Proxy-Server

An HTTP proxy server implemented via python socket programming with caching, threading and blacklisting.

It is capable of servicing requests from clients through sockets, and allowing clients on the inside to access outside servers. If the server runs on port 20100 of the machine. The ports 20000-20099 are the IP addresses of systems inside system, and 20101-20200 are servers outside IIIT. The clients inside IIIT send HTTP requests to the proxy server. The proxy server sends those requests to the destination IP (the destination can be any server on the intranet or on your machine), and gets the response, which gets passed on to the initial requester. On receiving a request, the proxy creates a new thread. The new thread will service the request, while the parent continues to listen.

Caching

When the proxy server gets a request, it checks if the requested object is cached (i.e. server already has the request webpage or file), and if yes, it returns the object from the cache, without contacting the server.

If the object is not cached, the proxy retrieves the object from the server, returns it to you and caches a copy of this webpage for future requests if the Cache-Control header is set to must-revalidate. If the Cache-Control header is set to no-cache, then the proxy server does not caches the webpage.

In case of any further requests if the webpage is already cached then, the proxy utilize the “If Modified Since” header to check if any updates have been made, and if not, then serve the response from the cache, otherwise webpage or file is again retrieves from the server.

Code Structure

- main.py is the main proxy file.
- s3.py contains the code of backend server.
- blacklist.txt contains list of blacklisted servers.
- cache folder will be contains the files cached by proxy server.

- auth.txt contains allowed user credentials to access blacklisted sites
- client.py contains code to make local request to proxy servers which will allow only inside ports. Arguments for server port and user credentials are also supported.

Features

● Threaded Proxy server

● The proxy keeps count of the requests that are made. If a URL is requested more than 3 times in 5 minutes, the response from the server gets cached. In case of any further requests for the same, the proxy utilises the “If Modified Since” header to check if any updates have been made, and if not, then serve the response from the cache. The cache has a memory limit of 3 responses.

● The proxy supports blacklisting of certain outside domains. These addresses are stored in “proxy/blacklist.txt” in CIDR format. If the request wants a page that belongs to one of these, then, return an error page.

● Handles proxy authentication using Basic Access Authentication and appropriate headers to allow access to blacklisted sites as well. The authentication is username/password based, and is assumed to be stored on the proxy server.

Handles both GET and POST request.

How to run

Proxy

`python main.py` runs proxy server on port 20100.

Server

- Run server from s5.py file using `python s5.py`

Client

Client can request proxy through client.py by specifying server port as a necessary argument and authorization credentials if wanted to access a blacklisted file.

Also we can use the proxy server by using curl or browser.

For browser to use the proxy, you'll need to set the proxy by changing in your system's network settings..