HOWTO use the Internet anonymously using Tor and Privoxy

This HOWTO will allow you to browse the web and user other Internet services (IRC, Usenet) anonymously using the Tor onion router and Privoxy.

Contents

The basic setup

First, install the needed software.

Fedora Core / Debian / Ubuntu apt-users:

```
apt-get install privoxy tor
```

Gentoo users:

```
emerge net-misc/tor net-proxy/privoxy
```

(You can also download the source packages from http://tor.eff.org/ and http://www.privoxy.org/ and ./configure && make && make install as normal.)

Privoxy

Add this to **/etc/privoxy/config** (anywhere, the end of the file is always good):

```
forward-socks4a / localhost:9050 .
```

Replace localhost with the IP if you plan on running Tor on another server on your local network (like your firewall):

```
forward-socks4a / 192.168.1.20:9050 .
```

That is actually all you need, now you can start privoxy:

```
/etc/init.d/privoxy start
```

And perhaps make it start at boot?

Fedora:

```
chkconfig privoxy on
```

Gentoo:

```
rc-update add privoxy default
```

Tor

The default configuration example is setup to run Tor in *client* mode and works out-of-the-box:

```
cp /etc/tor/torrc.sample /etc/tor/torrc
```

Change "SocksBindAddress" and set a "SocksPolicy" if you want to connect to the Tor service from another computer:

File: /etc/tor/torrc

```
SocksBindAddress 192.168.1.20
SocksPolicy accept 192.168.1.1/16
SocksPolicy reject *
```

..and now all you need to do is to start Tor:

```
/etc/init.d/tor start
```

Software Configuration

Shell programs like wget, lynx and curl will read the shell's proxy variables (if any) and use them. You likely want to set these variables:

File: ~/.bashrc or ~/.profile:

```
http_proxy=http://127.0.0.1:8118/
HTTP_PROXY=$http_proxy
export http_proxy HTTP_PROXY
```

There is a nice shell script which sets a few aliases available at shellscripts.org: Shell Aliases and functions for tor (http://shellscripts.org/project/toraliases) .

Browsers

Set the web browsers http proxy to use privoxy (host: 127.0.0.1 port: 8118)

- Under Firefox, go to the ***Edit menu*** -> ***Preferences*** -> ***General*** -> ***Connection*** (also see an extension for Firefox below)
- Under Opera, go to the ***Tools menu*** -> ***Preferences*** -> ***Advanced*** -> ***Network*** -> ***Proxy Servers***
- Under Konqueror, go to the Settings menu -> ***Configure Konqueror.*** Scroll down to ***Proxy.*** Click ***Manually*** specify the proxy settings and then click ***Setup***

Make sure you specify that your browser must use Tor for *all* protocols. Tor *does not work for ftp*, but if you do not *ask your browser to use Tor for ftp anyway* then a website which loads an image using ftp will still log your location. (How do I use my browser for ftp with Tor? (http://wiki.noreply.org/noreply/TheOnionRouter/TorFAQ#FtpProxy))

IRC

Irssi

Use the "torify" wrapper to start irssi:

```
torify irssi
```

Gaim

- Go to the Tools menu, select Accounts
- Select the IM protocol you want to anonymize
- Click Modify
- Click Show more options
- Under Proxy Options select proxy type SOCKS 5
- Enter 127.0.0.1 for the host
- Enter 9050 for the port
- Leave user/pass field blank

X-Chat

Settings *->* ***Preferences*** *->* ***Network*** *->* ***Network Setup****.*

Then enter 127.0.0.1 and port 9050 in the "Proxy Server" settings.

Some tricks

You can telnet to Tor's control port and enter commands.

- "signal newnym" will make Tor switch to clean circuits, so new application requests don't share any circuits with old ones.
- "getinfo circuit-status" will show what circuits are open and used.

Vidalia and Tork are good GUI programs to view the status of Tor and control it using it's control-port.

Configuration tips

Using the same exit for persistant connections

Some websites will log you out if you re-visit (while loggined in using a cookie to identify you) from a different IP. Tor has a feature called *long lived ports*. You could add the following to torrc to make connections to given ports use the same circut for a long period of time:

```
LongLivedPorts 80,23,21,22,706,1863,5050,5190,5222,5223,6667,8300,8888
```

A good alternative to *LongLivedPorts* is to use *MapAddress* for given sites. It allows you to make sure *every* connection to a given site goes through the same connection. This is also a good option if you *need* given sites to be visited from *a given country*.

For example,

```
MapAddress www.nsa.gov www.nsa.gov.nadia.exit
```

will make all visits to *www.nsa.gov* always use the edit node *nadia*, which is located in the US. There are anonymity issues with this; if you're the only one using it then *www.nsa.gov* can at least figure out that *it's the same guy* who's visiting when connections are coming from that exit node.

Make Tor act faster

It is also possible to make Tor connections seem faster by setting *CircuitBuildTimeout*. Setting this number lower than the default (60 seconds) makes Tor give up and try other paths if it takes longer than the limit to build a circut. A circut which takes 50 seconds to build will be slower than a circut that takes 15 seconds to build. For example, you could set:

```
CircuitBuildTimeout 10
```

However, it must be mentioned that you will be using a whole lot more different servers if you allow circuts who take 50 seconds to build than if you set the limit to 10 seconds. There isn't much solid research on exactly how this impacts traffic analysis resistance, but you're - generally speaking - better off using a lot of slow servers than a few fast ones.

Tor

- What is Tor?
- **HOWTO use the Internet anonymously using Tor and Privoxy**
- HOWTO setup a Tor-server
- HOWTO publish anonymously on the Internet using Tor
- Hidden Tor services
- Bad Tor exit servers

Tor news

- 2007-02-08: Tor 0.1.2.8-alpha is out
- 2006-12-17: Tor 0.1.1.26 fixes HttpProxyAuthenticator privacy flaw
- 2006-11-05: Vidalia 0.0.9 is released
- 2006-10-24: TorK-0.10 released

Tor GUIs

- Vidalia
- Tork