


| |
|---------------------------------|
| Documentation Overview |
| Install Tor Browser |
| Tor on Android |
| Other Tor software |
| ▼ Expert guides |
| The Tor Relay Guide |
| Installing Tor on Debian/Ubuntu |
| Installing Tor Source |
| OSX |
| Configuring an Onion Service |
| Understanding bridges |
| Verify package signatures |
| ► Manuals |
| Tor Wiki |
| General FAQ |
| Abuse FAQ |
| Trademark FAQ |
| Tor Legal FAQ |

 **Tor Tip**

Tor is written for and supported by people like you. [Donate today!](#)

Configuring Onion Services for Tor

Tor allows clients and relays to offer onion services. That is, you can offer a web server, SSH server, etc., without revealing your IP address to its users. In fact, because you don't use any public address, you can run an onion service from behind your firewall.

If you have Tor installed, you can see onion services in action by visiting this [sample site](#).

This page describes the steps for setting up your own onion service website. For the technical details of how the onion service protocol works, see our [onion service protocol](#) page.

Step Zero: Get Tor working

Before you start, you need to make sure:

- a. Tor is up and running,
- b. You actually set it up correctly.

Windows users should follow the [Windows howto](#), OS X users should follow the [OS X howto](#), and Linux/BSD/Unix users should follow the [Unix howto](#).

Step One: Install a web server locally

First, you need to set up a web server locally. Setting up a web server can be complex. We're not going to cover how to set up a web server here. If you get stuck or want to do more, find a friend who can help you. We recommend you install a new separate web server for your onion service, since even if you already have one installed, you may be using it (or want to use it later) for a normal website.

You need to configure your web server so it doesn't give away any information about you, your computer, or your location. Be sure to bind the web server only to localhost (if people could get to it directly, they could confirm that your computer is the one offering the onion service). Be sure that its error messages don't list your hostname or other hints. Consider putting the web server in a sandbox or VM to limit the damage from code vulnerabilities.

Once your web server is set up, make sure it works: open your browser and go to <http://localhost:8080/>, where 8080 is the webserver port you chose during setup (you can choose any port, 8080 is just an example). Then try putting a file in the main html directory, and make sure it shows up when you access the site.

Step Two: Configure your onion service

Next, you need to configure your onion service to point to your local web server.

First, open your torrc file in your favorite text editor. (See [the torrc FAQ entry](#) to learn what this means.) Go to the middle section and look for the line

```
##### This section is just for location-hidden services ###
```

This section of the file consists of groups of lines, each representing one onion service. Right now they are all commented out (the lines start with #), so onion services are disabled. Each group of lines consists of one *HiddenServiceDir* line, and one or more *HiddenServicePort* lines:

- *HiddenServiceDir* is a directory where Tor will store information about that onion service. In particular, Tor will create a file here named *hostname* which will tell you the onion URL. You don't need to add any files to this directory. Make sure this is not the same directory as the hidserv directory you created when setting up thttpd, as your HiddenServiceDir contains secret information!
- *HiddenServicePort* lets you specify a virtual port (that is, what port people accessing the onion service will think they're using) and an IP address and port for redirecting connections to this virtual port.

Add the following lines to your torrc:

```
HiddenServiceDir /Library/Tor/var/lib/tor/hidden_service/  
HiddenServicePort 80 127.0.0.1:8080
```

You're going to want to change the *HiddenServiceDir* line, so it points to an actual directory that is readable/writeable by the user that will be running Tor. The above line should work if you're using the OS X Tor package. On Unix, try `"/home/username/hidden_service/"` and fill in your own username in place of "username". On Windows you might pick:

```
HiddenServiceDir C:\Users\username\Documents\tor\hidden_service
HiddenServicePort 80 127.0.0.1:8080
```

Note that since 0.2.6, both *SocksPort* and *HiddenServicePort* support Unix sockets. This means that you can point the *HiddenServicePort* to a Unix socket:

```
HiddenServiceDir /Library/Tor/var/lib/tor/hidden_service/
HiddenServicePort 80 unix:/path/to/socket
```

Now save the torrc and restart your tor.

If Tor starts up again, great. Otherwise, something is wrong. First look at your logfiles for hints. It will print some warnings or error messages. That should give you an idea what went wrong. Typically there are typos in the torrc or wrong directory permissions (See [the logging FAQ entry](#) if you don't know how to enable or find your log file.)

When Tor starts, it will automatically create the *HiddenServiceDir* that you specified (if necessary), and it will create two files there.

private_key

First, Tor will generate a new public/private keypair for your onion service. It is written into a file called "private_key". Don't share this key with others -- if you do they will be able to impersonate your onion service.

hostname

The other file Tor will create is called "hostname". This contains a short summary of your public key -- it will look something like `duskgyt1dkxiuqc6.onion`. This is the public name for your service, and you can tell it to people, publish it on websites, put it on business cards, etc.

If Tor runs as a different user than you, for example on OS X, Debian, or Red Hat, then you may need to become root to be able to view these files.

Now that you've restarted Tor, it is busy picking introduction points in the Tor network, and generating an *onion service descriptor*. This is a signed list of introduction points along with the service's full public key. It anonymously publishes this descriptor to the directory servers, and other people anonymously fetch it from the directory servers when they're trying to access your service.

Try it now: paste the contents of the hostname file into your web browser. If it works, you'll get the html page you set up in step one. If it doesn't work, look in your logs for some hints, and keep playing with it until it works.

Step Three: More advanced tips

If you plan to keep your service available for a long time, you might want to make a backup copy of the *private_key* file somewhere.

If you want to forward multiple virtual ports for a single onion service, just add more *HiddenServicePort* lines. If you want to run multiple onion services from the same Tor client, just add another *HiddenServiceDir* line. All the following *HiddenServicePort* lines refer to this *HiddenServiceDir* line, until you add another *HiddenServiceDir* line:

```
HiddenServiceDir /usr/local/etc/tor/hidden_service/
HiddenServicePort 80 127.0.0.1:8080

HiddenServiceDir /usr/local/etc/tor/other_hidden_service/
HiddenServicePort 6667 127.0.0.1:6667
HiddenServicePort 22 127.0.0.1:22
```

Onion services operators need to practice proper operational security and system administration to maintain security. For some security suggestions please make sure you read over Riseup's "[Tor Hidden \(Onion\) Services Best Practices](#)" [document](#). Also, here are some more anonymity issues you should keep in mind:

- As mentioned above, be careful of letting your web server reveal identifying information about you, your computer, or your location. For example, readers can probably determine whether it's tthttpd or Apache, and learn something about your operating system.
- If your computer isn't online all the time, your onion service won't be either. This leaks information to an observant adversary.
- It is generally a better idea to host onion services on a Tor client rather than a Tor relay, since relay uptime and other properties are publicly visible.
- The longer an onion service is online, the higher the risk that its location is discovered. The most prominent attacks are building a profile of the onion service's availability and matching induced traffic patterns.

Another common issue is whether to use HTTPS on your relay or not. Have a look at this [post](#) on the Tor Blog to learn more about these issues.

Finally, feel free to use the [\[tor-onions\] mailing list](#) to discuss the secure administration and operation of Tor onion services.



Trademark, copyright notices, and rules for use by third parties can be found [in our FAQ](#).

About Tor

- [What Tor Does](#)
- [Users of Tor](#)
- [Core Tor People](#)
- [Sponsors](#)
- [Contact Us](#)

Get Involved

- [Donate](#)
- [Mailing Lists](#)
- [Onion Services](#)
- [Translations](#)

Documentation

- [Manuals](#)
- [Installation Guides](#)
- [Tor Wiki](#)
- [General Tor FAQ](#)

