English • Deutsch • Español • Français • Italiano • ??? • Polski • Português • ??????? • Svenska • ???(????)? • ???(??)? •

This guide assumes that you are running a version of DD-WRT that has Privoxy included into the DD-WRT firmware; it does not cover installing Privoxy from Optware, although many of the same instructions might still be useful if you do.

Contents

Privoxy basic configuration

Set up custom config for DD-WRT's built-in privoxy

In order to be able to customise Privoxy's configuration files, you will need to

- set up read/write storage i.e. JFFS2 or USB
- enable custom privoxy config (in GUI)
- paste contents of /tmp/privoxy.conf into custom config window in GUI

- turn on enable-edit-actions
- adjust the config to use /jffs/privoxy as confdir
- copy any relevant files from old confdir to new confdir
- now visit config.privoxy.org in your browser and you'll find "Edit" buttons next to the actions files

Add an exception for a site blocked in error

- edit user.actions
- scroll down until you find a ruleset that starts with:

-block, -crunch-incoming-cookies, -crunch-outgoing-cookies, -fast-redirects, -filter, -hide-referrer, -prevent-compression

- add a suitable URL pattern to that section and voila, you're unblocked.

Chaining Privoxy to other software / services

Privoxy supports **selectively** forwarding web traffic over HTTP or SOCKS proxy, which means you can use it to route some or all of your web traffic over Tor, and/or to a proxy server of your choice. This has a variety of possible uses, including

- caching HTTP GET requests to minimise bandwidth use
- circumventing the blocking of e.g. torrent websites by ISPs or governments
- providing network-wide access to Tor hidden services
- routing HTTP connections to location-sensitive services through a preferred country or jurisdiction

HTTP caching proxy

Privoxy can be chained with any of squid, polipo, tinyproxy, etc. for the purpose of local HTTP caching, although how much use this is to the average user on a fast, unmetered broadband connection is rather questionable, particularly if the cache is running on an embedded device such as a router. It might save some bandwidth if you have a metered connection, or an environment with a lot of users. If you want to do this, it works as described in the Privoxy manual and nothing is particularly note-worthy about doing this on DD-WRT. If you have a use for it, it works.

When using a local caching proxy, it is logical to assume that you'll be using it for all sites and therefore configuring it statically in the Privoxy config file. The same applies if you are forced to use a proxy in order to gain access to the outside world. In all other situations (and for the rest of this guide), we will assume that the purpose of chaining some other proxy onto a filtering proxy such as Privoxy is to use it selectively, with such URLs as we may specify to Privoxy to forward.

Remote HTTP proxy

Using a remote HTTP proxy will route your traffic via an alternative connection. Free proxies are generally worth exactly what you pay for them, unfortunately, so whilst this can substitute for a VPN if you have a suitable proxy available, chances are this won't be useful unless you run a proxy server of your own, or have access to a reliable one.

Immunicity

See https://immunicity.org/ for a service-based approach; it provides a public HTTP proxy that permits access only to a predetermined list of blocked sites. The advantage is that we can use it to unblock torrent sites (or porn, if that's your thing) without needing an external proxy, a working copy of Tor, or whatever.

To configure Privoxy to work with it, we'll need to generate a Privoxy actions file from Immunicity's PAC file.

```
#!/bin/sh
echo "# Immunicity forwarding ruleset created on `date`"
wget -q -O- http://clientconfig.immunicity.org/pacs/all.pac > /tmp/immunicity.pac
PROXY=`cat /tmp/immunicity.pac | grep "var proxyserver" | cut -d \' -f 2`
echo "{+forward-override{forward $PROXY}}"
cat /tmp/immunicity.pac | egrep \".*\"\,?$ | cut -d \" -f 2 | sed -e s/www.//g | sed -e s/\*.//g
```

Pipe that to /jffs/privoxy/immunicity.action and add an actionsfile line for it in the config file, after user.action.

Edit: whilst I was successful in configuring Privoxy to route traffic to Immunicity, their proxy at gateway-eu2 churned for about five minutes before throwing a "connection timed out" error on several different sites. Doesn't seem usable at present, and I can see it becoming a victim of its own success when it does. Worth revisiting to check this wasn't just bad timing, but I'll stick to tor.

Tor SOCKS5 proxy

Prerequisites

- Privoxy running on DD-WRT with custom configuration as above
- Tor either running on DD-WRT via Optware or on another machine
    - with an accessible SOCKS port
    - if you want network-wide access to hidden services, with accessible DNS and Trans ports also (see below)

Transparent proxying and Tor - a warning

Warning: transparently routing your HTTP connections through Tor provides obfuscation but not full anonymity. Your connections to TCP port 80 will be proxied via the Tor network, where so configured, but any other connections that your systems might make will not. For many users this is adequate, e.g. if you're just using Tor to get around website blocking, but it is not as secure as the dedicated Tor bundle. If that is a serious concern to you, read the Tor documentation on the subject (and don't rely on transparent proxying).

With that said, there are (at least) two useful things you can do with Tor chained to Privoxy:

Unblock torrent sites and the like

We assume that you have Tor set up on your router (or another machine), and are exposing a SOCKS proxy on 127.0.0.1:9050. You'll want to configure a Privoxy rule in user.action for selective forwarding to Tor. From the config.privoxy.org interface, edit user.actions, scroll to the end of the file, and add a new section below. Select "enable" for the forward-override match, and in the textbox, enter "forward-socks5 127.0.0.1:9050". Then add the sites that you want to route via Tor.

If you'd rather work at command line, edit /jffs/privoxy/user.action and add a section at the end that reads

```
+forward-override{forward-socks5 127.0.0.1:9050 .}
torrentz.eu
thepiratebay.se
```

and so on. This will circumvent basic web filtering by passing your connections to the listed sites off to Tor.

Generate a list of torrent sites to unblock

Which sites you should unblock is variable, country-specific, and not guaranteed, but there are published lists of sites that typically require a proxy. The following script uses the Immunicity URL list and generates output suitable for a Privoxy actions file.

```
#!/bin/sh
echo "# Immunicity forwarding ruleset created on `date`"
wget -q -O- http://clientconfig.immunicity.org/pacs/all.pac > /tmp/immunicity.pac
echo "{+forward-override{forward-socks5 127.0.0.1:9050 .}}"
cat /tmp/immunicity.pac | egrep \".*\"\,?$ | cut -d \" -f 2 | sed -e s/www.//g | sed -e s/\*.//g
```

Pipe that to /jffs/privoxy/tor.action and include the relevant actionsfile line in your Privoxy config after the user.actions file. You are not recommended to automate generating this file unless you want it to break, just to use it as a starting point.

Take note that, whilst tor can enable access to sites blocked by ISPs and governments, the use of it for bittorrent data traffic, streaming, p2p filesharing etc. is discouraged, both ideologically and technically speaking. Accessing your blocked .torrent files is fine, but please do not configure your p2p clients (including bittorrent) to use tor. See e.g. http:/http://wiki.dd-wrt.com/wiki.vuze.comhttp://wiki.dd-wrt.com/w/Tor_HowTo for some guidelines.

Transparent access to .onion sites

See https://grepular.com/Transparent_Access_to_Tor_Hidden_Services for the general idea.

Configure (fake) DNS for Tor

As per the article, you'll want Tor operating a DNSPort on e.g. localhost port 5300, and you'll want your local DNS server, presumably dnsmasq, resolving the queries for .onion by handing them off to Tor's DNS, which will return an IPv4 address in 10.x.x.x. Proceed exactly as per the article, but to configure dnsmasq, add

```
server=/onion/127.0.0.1#5300
```

into the dnsmasq options field of the DD-WRT GUI. You will also need to disable NoDNSRebind.

Configure transparent forwarding to Tor

We used a SOCKS connection to Tor for regular domains (where we get a valid IP address), but for hidden services, we need to intercept the browser's attempt to connect to our 10.x.x.x address and feed that connection back into Tor via a transparent proxy port. Tor will resolve our IPv4 address back to the hidden service we requested and route our connection there.

```
iptables -t nat -A OUTPUT -p tcp -d 10.192.0.0/11 -j REDIRECT --to-port 9040
```

should do the trick for connections coming through Privoxy. You should now be able to enter an .onion URL into your browser, have it resolve to a 10.x.x.x address and connect through Tor to the hidden service. You might also want to add a

```
iptables -t nat -A PREROUTING -p tcp -d 10.192.0.0/11 -j REDIRECT --to-port 9040
```

which will take care of non-Privoxy connections to hidden services, useful if you want to connect to e.g. IRC from something other than the router. You'll probably want to make these rules permanent by adding them to your firewall startup commands in DD-WRT.

Geographical restrictions on Tor exit nodes

It is possible to restrict the exit nodes that Tor will use, such that your client circuits will always finish in e.g. the UK or the USA. This is useful if you need to access something where geographical restrictions apply.

Take note that whilst you can restrict your Tor exit nodes to a specific country, you should generally not do so for the purpose of accessing streaming media services, i.e. Netflix, YouTube or other gratuitously high-bandwidth applications. See http://tor.stackexchange.com/questions/733/can-i-exit-from-a-specific-country-or-node for relevant considerations, as well as for how to make the relevant settings if appropriate.

If what you want to do is precisely to route connections through a specific country for multimedia or file-sharing applications, e.g. BBC iPlayer or Netflix, then please consider buying a VPN service or using a virtual server. This is not normally an appropriate usage of Tor, and you won't get great performance either.

OpenSSH / AutoSSH SOCKS5 proxy

As far as Privoxy is concerned, SOCKS5 is just a protocol, and we don't care how it is provided. The

OpenSSH client is capable of creating a SOCKS5 proxy on demand with a one-line command, provided that you have a login on a suitable machine to proxy your connections through. Here, a 'suitable machine' might mean any of:

- a virtual private server (VPS) in a datacentre of your choice (VPS hosting options here)
- a physical server to which you have access
- a remote router to which you have access
- a phone or mobile device that supports an SSH server
- a Unix/Linux shell account (subject to forwarding being allowed, which is relatively uncommon)

Prerequisites

- Privoxy running on DD-WRT with custom configuration as above
- Suitable remote machine
    - ♦ with SSH access
    - ♦ with TCP forwarding
    - ♦ with access to the hosts that you intend to connect to
    - ♦ with permission (I strongly suggest)
- OpenSSH client installed on DD-WRT via Optware (not the dropbear SSH client, it doesn't support SOCKS5)
    - ♦ passwordless, key-based SSH login to your target machine