

## Assignment 1: Insidious Proxy

*Due date: Monday, April 6, 2020 at 22:00*

*This is an individual assignment. You may discuss it with others, but your code and documentation must be written on your own.*

Write an HTTP proxy server called `insidious_proxy` that surreptitiously changes some of the content it serves. Because of performance and functionality reasons the server should be multithreaded. In particular, the proxy should flip images vertically or rotate them by 90 degrees.

It must be possible to configure a web browser's proxy settings to point to `insidious_proxy`; it should also be possible to accept raw HTTP/HTTPS requests from applications like netcat (as seen on class) or telnet. All HTTP methods<sup>1</sup> must be supported.

Note that HTTPS sites should also be handled by the proxy, but in this case it won't be possible to flip the images (why?). As modern browsers try to force all requests to be HTTP, WhyNoHTTPS<sup>2</sup> may prove useful to find some sites that still answer HTTP requests (e.g. <http://imageshack.us/>)

You can always refer to the HTTP/1.1 RFC<sup>3</sup> for more information about what is *expected* from a proxy, what it *must/should, must not/should not*, do.

The proxy must accept the following command-line arguments:

- `PORT_NUMBER`: although 8080 (an alternative to HTTP port 80) is the well-known port for HTTP proxies, this should be configurable.
- `IMAGE_ORIENTATION`: accepted values for the argument should be *flip*, *clockwise*, *counterclockwise*, or *random* (any of the previous choices at random). This should be the modification applied to the images of the visited sites.

### Image Processing

You may use any library you want to modify the images<sup>4</sup>, although ImageMagick is recommended, as one of the most known image manipulation libraries available. It has wrappers for C (MagickWand<sup>5</sup>), C++ (Magick++<sup>6</sup>), and Java (JMagick<sup>7</sup>).

### Submission Instructions

You may write your solution in C, C++, or Java. Package all the source files plus a README file in a single zip or tar archive. Make sure that you include all the necessary components to build and run your solution on a standard installation of a C, C++, or Java environment. In particular, make sure your solution works with the most basic command-line tools, outside of any integrated development environment.

Add comments to your code to explain sections of the code that might not be clear. Use the README file to add general comments to properly acknowledge any and all external sources of information you may have used,

---

<sup>1</sup>GET, HEAD, POST, PUT, DELETE, CONNECT

<sup>2</sup><https://whynohttps.com/>

<sup>3</sup>In particular, RFC7231

<sup>4</sup>It would be even possible to implement it oneself, although it is not the objective of this assignment

<sup>5</sup><https://imagemagick.org/script/magick-wand.php>

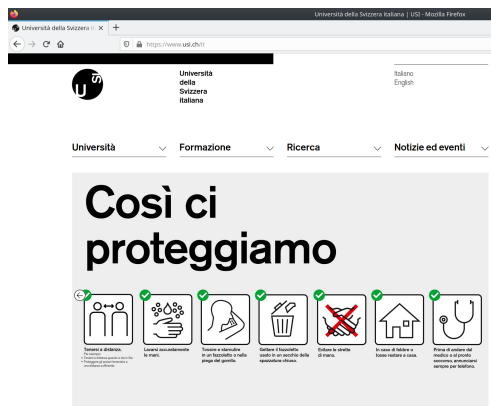
<sup>6</sup><https://imagemagick.org/script/magick++.php>

<sup>7</sup><https://github.com/techblue/jmagick>

including code, suggestions, and comments from other students. If your implementation has limitations and errors you are aware of (and were unable to fix), then list those as well in the README file.

Submit your solution package through the iCorsi system.

## Example



(a) Without proxy



(b) With proxy