A Web Proxy Server

Lucia Brown

My web proxy server consists of three Python files – one containing the proxy server (proxy.py), and two test scripts which provide meaningful outputs. One test script deals with HTTP requests, using example.com as its input (http_test.py). The other test script deals with HTTPS requests, using google.com as its input (https_test.py).

Proxy.py Overview:

handle_client_request; This is a large function where most of the work occurs. It handles incoming client requests, forwards them to the target server and relays the response back. It times out the client socket after 25 seconds. I chose 25 seconds as my initial code with a 2 second timeout presented issues when I attempted to implement the caching portion of the requirements.

read_full_request; This function reads the full HTTP request from the client socket. It reads in the data chunk by chunk using the recv functionality. It ceases reading when there is no more data being received, or once full HTTP headers have been read.

handle_http; This function is invoked by handle_client_request when the port numer is realised to not be 443, thus it is an unsecured connection. This function first searches the cache to see if the cache has expired for the provided host and port. If the cache has not yet expired, it can service the request using the cache. If the cache cannot be used to handle the request then fresh data is fetched.

handle_https; This function is invoked by handle_client_request when the port number is discovered to be equal to 443/there is a CONNECT method.

forward_data; This function is used to forward data between two sockets. Due to the cognitive complexity of the program, Python prompted me to split up larger functions into smaller ones, thus I had to create this function to reduce the load on my larger functions.

close_sockets; This function is used to safely close the sockets involved in the exchange.

url_is_blocked; This function is used during requests to validate if the requested URL is or is not blocked.

blocklist; This presents a menu to the user that allows them to add URLs to a blocklist, remove URLs from the blocklist, view the blocklist or exit.

extract_host_port_from_request; This function extracts the host and the port from the HTTP request. It is used by handle_client_request to ensure that the host and port are valid. And used by url_is_blocked to check that the URL is valid to be used.

http_test.py and https_test.pyOverview:

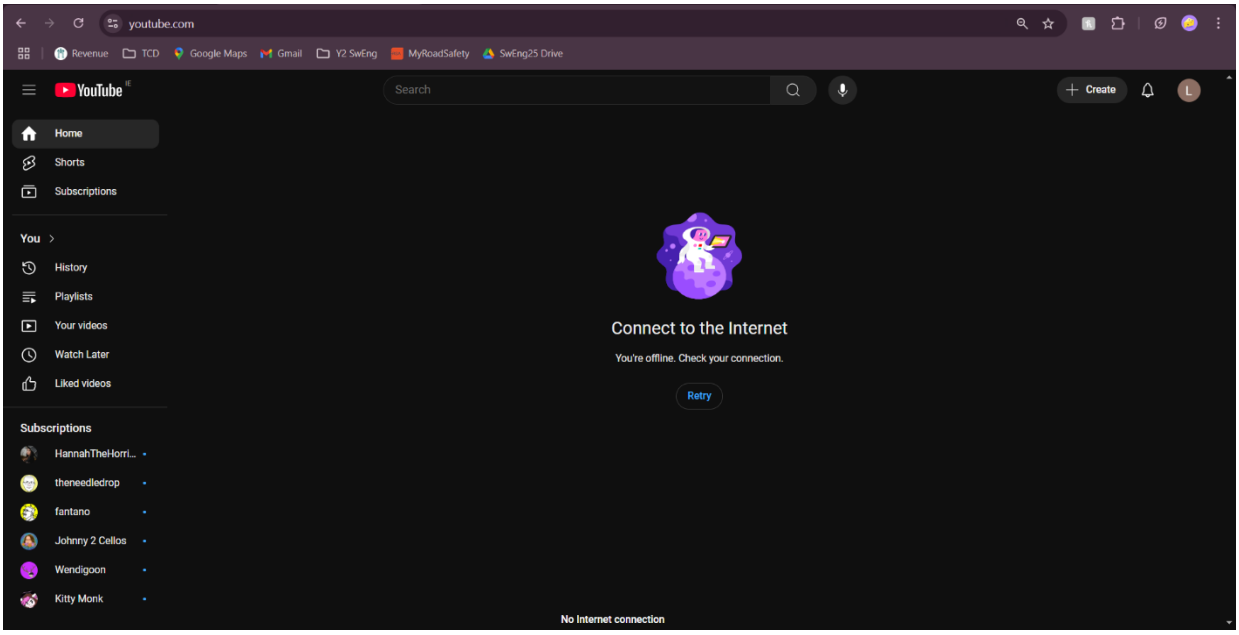
There are three tests in http_test.py and https_test.py respectively with the same test cases: the first test simply conducts one basic connection with the proxy server. It prints the HTTP response code, which should be 200, it prints the response header and the first 200 characters of the response body to ensure that everything is successful.

The second test ensures that the caching functionality works. This test works by making a request to the proxy. After a short propagation delay, a second identical request is made to the server and response1 is compared to response2. If response1 and response2 are identical, it can be inferred that a cache hit was likely. After a larger delay, past the cache expiry time, a third request is sent. Thus, response3 should take significantly longer to return than response2. Timing data is included in this test for comparison.

The third test ensures that the concurrency functionality works. This test simulates multiple users (3) requesting the same URL. This test verifies that all of the 'users' were able to access the server and that they all received the same responses from the server.

In-Browser Testing:

To test the efficiency of the URL blocking feature, with my proxy server running, I used the management console to block the www.youtube.com URL. This successfully prevented me from accessing the site:


youtube.com
Revenue
TCD
Google Maps
Gmail
Y2 SwEng
MyRoadSafety
SwEng25 Drive
YouTube
Search
Create
Home
Shorts
Subscriptions
You
History
Playlists
Your videos
Watch Later
Liked videos
Subscriptions
HannahTheHorri...
theneedledrop
fantano
Johnny 2 Cellos
Wendigoon
Kitty Monk
Connect to the Internet
You're offline. Check your connection.
Retry
No Internet connection