Net Ninny Proxy Server

by Alexander Jonsson (alejo720@student.liu.se)
and Jonas Malmlöf (jonma993@student.liu.se)|

How to use:

1. Start the proxy-server by typing (in the terminal):

   Either

   ```
   ruby proxy.rb [port]
   ```

   or

   ```
   ./proxy.rb [port]
   ```

   where [port] is greater than 1024

2. Go to your browser's network settings and change its proxy connection to either “localhost” or “127.0.0.1”, using the same port-number as you chose above.

Alternatively go to your operating systems Network Preferences to change it there in the same way, but for all browser.

Features:

There is no need to compile the program.

This Net Ninny will block any websites that mention "Spongebob", "Paris Hilton", "Britney Spears" or "Norrköping".

Supports both HTTP/1.0 and HTTP/1.1.

Handles simple HTTP GET interactions between client and server.

Detects inappropriate content bytes within a Web page before it is returned to the user, and redirects to an error page.

Imposes no limit on the size of the transferred HTTP data.

Compatible with all major webbrowsers (Chrome, Firefox, Internet Explorer etc.).

Works with major operating systems (Linux, OS-X, Windows)

Allows the user to select the proxy port.

Is smart in selection of what HTTP content should be searched for forbidden keywords and when (the search is performed on text-files).

Limitations:

This Net Ninny will not block any websites using HTTPS-protocol.

No caching is performed by the Net Ninny.

Does not support POST HTTP-method.

References:

1: Brian “Beej Jorgensen” Hall "Beej's Guide to Network Programming Using Internet Sockets" 2016

Details in how the proxy server does its job:

2 Handles simple HTTP GET interactions between client and server:
The server accepts a client by using the "server.accept" function. Then the server fetches the client's HTTP-request by using "client.gets". Various information from the request is then stored in variables. We then use this information to create a new HTTP request we send to the web server:

```
        "proxy_client.print("#{http_method} #{path} HTTP/1.1\r\nHost:
#{hostname}\r\nConnection: close\r\n\r\n")"
```

("proxy_client" is the socket connected to the web server).

3 Blocks requests for undesirable URLs.
The code first checks if a bad word was found in the URL. If that's the case the "redirect_client"-function is called which redirects a specified client to a specified URL (in this case the URL is hardcoded to be the net nanny error page). Example of what it can look like:

```
        "if ( has_bad_word(body) )
           redirect_client(client,
              "http://www.ida.liu.se/~TDTS04/labs/2011/ass2/error2.html")".
```

Here we check if the body has a bad word and then calls the "redirect_client"-function with the specified client and URL as parameters.

8 Is smart in selection of what HTTP content should be searched for the forbidden keywords:
After fetching the header of the received package from the web server we use the following code to check if it's a text-file with Content-Length before searching for bad words:

```
      line = proxy_client.read(1024)
        if ( header )
          If ( line.include?("Content-Length:") )
            has_content_length = true
          end
          if ( line.include?("Content-Type: text") )
            textfile = true
          end
        end
        if ( textfile && has_content_length )
          if ( has_bad_word(line) )
```

Pages like www.google.se use Content-Type: text, but do not have a "Content-Length:" and thus will be loading in perpetuity if you try to load them without this double-check.

References:

1: Brian “Beej Jorgensen” Hall "Beej's Guide to Network Programming Using Internet Sockets" 2016