TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN - ĐHQG HCM
KHOA CÔNG NGHỆ THÔNG TIN

MÔN MẠNG MÁY TÍNH

BÁO CÁO ĐỒ ÁN
LẬP TRÌNH SOCKET

Phạm Hải Dương - 19120490
Trần Quốc Huy - 19120537
Nguyễn Bá Ngọc - 19120603

TP.HCM, ngày 26 tháng 12 năm 2020

Mục lục

1 Phân công công việc

Họ và tên	MSSV	Công việc
Phạm Hải Dương	19120490	• Cài đặt trích xuất thông tin của web server từ request. • Tìm hiểu ứng dụng của proxy server trong thực tế. • Lập báo cáo.
Trần Quốc Huy	19120537	• Cài đặt gửi request từ client đến proxy server và từ proxy server đến web server. • Cài đặt việc cấm một host. • Tìm hiểu tài liệu và thiết kế kịch bản cách chạy chương trình.
Nguyễn Bá Ngọc	19120603	• Cài đặt kiểm tra host có bị cấm hay không. • Cài đặt nhận response từ web server và cuối cùng trả về client. • Xử lý đa luồng.

2 Các biến toàn cục và các hàm chức năng chính

2.1 Các biến toàn cục

- *HOST* là địa chỉ IP loopback, do proxy server được cài đặt chung thiết bị với client (127.0.0.1).
- *PORT* là cổng mà proxy server lắng nghe kết nối (8888).
- *SIZE* là kích thước của mỗi phần dữ liệu web server trả về.
- *TIMES* là số kết nối tối đa giữa client và proxy server.

2.2 Các hàm chức năng

- Hàm *Forbid*
  - Tên đầy đủ: **def Forbid(server, site):**
  - Tham số truyền vào:
    - **server**: proxy server của request hiện tại.
    - **site**: web server mà client muốn truy cập (trang đang bị chặn bởi proxy).

  - Chức năng: Tạo một chuỗi *msg* chứa thông tin status code 403 Forbidden, chuỗi này có nội dung như một trang HTML. Sau đó proxy gửi *msg* trực tiếp về client, trình duyệt sẽ hiển thị trang HTML lên màn hình thông báo rằng bị cấm truy cập đến web server này.
  - Kết quả: Hàm không trả về giá trị.

- Hàm *Process*
  - Tên đầy đủ: **def Process(request):**
  - Tham số truyền vào:
    * **request**: là một chuỗi lưu toàn bộ nội dung gói tin HTTP request từ client.
  - Chức năng: Xử lý chuỗi request để trích địa chỉ của web server và port tương ứng. Nếu trong request không có port thì hàm sẽ tự hiểu port là 80.
  - Kết quả: Trả về một tuple gồm hai phần tử, phần tử đầu là địa chỉ web server (biến host), phần tử kia là port (biến port).

- Hàm *IsBlocked*
  - Tên đầy đủ: **def IsBlocked(host):**
  - Tham số truyền vào:
    * **host**: địa chỉ của web server mà client đang request tới.
  - Chức năng: Đọc file *blacklist.conf* và lưu các trang web bị cấm vào trong một danh sách. Sau đó dùng vòng lặp kiểm tra xem host có nằm trong danh sách này hay không. Ở đây dùng vòng lặp vì trang web trong file có thể có thêm tiền tố "http://", nếu dùng toán tử **in** trong Python thì không chính xác cho tất cả các trường hợp.
  - Kết quả: Trả về True nếu host nằm trong danh sách bị cấm, nếu không thì trả về False.

- Hàm *GetRequest*
  - Tên đầy đủ: **def GetRequest(request, server):**
  - Tham số truyền vào:
    * **request**: là một chuỗi lưu toàn bộ nội dung gói tin HTTP request từ client.
    * **server**: proxy server của request hiện tại.
  - Chức năng: Lấy kết quả trích xuất địa chỉ web server (biến host) và port tương ứng từ hàm Process, sau đó gọi hàm IsBlock() xét xem host có nằm trong danh sách cấm của proxy hay không, nếu có thì gọi hàm Forbid() xuất trang HTML 403 Forbidden, còn không thì chuyển tiếp gói tin http request đến web server.
  - Kết quả: Hàm không trả về giá trị.

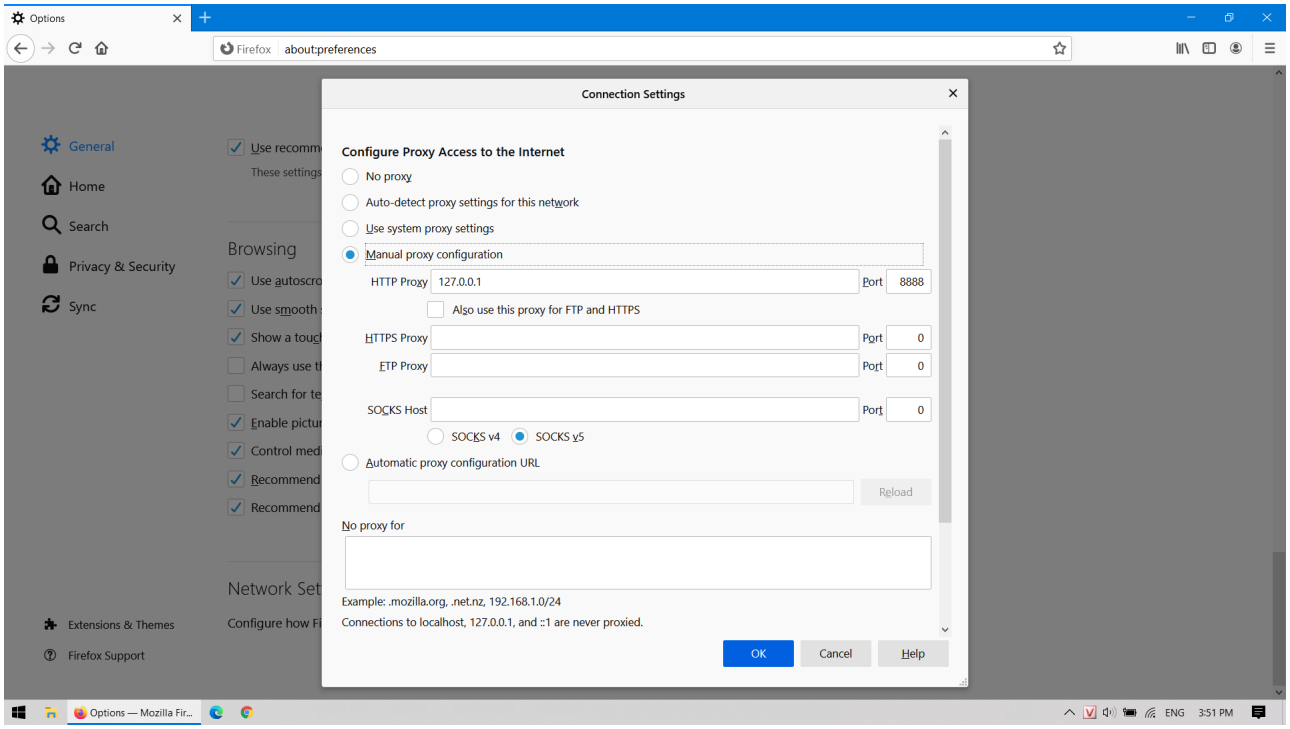
- Hàm *GetResponse*
  - Tên đầy đủ: **def GetResponse(client, server):**
  - Tham số truyền vào:
    * **client**: trình duyệt đang dùng để duyệt web.
    * **server**: proxy server của request hiện tại.
  - Chức năng: Proxy server nhận từng phần của trang từ response của web server, rồi chuyển tiếp chúng lần lượt xuống cho client. Client được một trang web hoàn chỉnh.

  - Kết quả: Hàm không trả về giá trị.

- Hàm *___main___*
  - Tham số truyền vào: Hàm không có tham số.
  - Chức năng: Tạo socket proxy và lắng nghe kết nối. Mỗi khi có một kết nối được thiết lập, một luồng thực thi mới được tạo ra.
  - Kết quả: Hàm không trả về giá trị.

3 Cách chạy chương trình và kết quả chạy được

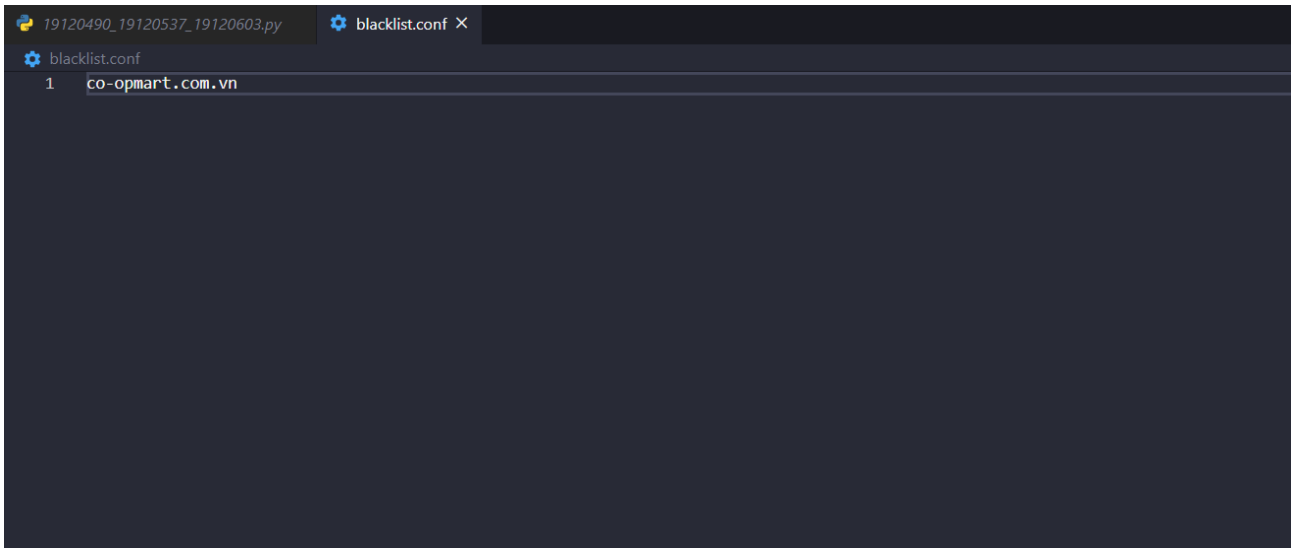
3.1 Cấu hình proxy server cho Firefox



Hình 1: Cấu hình proxy server cho trình duyệt Mozilla Firefox

- Chọn Manual proxy configuration.
- Trong mục HTTP proxy nhập 127.0.0.1 là địa chỉ loop back.
- Trong mục Port tương ứng, nhập 8888 là port mà proxy server sẽ lắng nghe kết nối từ trình duyệt.

3.2 File blacklist.conf

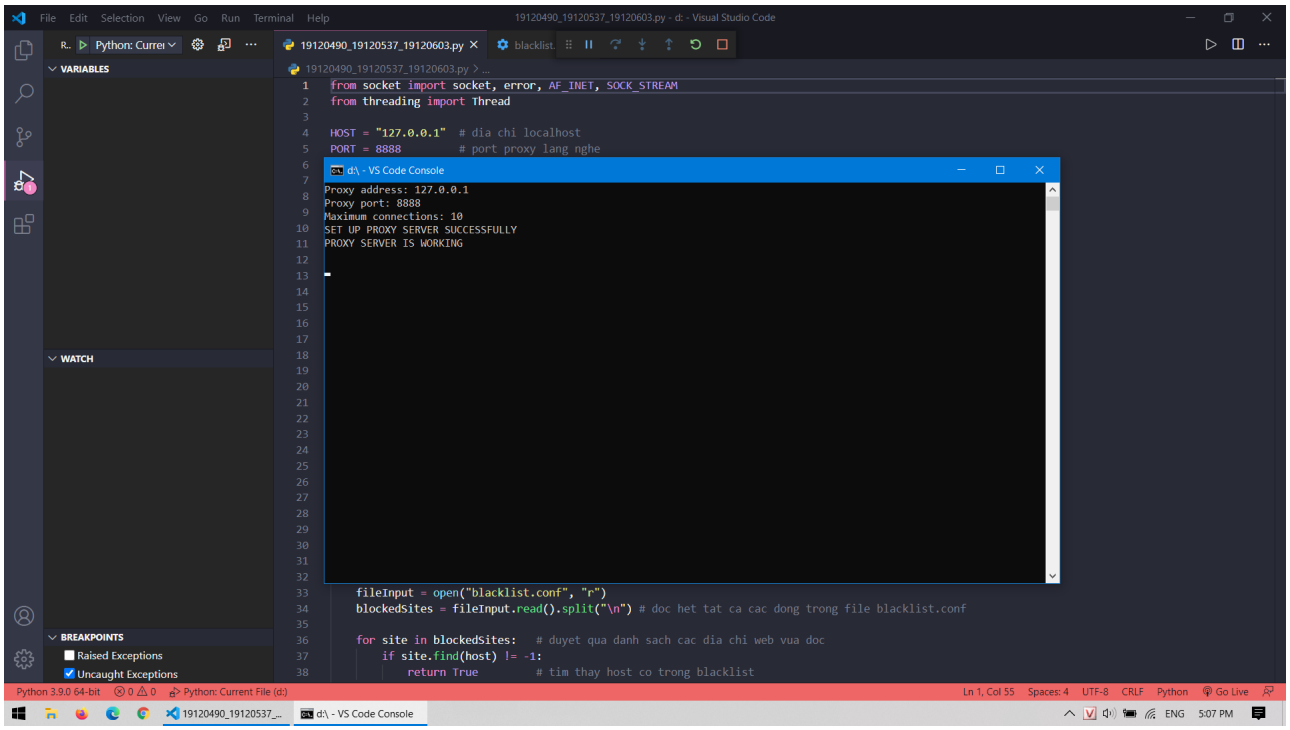


Hình 2: Nội dung file blacklist.conf

- Tạo một file blacklist.conf nằm cùng thư mục với file mã nguồn.
- Thêm vào đó co-opmart.com.vn là địa chỉ cần cấm.

3.3 Chạy chương trình

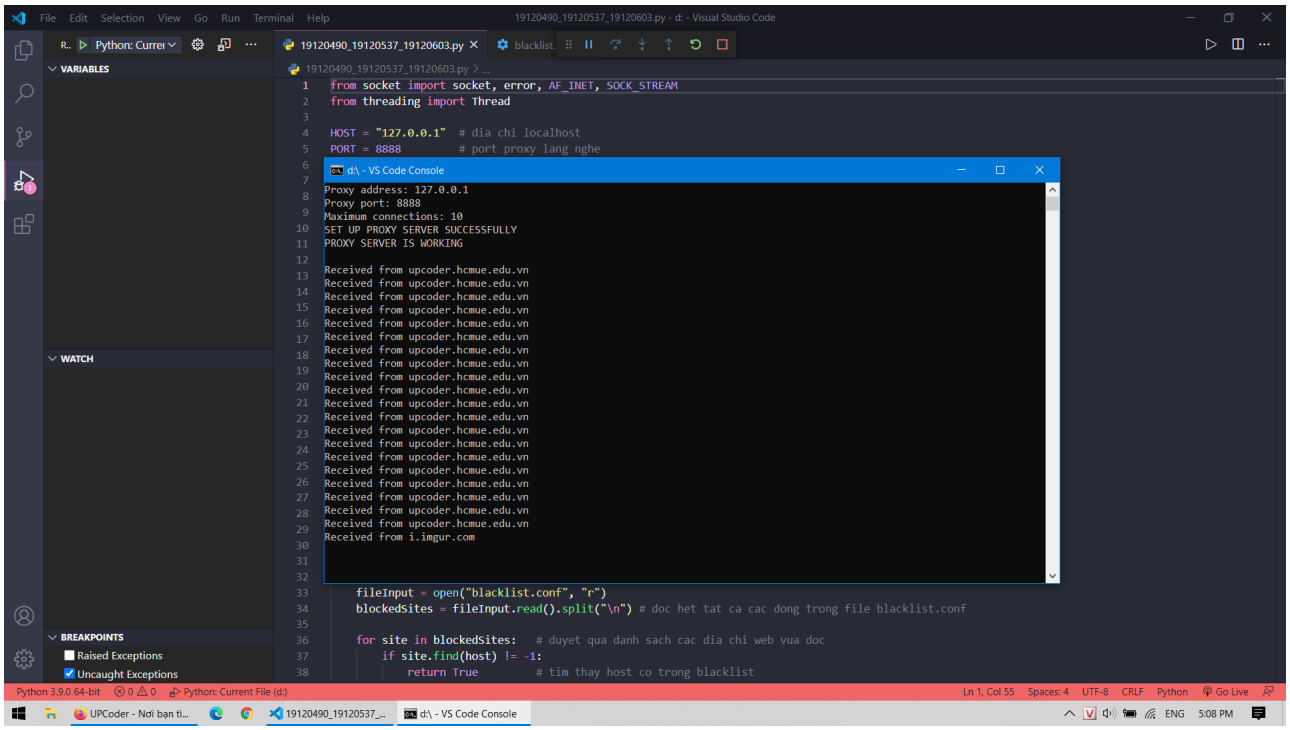
- Nếu dùng Visual Studio Code, cần cấu hình file .json cho Python để chạy trên console screen. Ngoài ra có thể dùng IDLE, Pycharm hay bất kỳ IDE, text editor nào khác.
- Tiến hành chạy (F5), ta được các thông số kết quả như khi đã cấu hình.



Hình 3: Proxy server được thiết lập thành công, đang đợi kết nối

3.4 Truy cập một trang web (http)

Mở firefox, nhập vào thanh địa chỉ upcoder.hcmue.edu.vn, trong console screen hiện lên thông tin nhận trang web như hình dưới.



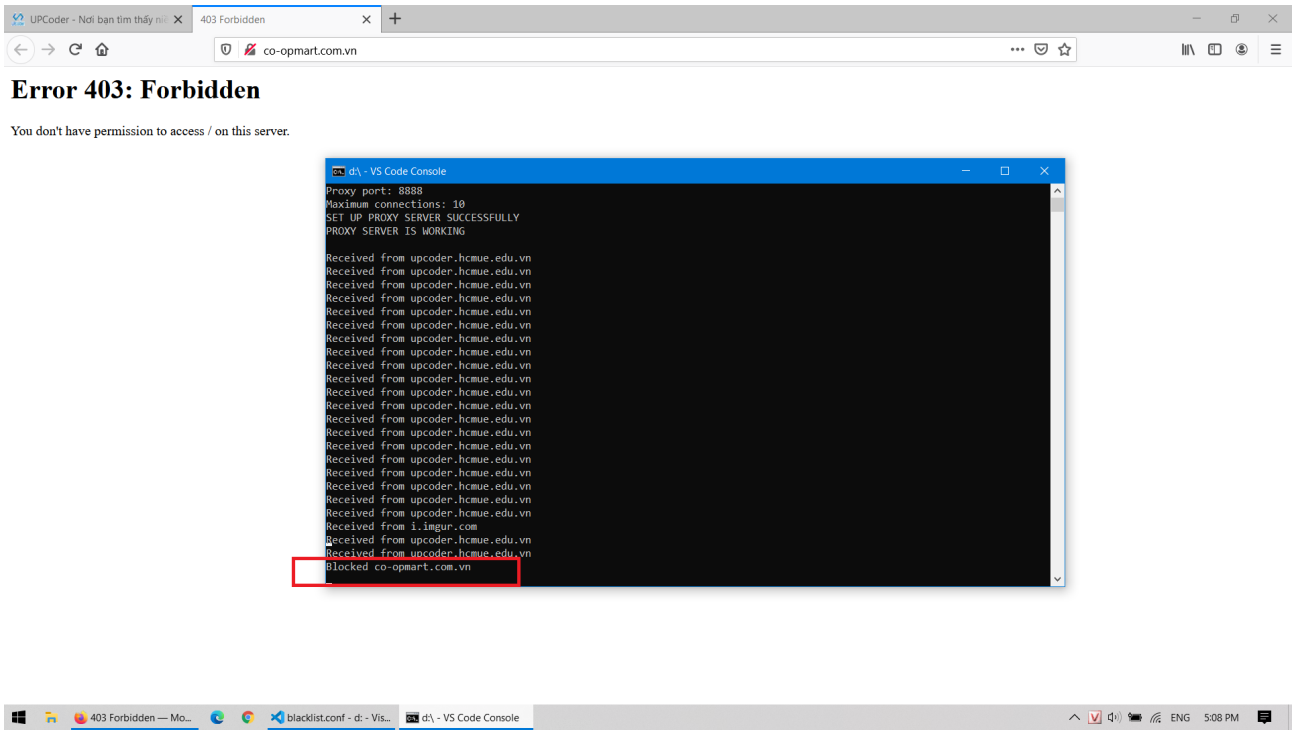
Hình 4: Quá trình nhận các phần từ trang web upcoder.hcmue.edu.vn

Kết quả proxy trả về cuối cùng là một trang web hoàn chỉnh được hiển thị lên trình duyệt.

Hình 5: Trang web kết quả từ request upcoder.hcmue.edu.vn

3.5 Cố tình truy cập một trang bị cấm

Tiếp tục, ta thử truy cập vào co-opmart.com.vn.



Vì trang này nằm trong blacklist.conf, nên khi truy cập, trong console screen chỉ nhận được thông báo rằng đã bị chặn. Phía trình duyệt, một trang HTML báo status Error 403: Forbidden được hiển thị.

4 Mức độ hoàn thành

4.1 Chức năng làm được

- Cho phép client truy cập website thông qua các phương thức GET, POST.
- Hỗ trợ cho giao thức HTTP.
- Xử lý đồng thời các request từ client.
- Proxy server lắng nghe tại cổng 8888, chờ kết nối từ client.
- Chặn tất cả các truy cập đến các domain website có trong file blacklist.conf

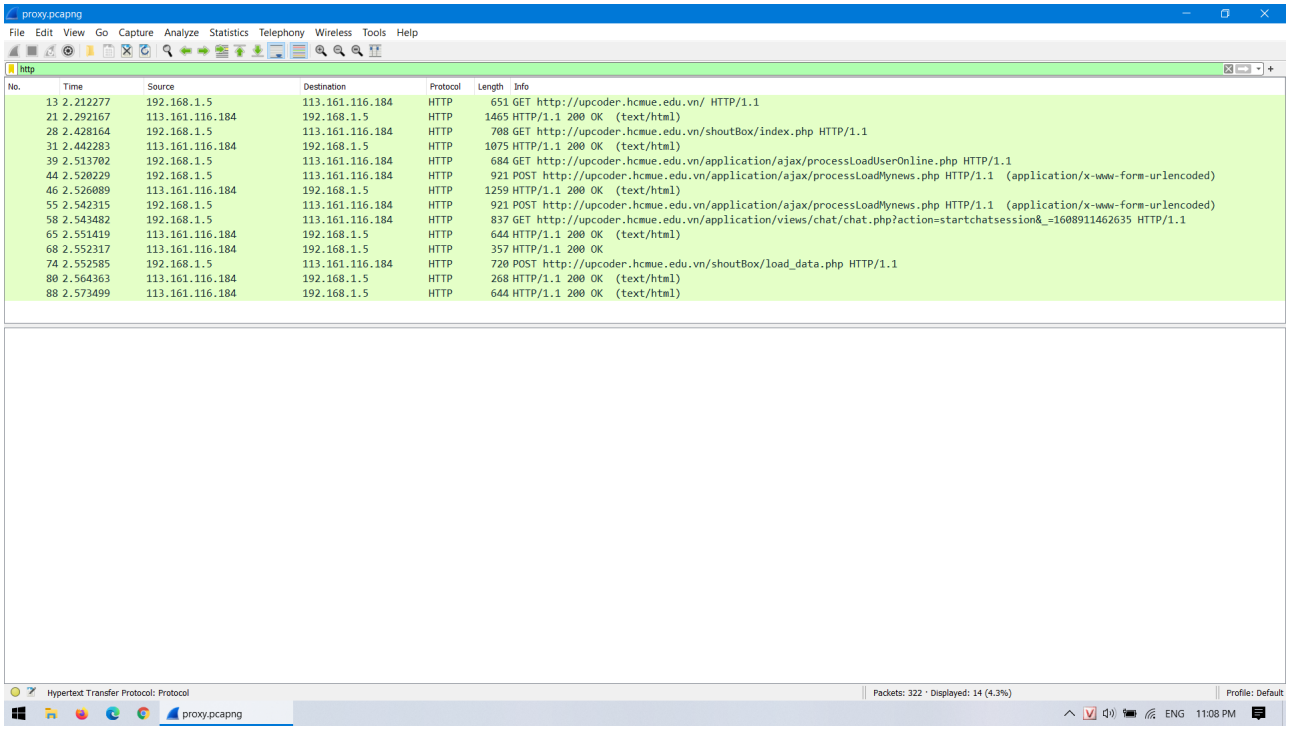
4.2 Chức năng chưa làm được

- Không có.

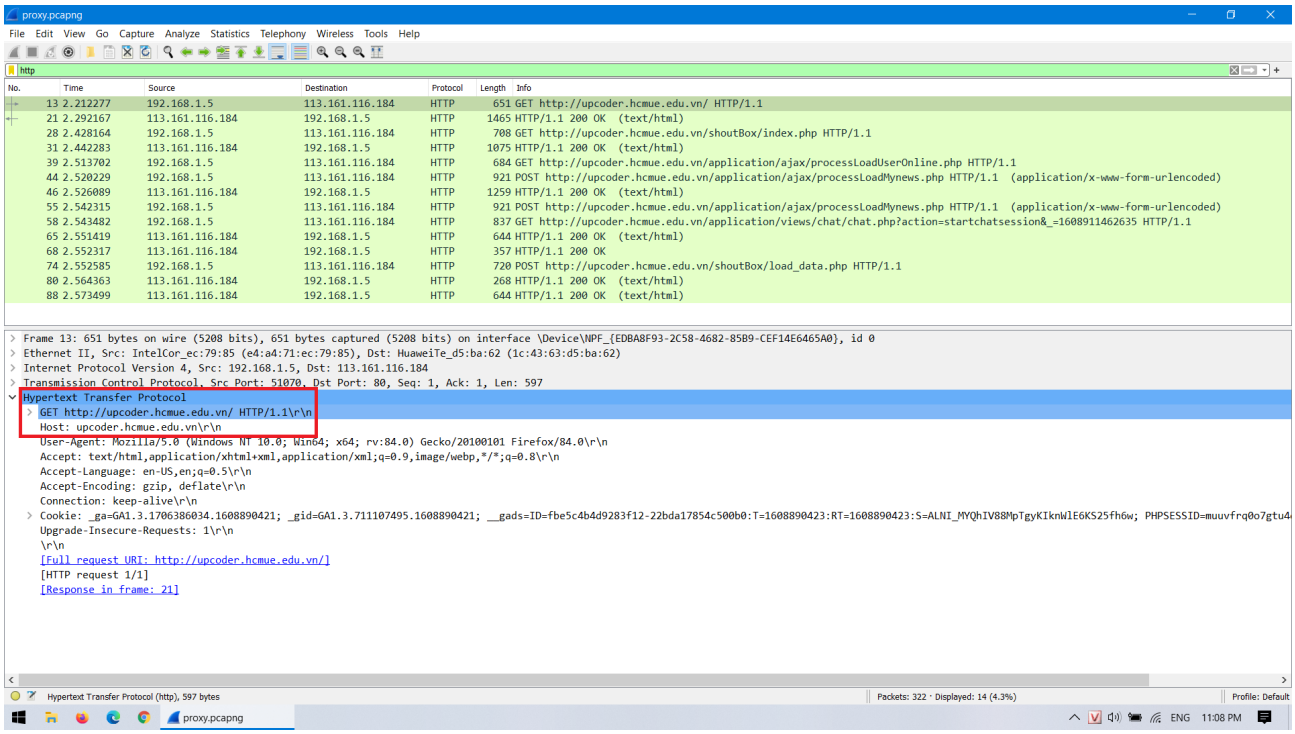
4.3 Mức độ hoàn thành

- Hoàn thành 100% các yêu cầu của đồ án.

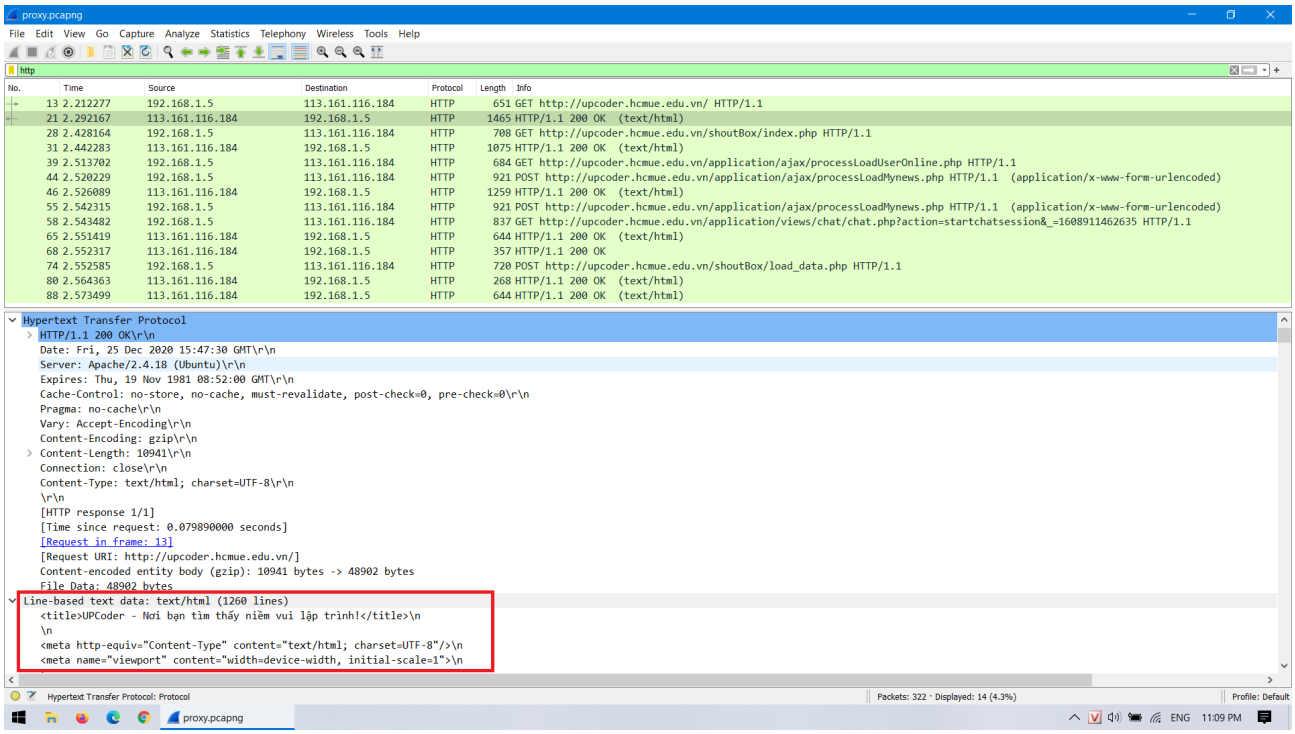
5 Bắt gói tin bằng Wireshark



Hình 6: Các gói tin HTTP bắt được khi truy cập upcoder.hcmue.edu.vn



Hình 7: Cấu trúc một gói tin HTTP request



Hình 8: Cấu trúc một gói tin HTTP response và phần trang web trả về

Quá trình nhận dữ liệu giữa client - proxy server và proxy server – web server:

- Khi bắt đầu thao tác truy cập website, client sẽ gửi một request đến proxy server.
- Proxy server lấy ra những thông tin bao gồm host, port (*Hình 7*) từ gói HTTP request nhận từ client, sau đó kiểm tra host có nằm trong danh sách bị cấm hay không. Nếu có thì gửi gói HTTP 403 Forbidden về cho client, quá trình duyệt trang web dừng tại đây. Nếu không thì proxy server chuyển tiếp gói HTTP request lên web server.
- Sau đó web server sẽ gửi gói HTTP response cho proxy server (*Hình 8*).
- Proxy server chuyển tiếp gói HTTP response này đến cho client.
- Cứ như vậy, client gửi lần lượt từng gói HTTP request và nhận về từng gói HTTP response cho đến khi được một trang web hoàn chỉnh.

6 Proxy Server trong thực tế

6.1 Lọc nội dung truy cập (content filtering)

Ứng dụng này của proxy server đã được minh họa một cách đơn giản trong đồ án, bằng việc cấu hình một file blacklist.conf và cấm bất cứ request nào gửi về các host có trong file, proxy server đã lọc được các trang mà các máy trong mạng cục bộ không được phép truy cập. Trong thực tế cơ chế hoạt động cũng tương tự nhưng hệ thống phức tạp hơn.

6.2 Giám sát truy cập (monitoring)

Ngoài việc lọc bỏ các request, proxy server có thể giám sát truy cập của các máy trong mạng cục bộ, vì các request và response đều đi qua proxy. Do đó có thể dễ dàng kiểm tra được nội dung truy cập của các client.

6.3 Bảo mật thông tin (privacy)

- Người dùng trong mạng cục bộ có thể kết nối với trang web bên ngoài thông qua proxy server thay vì kết nối trực tiếp. Như vậy, bất kỳ request nào đi qua proxy server chỉ được web server hiểu là đến từ proxy đó, web server không biết thông tin, IP của các máy nằm đằng sau proxy server. Điều này cho phép các máy trong mạng cục bộ kết nối ẩn danh với Internet, giúp tăng cường bảo mật cho hệ thống.

- Do trong đồ án này proxy server chạy trực tiếp trên client, nên ứng dụng này không được thể hiện.

6.4 Giảm thời gian tải trang, tiết kiệm băng thông (caching)

- Trong thực tế, proxy server luôn hỗ trợ chức năng caching. Nó tải bản sao của một trang web thường truy cập xuống và lưu vào bộ nhớ đệm (cache). Các bản sao luôn được kiểm tra và cập nhật mới nhất như trên web server.

- Các trang web caching này có thể được truy cập bất cứ lúc nào, bởi bất kỳ các máy nào sử dụng chung proxy server, mà không cần trực tuyến. Như vậy, caching làm giảm độ trễ khi tải trang cũng như tiết kiệm băng thông.

7 Tham khảo

- https://www.youtube.com/watch?v=Lhxwh30kqQ0&t=159s&ab_channel=Devsec
- https://luugiathuy.com/2011/03/simple-web-proxy-python/
- https://realpython.com/intro-to-python-threading/
- https://wheelhouse.solutions/the-benefits-of-using-a-network-proxy-server/
- https://www.rswebsols.com/tutorials/technology/proxy-server-advantages-disadvantages