Tor: a sociological problem. Or umbrella: one can be stopped but not two million.

Camille Akmut
June 19, 2019

Abstract
Computer science and technology are both a scientific, technical and technological as well a sociological, societal and political problem.

Introduction: computer science and technology, a sociological problem.

“The systems both human and computer” wrote Berners-Lee, while Adele Goldberg addressed explicitly the “sociological” problems of her own (that are those of the systems of programming languages, and their various communities).

As a project reaches some degree of maturity and popularity, its problems become inevitably both technical and sociological (and political).

So in the case of the Web, so also with Smalltalk. (To not speak of Google or Facebook, or Haskell.)

—

What may be called the “Hong Kong law” or problem (after a city where both technological and human problems are exacerbated).

—

In the following we present a small program entitled umbrella whose purpose it is to formalize, if not fix one of these problems.

For the Tor Project, this problem takes on the following form:

How to engage people so they will contribute to the network? The solution, at first, seems evident, and should be easy, as it is in their own interests to do so; and yet the technical problems cover sociological ones.

This project faces real difficulties in its communication with the wider outside world, partly self-inflicted, but partly also imposed by others; and in trying to fix them perhaps “falls from one trap into the next” – as we have adventurously proposed.
umbrella

"an object (...) to protect [one]self from the rain or hot sun"

https://en.wikipedia.org/wiki/2014_Hong_Kong_protests

Tor relay/bridge easy, automated configuration, deployment; and a documentation.

Install  + set up and build umbrella

Prerequisite: Docker.

1. Setting up umbrella

Folder structure:

- umbrella-package
  -- Dockerfile
  -- torrc

where 'umbrella-package' is a directory and 'Dockerfile' and 'torc' are both files within it.

2. Building umbrella (i.e. building its "image" in Docker terminology)

```
sudo docker build -t "umbrella" umbrella-package
```

where `-t "<name>"` is the Docker syntax to name an image (or tag, hence `-t`), here 'umbrella'.

Use  /\ open and close \ umbrella

- Starting umbrella (i.e. creating a "container" or small system based on its image)

```
sudo docker run -dit umbrella
```

where `-dit` is necessary as otherwise the container would stop ("exit") immediately; 'umbrella' being the newly created, eponymous image (on which the container is based). `-p <port>` (publish) may be needed additionally.

- Stopping umbrella

```
sudo docker stop <container ID>
```

- Accessing a shell inside an umbrella container:

```
sudo docker exec -it <container ID> sh
```
Uninstall => throw away umbrella

```bash
sudo docker rmi umbrella
```

Make sure no container using the `umbrella` image still exists; otherwise remove it/them:

```bash
sudo docker ps -a
sudo docker container list
sudo docker rm <container ID>
```

Components [ ] umbrella package contents

**Dockerfile**

```bash
FROM debian:9-slim
RUN apt-get update
RUN apt-get install -y tor
COPY torrc /etc/tor/
EXPOSE 443 #9001
RUN adduser user
USER user
CMD ["tor"]
```

torrc

```bash
ORPort 443
Exitpolicy reject *:*
BridgeRelay 1  # only add this line if you want to be a bridge
```

torrc    // umbrella manual #1

torrc is the file that holds the configuration information for the Tor program.

```bash
ORPort
```

has for background "When Tor is configured to be a relay or bridge, it requires (...) at least a single TCP port for its Onion Router Port (ORPort)." [3]

```bash
Exitpolicy reject *:*`
```
"accept/reject *, (...) applies to both IPv4 and IPv6" [2]

ServerTransportPlugin obfs3,obfs4 exec /usr/bin/obfs4proxy


More information:
https://trac.torproject.org/projects/tor/wiki/TorRelayGuide
https://trac.torproject.org/projects/tor/wiki/doc/TorFAQ

All torrc options. See also man tor ("THE CONFIGURATION FILE FORMAT" and "GENERAL OPTIONS")


**Dockerfile**  // umbrella manual #2

A **Dockerfile** is a file that holds configuration information for a Docker image/container.

FROM debian:9-slim

installs Debian 9 (‘Stretch’) based on the image downloaded from the official Debian repository (in the less voluminous "slim" variant).

More information: https://hub.docker.com/_/debian/

RUN apt-get update

runs the command `apt-get` -- ~ Debian's package manager -- with the `update` argument: this updates the information available for all packages, and has for principal utility to make sure one gets and can install the latest version of each. (This comes in handy in the next step, where the `tor` package is installed.)

RUN apt-get install -y tor

installs the Tor program. The `-y` flag is necessary as otherwise building of the `umbrella` image would be interrupted by a dialog asking to confirm the installation of the program before proceeding; resulting in an incomplete (failed) image.

COPY torrc /etc/tor/
copies the torrc file into the appropriate directory.

```
EXPOSE 443 #9001
```

exposes port 443 (corresponding to the ORPort). 9001 is commented out but could be picked instead or most any other port (the ORPort entry in the torrc file would need to be changed accordingly as well).

```
RUN adduser user
USER user
```

runs adduser -- the tool used to create new users in Debian -- with the argument user, the name of the (new) user.

In the next step a switch to this (non-root) user, user, occurs. Why?

If this step were omitted, which is to say if one proceeded to start the Tor program as root user, the following message would warn, but not prevent execution:

"[warn] You are running Tor as root. You don't need to, and you probably shouldn't."

More information: https://trac.torproject.org/projects/tor/wiki/doc/TorFAQ where is stated "running Tor as root is not recommended (in case there are unknown exploitable bugs)."

```
CMD ["tor"]
```

starts the Tor program.

**Monitoring oo rain or hot sun?**

Nyx (formerly arm). More information: https://nyx.torproject.org/

```
sudo apt-get install nyx
```

A ControlPort is needed, as an error message will indicate if trying to run nyx without having set up one.

In torrc add e.g.:

```
ControlPort 9051
```

**FAQ, Troubleshooting // umbrella manual #3**

- Before anything else, make sure that Docker has been installed correctly / works

```
sudo systemctl status docker
```

Alternatively, or additionally:

```
sudo docker run hello-world
```
• **Cannot build the Dockerfile**

Contrary to intuition, a Dockerfile is not built like so:

```bash
# sudo docker build Dockerfile
unable to prepare context: context must be a directory: (...) 
```

As the error message makes somewhat clear, docker build is meant to be executed on a directory (the one containing Dockerfile), not on the Dockerfile itself. Hence e.g.:

(from just outside of the folder)

```bash
sudo docker build <folder containing the Dockerfile>
```

or (from anywhere in the file system)

```bash
sudo docker build '<path to the folder>'
```

• **"[warn] Could not bind to 127.0.0.1:9050: Address already in use. Is Tor already running?"**

```bash
sudo systemctl stop tor
```

then start tor again.

```bash
tor
```

• **"[warn] Could not bind to 0.0.0.0:443: Permission denied"**

Context: in torrc ORPort is set to 443. (There is nothing wrong with this, and in fact seems to be the recommended ORPort.)

This is due to running tor as a non-root user -- as it should be -- while trying to bind to a (privileged) port under 1024 (for which in turn root rights are required).

More information: [https://trac.torproject.org/projects/tor/wiki/doc/TorFAQ](https://trac.torproject.org/projects/tor/wiki/doc/TorFAQ) ("How can I make my relay accessible to people stuck behind restrictive firewalls?")

• **Tor, (consumer) routers, port forwarding (UPnP, NAT-PMP, tor-fw-helper, etc.)**

[https://github.com/Yawning/tor-fw-helper](https://github.com/Yawning/tor-fw-helper) (tor-fw-helper by Tor Project contributor Yawning Angel)

Applebaum, Jacob. 2012. "Tor and NAT devices: increasing bridge & relay reachability".

• **Software firewalls**

ufw (user-friendly front-end for iptables)

```bash
sudo ufw allow 9001
```

```bash
sudo ufw status verbose
```
A Tor bridge is not public, how do computers know to connect to it?

"Your bridge relay will automatically publish its address to the bridge authority, which will give it out via https or email" (BillRM in https://able2know.org/topic/168013-1)


**Why not just use docker build umbrella?**

This works too; with the following caveat

```bash
# sudo docker image list
REPOSITORY   TAG     IMAGE ID
<none>       <none>  ............
```

as opposed to:

```bash
# sudo docker build -t "umbrella" umbrella-package
(...)
Successfully tagged umbrella:latest

# sudo docker image list
REPOSITORY   TAG     IMAGE ID
umbrella     latest  ............
```

**Other Docker commands.**

docker image list and docker image ls do the same things for images, as do for containers container list and container ls. (Note docker ps -a is not equivalent to the latter!)

**errata (goofs and gaffs) ! suncream, water ...**

ExitPolicy reject6 ":*, reject ":*"

https://tor-relay.co/

Comment: "accept/reject ":*, (...) applies to both IPv4 and IPv6" (tor manual)

ControlSocket 0

https://hub.docker.com/r/chriswayg/tor-server ("A complete, efficient and secure Tor relay server Docker image", "on Debian")

Comment: "0 disables ControlSocket. (Unix and Unix-like systems only.) (Default: 0)" (tor manual)

# Launch tor in foreground
CMD ["tor", "-f"]

https://github.com/Themimitoof/docker-tor-relay

Comment: "-f FILE" (man tor)