

Now available as Onion Service through the Tor Network

Date: 2022-03-13
modified: 2022-03-13
tags: Tor, Onion Service, FreeBSD, nginx, Pelican
description: Access The Wombelix Post anonymously as Hidden Service
category: Misc
slug: now_available_as_onion_service_through_the_tor_network
Author: Dominik Wombacher
lang: en
transid: now_available_as_onion_service_through_the_tor_network
Status: published

I'm happy to announce that this site is also published as Onion Service from now on, which means a focus on privacy, security, freedom and support for the Tor Project as well as a statement against censorship.

The new Tor URL: <http://2xwpdwnzmag3ewobwsdewpor4gmca4d5gltviol3u6upihb6m6m6xaad.onion>

Also it was fun to setup ;) To ensure the right URLs are used, I decided to publish two versions, which was quite a simple task by adjusting a few lines in my [Pelican](#) configs.

The Tor Service is running on the same FreeBSD Jail as my (static) site and nginx, let me share some technical details.

Installing Tor is straight forward, just run `pkg install tor` and `sysrc tor_enable="YES"`.

Two lines in `/usr/local/etc/tor/torrc` are enough to enable a new Onion Hidden Service:

```
HiddenServiceDir /var/db/tor/keys/<website>/
HiddenServicePort 80 unix:/var/run/tor-<website>.sock
```

For nginx I adjusted the existing https config to publish the [Onion-Location](#) (Archive: [\[1\]](#), [\[2\]](#)) header, which will advertise the `.onion` URL of this Site to visitors that are using the Tor Browser.

The Onion URL can be found in `/var/db/tor/keys/<website>/hostname`.

```
server {
    listen 443 ssl http2;
    # Tor unrelated config omitted
    add_header Onion-Location http://<onion_url>$request_uri;
}
```

As recommend in the [Tor Setup Guide](#) (Archive: [\[1\]](#), [\[2\]](#)) I added an additional server section and use a unix socket to listen for Tor requests.

```
server {
    listen unix:/var/run/<website>.sock;
    # Tor unrelated config omitted
    server_name <onion_url>;
    root <path_to_web_document_root>;
}
```

From a Pelican perspective, I created a second `publishconf` to set the `SITEURL` to my `<onion_url>` and adjusted the `Makefile` a little to upload the `regular` and the `tor` version at once.

Following the additions on top of the standard Makefile when installing Pelican.

```
PUBLISHCONF_TOR=$(BASEDIR)/publishconf_tor.py
SSH_TARGET_DIR_TOR=<path_to_web_document_root>

# Tor unrelated config omitted

rsync_upload_tor: publish_tor
    rsync -e "ssh -p $(SSH_PORT)" -P -rvzc --include tags --cvs-exclude --delete "$(OUTPUTDIR)"/ *$(SSH_USER)@$(SSH_HOST):$(SSH_TARGET_DIR_TOR)"
rsync_upload_all: rsync_upload rsync_upload_tor
```

Last step was to start the tor service `service tor start`, apply the new nginx config `service nginx reload` and to publish the site `make rsync_upload_all`.