionner un point d'accès wifi simple, avant d'essayer de le torréifier.

Remarques sur Tor

Ce point d'accès wifi à lui seul ne suffit pas à vous rendre anonyme !
Si vous vous y connectez avec vos appareils habituels, les cookies de vos navigateurs, les préférences de vos apps, etc… suffisent à vous déanonymiser !
De plus, vous pourrez rencontrer des problèmes à vous connecter à vos services habituels (messageries, etc.).

À utiliser avec précaution !

1)
N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !
2)
ou tout autre nom que vous avez indiqué dans le fichier hostapd.conf plus haut

From:
http://debian-facile.org/ - **Documentation - Wiki**

Permanent link:
http://debian-facile.org/doc:reseau:wifi:raspberry:hotspot:tor

Last update: **18/01/2021 14:17**