

How to setup a bride?

A Tor Bridge ...

This guide is not finished yet!

Hardware

As minimum hardware requirements we recommend the following:

| | |
|------------|----------------|
| RAM | 256 MB |
| HDD | 5 GB Diskspace |
| CPU | Singlecore |

A Bridge doesn't need much of anything. Some more RAM could be useful if you decide to run more than one Bridge on the server.

Installation

Ubuntu 14.02 LTS

Tor

```
root@hostname:~# echo "deb http://deb.torproject.org/torproject.org trusty
main" >> /etc/apt/sources.list.d/tor
root@hostname:~# gpg --keyserver keys.gnupg.net --recv 886DDD89
root@hostname:~# gpg --export A3C4F0F979CAA22CDBA8F512EE8CBC9E886DDD89 |
sudo apt-key add -
root@hostname:~# apt-get update
root@hostname:~# apt-get install deb.torproject.org-keyring
root@hostname:~# apt-get install tor
```

obfsproxy

```
root@hostname:~# apt-get update
root@hostname:~# apt-get install obfsproxy
```

obfs4proxy

```
root@hostname:~# echo "deb http://deb.torproject.org/torproject.org
obfs4proxy main" >> /etc/apt/sources.list.d/tor
root@hostname:~# apt-get update
```

```
root@hostname:~# apt-get install obfs4proxy
```

fteproxy

```
root@hostname:~# echo "deb http://fteproxy.org/deb/ stable/" >
/etc/apt/sources.list.d/tor
root@hostname:~# gpg --keyserver keys.gnupg.net --recv 6B898EE18FBA6390
root@hostname:~# gpg --export 6B898EE18FBA6390 | sudo apt-key add -
root@hostname:~# apt-get update
root@hostname:~# apt-get install fteproxy
root@hostname:~# apt-get update
root@hostname:~# apt-get install python-dev python-pip libgmp-dev build-
essential
root@hostname:~# pip install fteproxy
```

flashproxy

```
root@hostname:~# apt-get update
root@hostname:~# apt-get install git golang
root@hostname:~# git clone https://git.torproject.org/pluggable-
transports/websocket.git
root@hostname:~# go get git.torproject.org/pluggable-transports/goptlib.git
root@hostname:~# cd websocket; make; make install
```

...

```
<sxh bash;> ExtORPort auto ServerTransportPlugin websocket exec /usr/local/bin/pt-websocket-
server -port 9901 </sxh>
```

Configuration

```
root@hostname:~# wget https://bitbucket.org/fvde/tor-
autoconfig/raw/tip/autoconf.pl
root@hostname:~# perl autoconf.pl bridge [YOUR-NODE-NICKNAME] [NETWORK
SPEED] [METERED|UNMETERED] ([TRAFFIC LIMIT])
root@hostname:~# mkdir /opt/ennstatus
root@hostname:~# wget -O /opt/ennstatus/update_server.pl
https://bitbucket.org/virii/update-ennstatus/raw/tip/update_server.pl
root@hostname:~# crontab -e
```

...

```
<sxh bash;> */10 * * * * perl /opt/ennstatus/update_server.pl torrc </sxh>
```

Launching the Bridge

Control your torrc. A functional Bridge torrc has to look like [this!](#) `<sxh bash;>service tor restart</sxh>`

Get int touch with info@enn.lu to register your IP.

Troubleshooting

[warn] Your server (IP:PORT) has not managed to confirm that its ORPort is reachable. Please check your firewalls, ports, address, /etc/hosts file, etc.

```
sudo nmap -sT -O IP
```

Among all other information nmap outputs, this one needs to be equivalent to this:

```
...  
9001/tcp open      tor-orport  
...
```

The command

```
root@hostname:~# netcat -w1 -vzn IP PORT
```

needs to return

```
IP PORT (etlservicemgr) open
```

According to [this](#) thread on the Tor Developer mailing list, it doesn't necessarily needs to come down to your parameters, but simply to the fact, that no one has yet connected to your bridge.

apparmor

Apparmor can sometimes cause trouble as it blocks the access to directories or refuse to start/stop pluggable transports. The following is a working system_tor apparmor config (works for obfsproxy, FTEproxy, obfs4proxy and flashproxy). As we use other directories for our hidden services, we had to put them in there as well.

```
#include <tunables/global>  
  
profile system_tor {  
    #include <abstractions/tor>  
  
    owner /var/lib/tor/** rwk,  
    owner /var/log/tor/* w,
```

```
/usr/local/bin/obfs4proxy PUx,  
/usr/local/bin/obfsproxy PUx,  
/usr/local/bin/fteproxy PUx,  
/usr/local/bin/pt-websocket-server PUx,  
  
/home/hidden_service_0/hostname rw,  
/home/hidden_service_0/hostname.tmp rw,  
/home/hidden_service_0/private_key rw,  
/home/hidden_service_0/private_key.tmp rw,  
/home/hidden_service_1/hostname rw,  
/home/hidden_service_1/hostname.tmp rw,  
/home/hidden_service_1/private_key rw,  
/home/hidden_service_1/private_key.tmp rw,  
/home/hidden_service_2/hostname rw,  
/home/hidden_service_2/hostname.tmp rw,  
/home/hidden_service_2/private_key rw,  
/home/hidden_service_2/private_key.tmp rw,  
  
/{,var/}run/tor/control w,  
/{,var/}run/tor/tor.pid w,  
/{,var/}run/tor/control.authcookie w,  
/{,var/}run/tor/control.authcookie.tmp rw,  
  
signal (send) set=("term") peer="unconfined",  
  
#include <local/system_tor>  
}
```

tor-arm

From:
<http://wiki.enn.lu/> - Frënn vun der Ënn A.S.B.L.

Permanent link:
<http://wiki.enn.lu/doku.php?id=howto:bridge>



Last update: **2015/03/20 18:20**