TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN – ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

MẠNG MÁY TÍNH

BÁO CÁO

PROXY SERVER

Tên sinh viên:

Nguyễn Chí Thành – 1712774

Nguyễn Tấn Thái – 1712753

Nguyễn Viết Thanh – 1712767

Nguyễn Trọng Thắng – 1712757

Lớp: *17CTT6*

Giáo viên: *Lê Hà Minh*

Hồ Chí Minh 27/05/2019

MỤC LỤC

I. TỔNG QUAN

THÔNG TIN ĐỒ ÁN

- Bộ môn: Mạng máy tính
- Giáo viên: Lê Hà Minh
- Tên đồ án: Proxy Server
- Deadline: 23:55 2/6/2019
- Thời gian thực hiện: 1/5/2019 – 29/5/2019

PHÂN CÔNG CÔNG VIỆC VÀ MỨC ĐỘ HOÀN THÀNH

BẢNG PHÂN CÔNG VÀ MỨC ĐỘ HOÀN THÀNH				
Họ Tên	MSSV	Nội dung công việc	Mức độ hoàn thành	Đánh giá
Nguyễn Viết Thanh	1712767	Khởi tạo proxyserver, ý tưởng và code chức năng cache,bắt gói tin dùng wireshark và mô tả quá trình truyền dữ liệu, viết báo cáo	100%	Tốt
Nguyễn Tấn Thái	1712753	Khởi tạo Multithreading cho PS, xử lý các port phụ http. http1.1 và http1.0	100%	Tốt
Nguyễn Chí Thành	1712774	Khởi tạo proxyserver, viết các hàm phụ, test chương trình, viết báo cáo	100%	Tốt
Nguyễn Trọng Thắng	1712757	Chức năng blacklist.conf, test chương trình	100%	Tốt

KIẾN THỨC CẦN CÓ ĐỂ GIẢI QUYẾT

1. Ngôn ngữ lập trình c++
2. Các hàm và thư viện của Socket c++
3. Cách thức hoạt động và vai trò của Proxy Server
4. Nắm rõ mô hình Client – Server
5. Cách sử dụng WireShark

II. CHỨC NĂNG CHÍNH CỦA CÁC FUNCTION

1. `void SetMapBlackList();`
   - Không có tham số truyền vào.
     Chức năng :
   - map<string, int> domainblacklist; được khởi tạo global
   - Hàm này sẽ đọc dữ liệu từ file blacklist.conf sau đó nếu trang web nào mà xuất hiện trong file thì giá trị của nó trong map sẽ được tăng lên.
   - Kết quả : domainblacklist là map chứa các domain trong đó các domain bị blacklist có giá trị là 1.
   - Khi cần kiểm tra trang web đó có nằm trong blacklist hay không thì chỉ cần sử dụng toán tử [ ].
2. `UINT ProxyServer(void* Iparam)`
   - Tham số truyền vào : Iparam kiểu void tự định nghĩa, trong chương trình là 1 socket được truyền vào (Client Socket)
   - Hàm này trả về kiểu UINT tương ứng 0 (tiểu trình có lỗi) hoặc 1(tiểu trình hoàn thành)

➔ Chức năng : thực hiện các công việc của ProxyServer. Gồm

2.1. Nhận request từ client.

2.2. Kiểm tra request, nếu rơi vào các trường hợp không hợp lệ (domain không hợp lệ,là HTTPs,không phải method get hoặc post) thì thoát thread.

2.3. Nếu phần Host nằm trong blacklist.conf thì thực hiện trả về header HTTP 403 và html 403 để thông báo cho client trang này đã bị block bởi Proxy server.

2.4. Mở 1 thread mới với thông số chính là Iparam để thực hiện với các request khác trên cùng 1 Client(chỉ hỗ trợ http1.1 vì http1.0 yêu cầu kết nối khác nhau cho mỗi lần truyền dữ liệu).

2.5. Thực hiện việc kiểm tra file cache có tồn tại hay không, nếu có tồn tại và chưa expire thì sẽ load từ file cache để trả về client và thoát.

2.6. Tạo một socket mới để connect tới webserver với phần addr là địa chỉ host, và IP là ip của URL đã được tách ra (mặc định là 80, hoặc có thể là các port 8080 hoặc 8008, là các port phụ của http.)

2.7. Gửi request lên webserver. Sau đó nhận phần header thuộc phần response của request về và kiểm tra các trường hợp.(HTML response code,Cache control,…) và gửi response này về client.

2.8. Nếu HTTP CODE là 304 thì thực hiện việc đóng các socket và thoát thread vì browser đã cache phần này, browser sẽ tự load dữ liệu lên.

2.9. Thực hiện việc load dữ liệu từ Webserver và gửi về Client và lưu thành file cache cùng thời gian file cache được lưu. (Chỉ lưu cache khi trang web có tag Cache-control là public hoặc không có Cache-control)

2.10. Đóng các socket và giải phóng bộ nhớ (nếu có) nếu cần phải thoát thread.

- ***Phương pháp cache :*** Cache Expiration
- Các file cache khi được lưu xuống (lưu vào thư mục cache) sẽ lưu kèm 1 t_time là giá trị thời gian của thời điểm lưu xuống(thư mục time), khi các file cache này được load lên, thì đầu tiên sẽ load giá trị time để so sánh với thời gian hiện tại, nếu khoảng cách giữa 2 thời gian này >= 3600 giây (1 giờ) thì sẽ báo file cache này hết hạn và tiến hành renew.

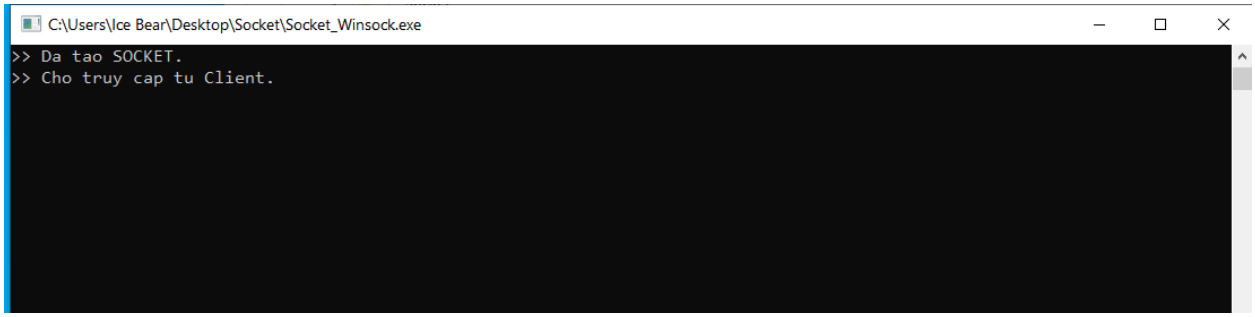
3. `void Init_Server()`
   - Hàm khởi tạo SOCKET và tạo SOCKET cho ProxyServer với port 8888 để lắng nghe và kết nối các Client, sau đó sẽ bật 1 thread ProxyServer để xử lý riêng với từng Client, thực hiện chức năng multithreading cho nhiều client.
   - Không có tham số truyền vào và cũng không có giá trị trả về
4. `bool isGETorPOSTmethod(char* req);`
   - Hàm kiểm tra HTTP request là phương thức GET hay POST
   - Tham số truyền vào: Chuỗi char * req (Chuỗi HTTP request)
   - Hàm trả về kiểu bool: nếu là 2 phương thức trên thì trả về true, ngược lại là false
5. `bool GetDomainName(char* request, char* dname)`
   - Hàm lấy Domain name (thực chất là hostname) của HTTP request
   - Tham số truyền vào: char* request (chuỗi HTTP request) và char *dname (chuỗi lưu kết quả)
   - Kết quả được lưu vào char* dname
   - Hàm trả về true nếu thành công, ngược lại thất bại là false
   - Ví dụ : vingroup.net, baochinhphu.vn
6. `bool isHTTPs(char* request)`
   - Hàm kiểm tra domain có phải https hay không (kiểm tra có cổng 443)
   - Tham số truyền vào: char* domain (chuỗi domain name)
   - Hàm trả về true nếu domain dùng giao thức https còn false là dùng http
7. `bool isHTTP10(char* request)`
   - Hàm kiểm tra phiên bản của HTTP được request sử dụng có phải là HTTP1.0 hay không
   - Tham số truyền vào: char* request (chuỗi HTTP request)

- Hàm trả về true nếu phiên bản HTTP là 1.0, ngược lại false

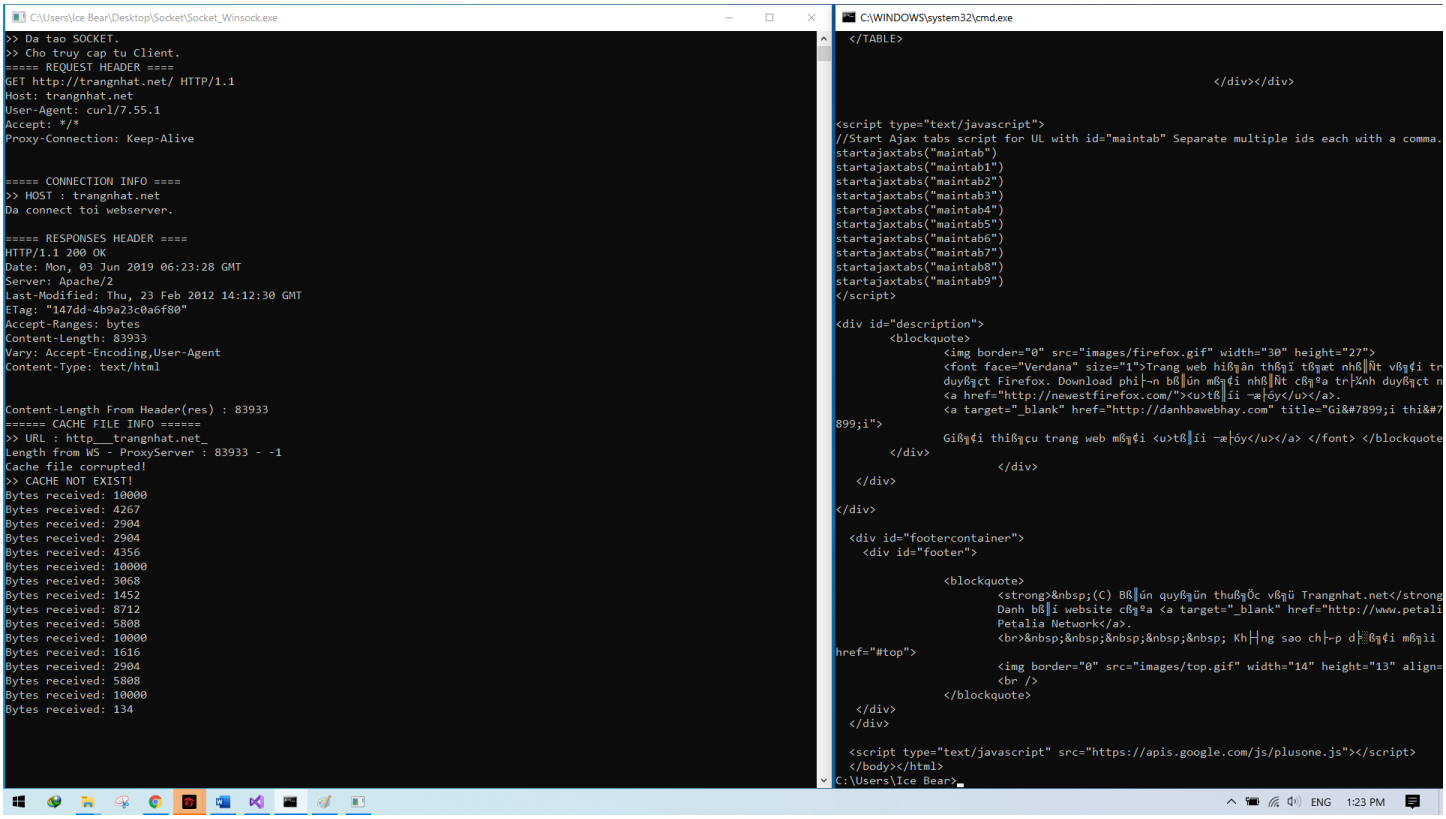
8. `int getLengthOfFile(string)`
   - Hàm lấy độ dài của phần dữ liệu được server trả về (Content – length trong header)
   - Tham số truyền vào: string head (Chuỗi header)
   - Hàm trả về kích thước của phần dữ liệu đó (kiểu int)
9. `int getFileSize(const std::string& fileName);`
   - Tham số truyền vào: const string &filename (Tên file)
   - Hàm trả về kích thước của file cache được lưu có tên filename.
10. string getCacheLink(char* request)
    - Lấy URL thuộc phần GET trong HTTP request để dùng cho cache
    - Tham số truyền vào: char* request (chuỗi request)
    - Kết quả trả về là chuỗi URL lấy được.
    - Ví dụ : http://tiin.vn/default.html
11. string convertFileName(string linktemp)
    - Hàm làm nhiệm vụ chuẩn hóa tất cả các ký tự đặc biệt (/, \, *, :, ….) để có được một chuỗi tên hợp lệ cho các file trong cache
    - Tham số truyền vào: string linktemp (Chuỗi chưa được chuẩn hóa)
    - Kết quả trả về là chuỗi string đã được chuẩn hóa
    - Ví dụ : http___tiin.vn_default.html
12. bool checkCacheControl(string response)
    - Tham số truyền vào: string response (Chuỗi HTTP response)
    - Cache-control là tham số cho biết client hay proxy server có thể cache hay không. Để proxy server có thể cache được thì cần phải có thuộc tính public.
    - Được tham khảo tại : https://viblo.asia/p/nhung-dieu-can-biet-ve-web-cache-Qbq5QJLLKD8
    - Trả về true nếu trong response có public và ngược lại.

III. CHẠY CHƯƠNG TRÌNH VÀ KẾT QUẢ

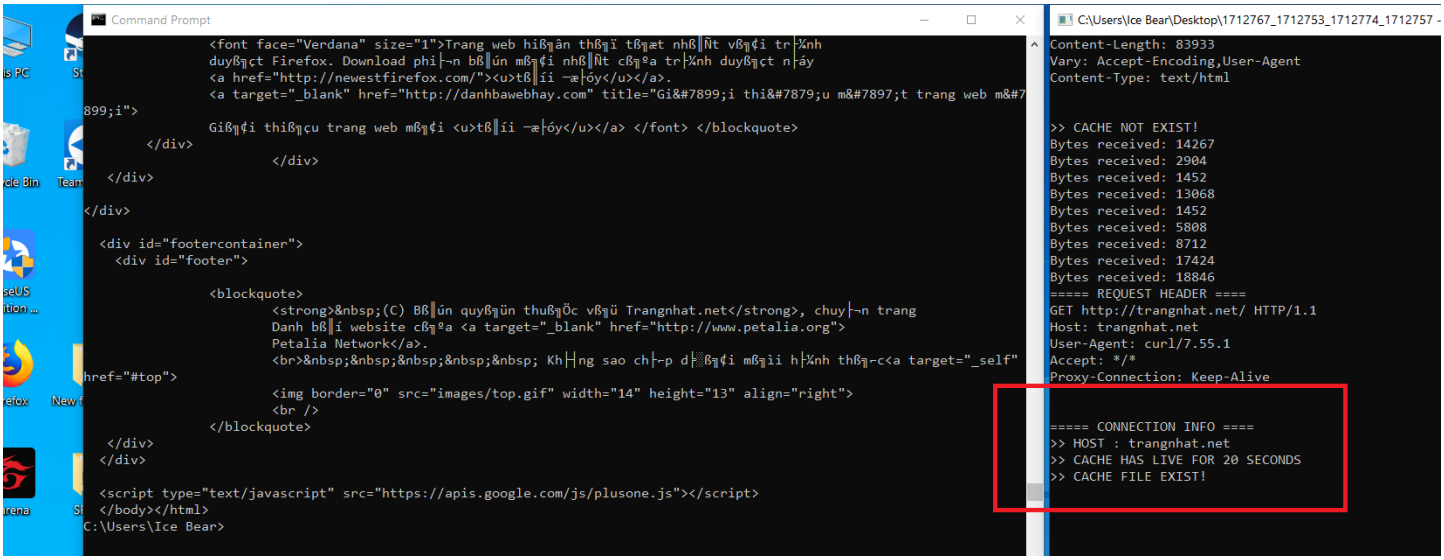
- Mở file ProxyServer



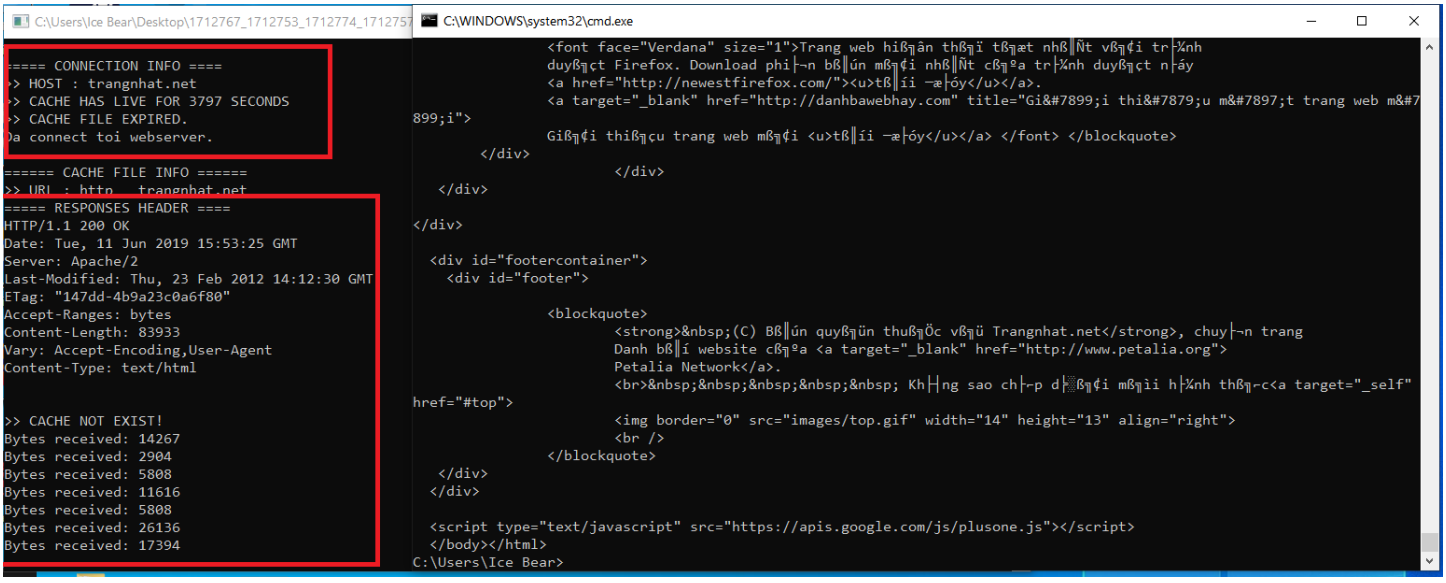
- Thực hiện sử dụng cURL để tiến hành test proxy server.
  Xử dụng cURL với lệnh curl http://example.com --proxy 127.0.0.1:8888 để lấy dữ liệu từ trang web example.com thông qua proxy server. (giao thức http 1.1)



- Dựa vào ảnh trên ta có thể thấy các header, thông tin file cache và size của phần response được WS trả về của tab Proxyserver , bên phần command prompt chính là thể hiện phần dữ liệu được WS trả về.
- Thực hiện đoạn lệnh curl http://example.com --proxy 127.0.0.1:8888 một lần nữa.

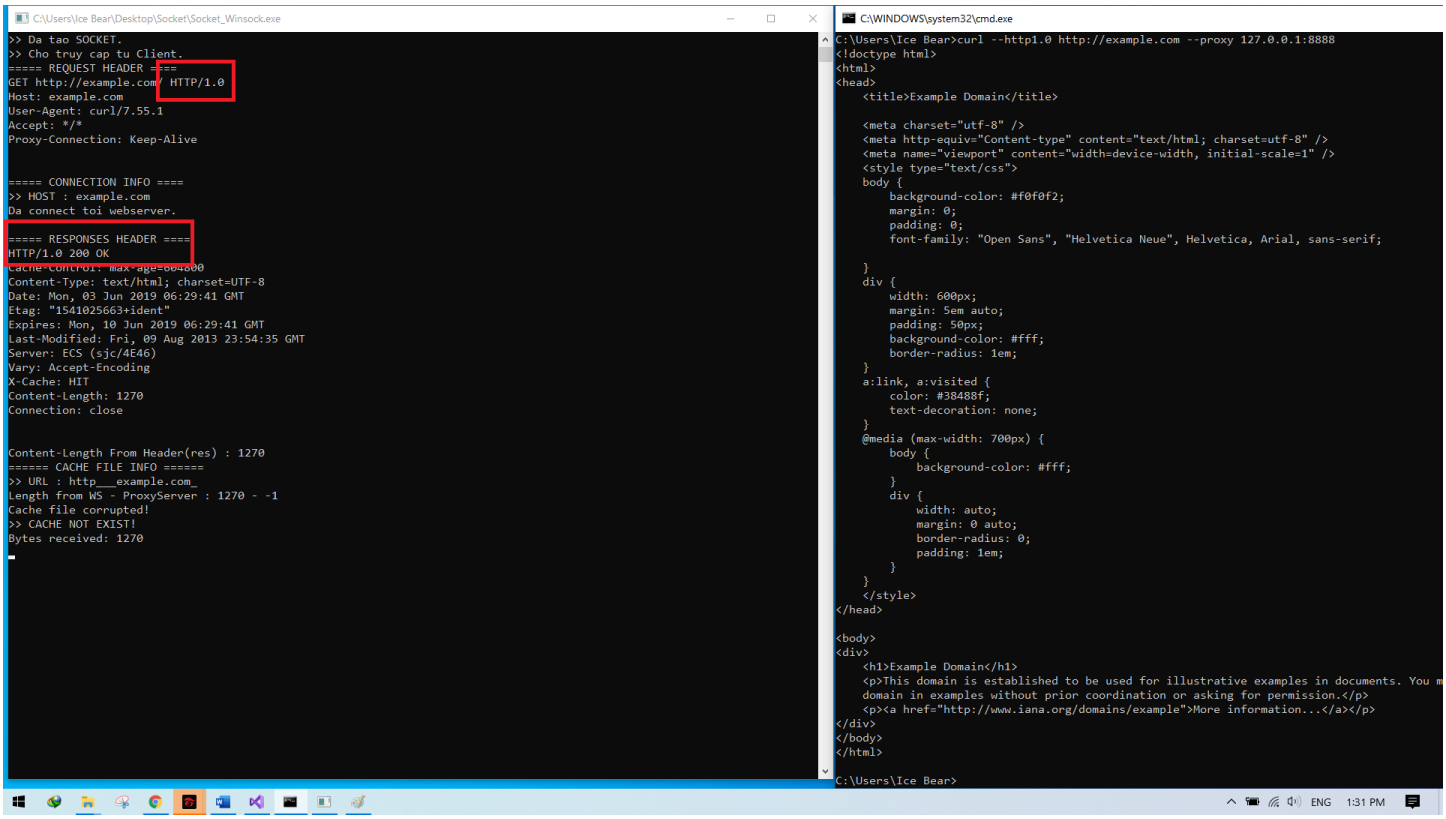


- Sau khi load lại bằng command trên thì ta thấy dữ liệu đã được sử dụng từ file cache (thay vì nhận dữ liệu và in ra byte receives như lần đầu) và file cache đã tồn tại 20 giây.



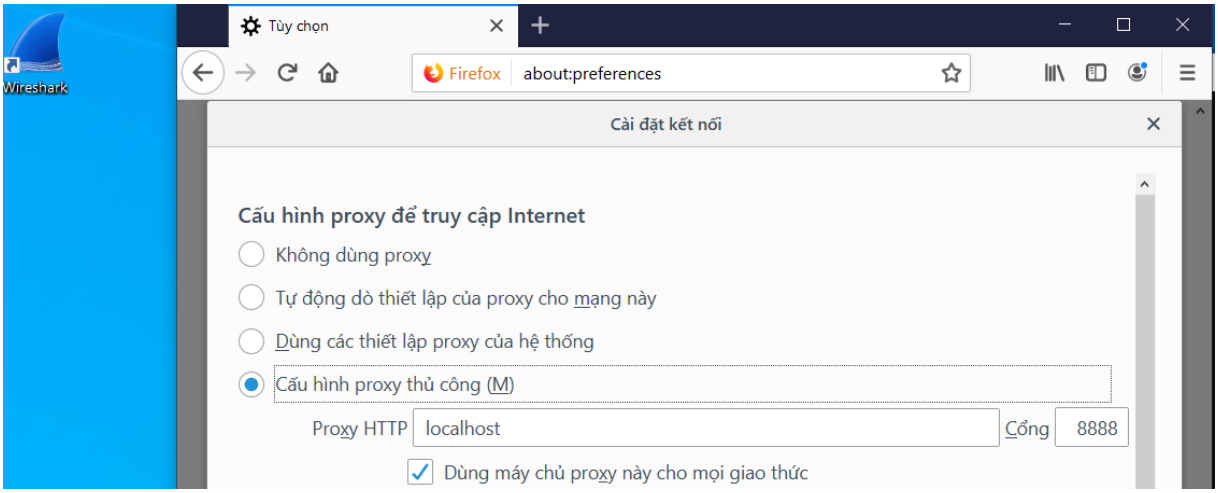
Đối với trường hợp file cache bị hết hạn (LIVE 3797s > cacheTTL: 3600s) thì sẽ thực hiện quá trình connect tới webserver để gửi request, nhận về dữ liệu gửi về client và lưu thành file cache mới + update thời gian cập nhật cache mới.

- **Kiểm tra với http 1.0**
  - Sử dụng curl với lệnh curl --http1.0 http://example.com --proxy 127.0.0.1:8888

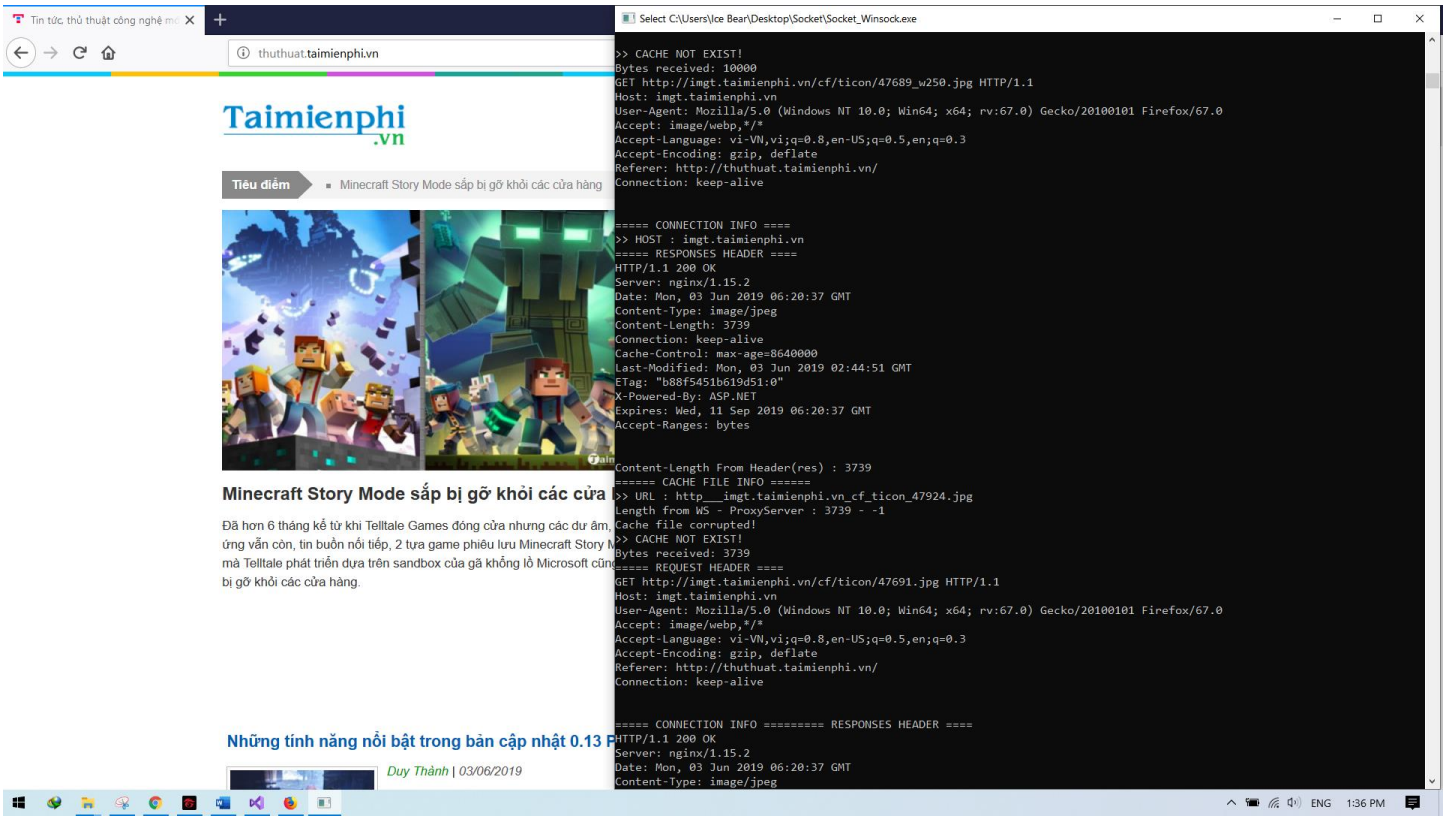


  - Ta thấy phần request và response đều là http1.0 và việc gửi và nhận dữ liệu vẫn diễn ra bình thường.

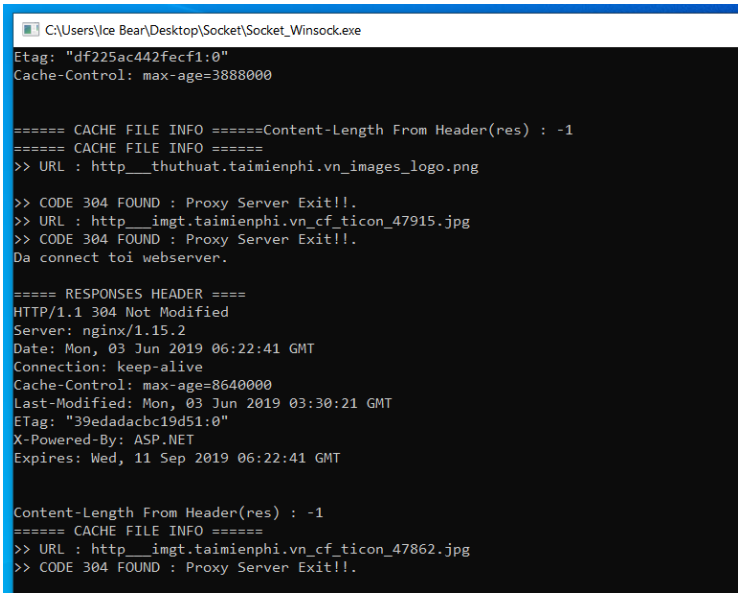
- **Test chương trình sử dụng trình duyệt.**



- Sử dụng trình duyệt firefox đã được cấu hình với proxy server.



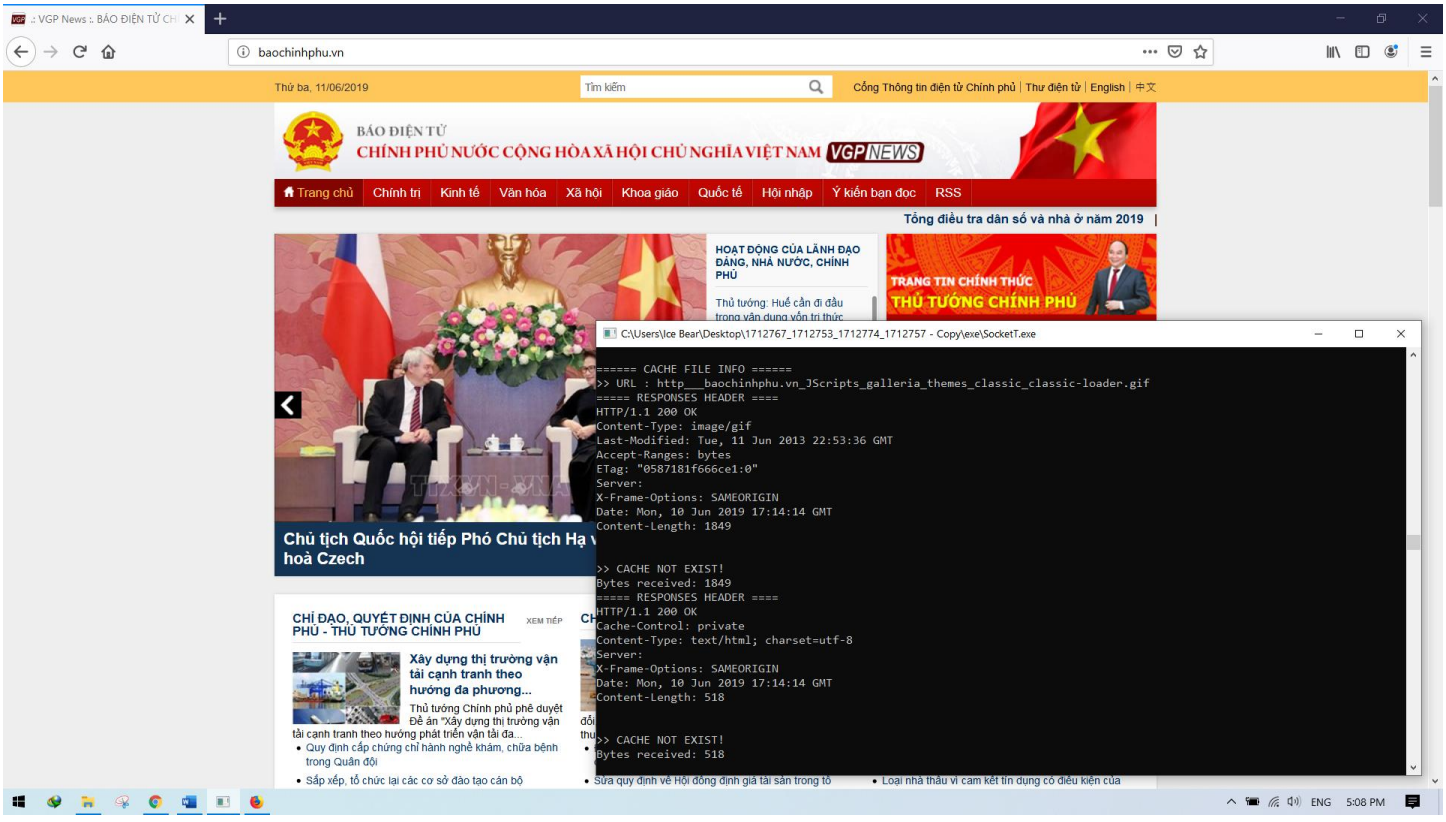
- Thực hiện load trang web thuthuat.taimienphi.vn, ta thấy rất nhiều request và response được trả về.
- Khi thực hiện việc reload trang web.



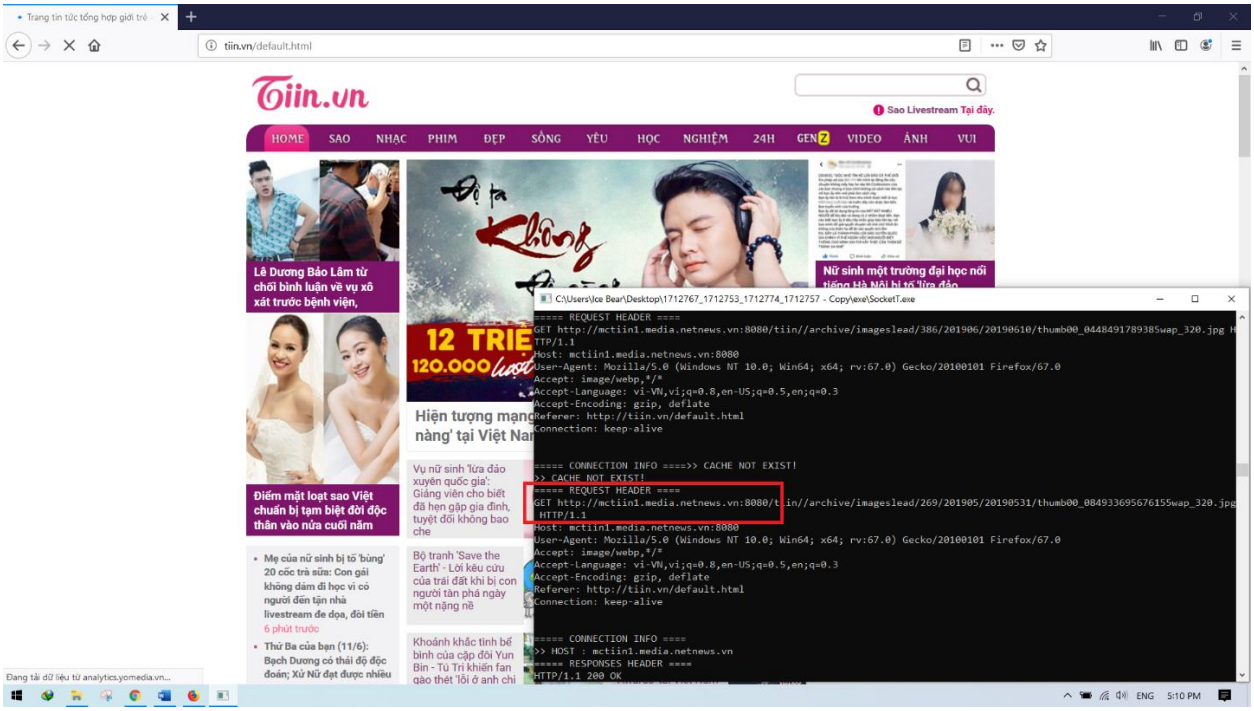
- Các response trả về của hình ảnh hầu hết là 304 vì đã nằm trong cache của browser. Nên proxy server không cần lấy và trả về dữ liệu

Thử với các trang web khác :

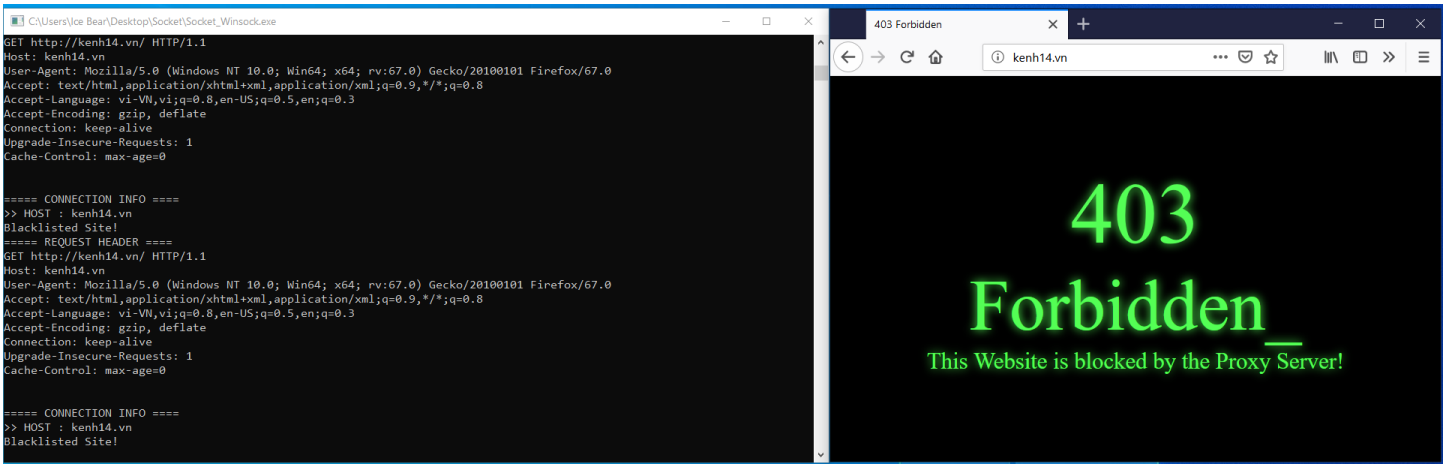
Trang baochinhphu.vn



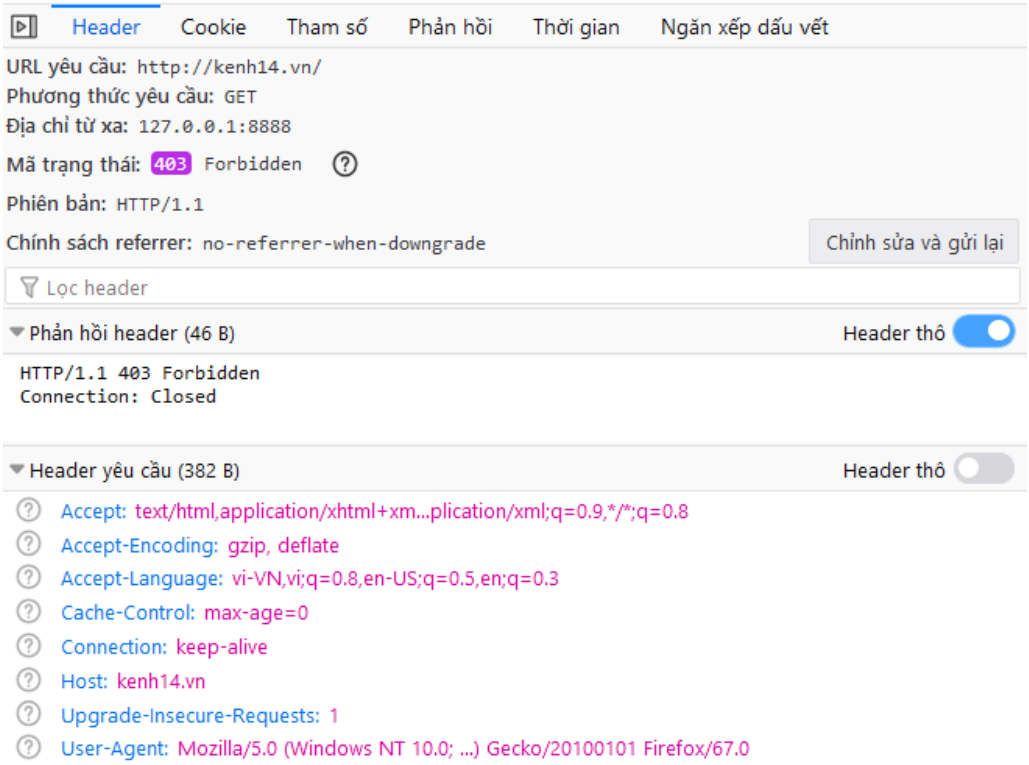
Trang tiin.vn : trang web này lưu ảnh ở webserver có port là 8080.



- **Đối với các website nằm trong blacklist.conf**

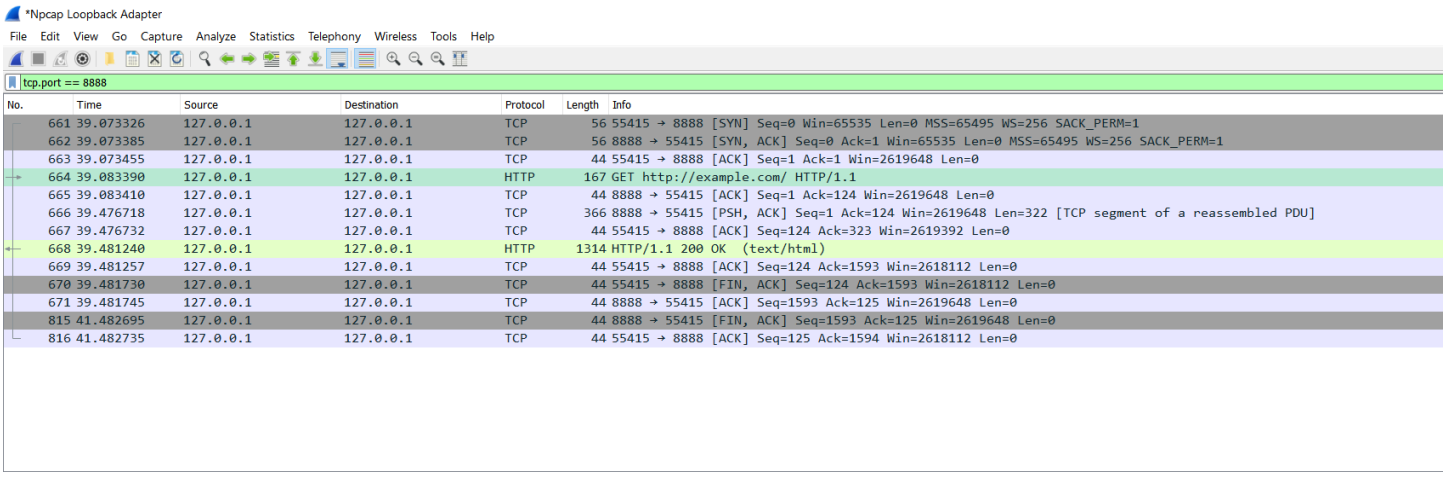


- Proxy server sẽ báo website này nằm trong blacklist, và trả về đoạn header + code html 403
- Phản hồi header : HTTP/1.1 403 Forbidden
- Phương pháp blacklist : Sử dụng STL MAP để truy xuất nhanh các domain bị blacklist thay vì phương pháp truyền thống là mỗi thread ta cần mở file và duyệt từng dòng xem domain có nằm trong file hay không thì trước khi khởi tạo proxy server ta sẽ khởi tạo một Map các domain bị blacklist và sau đó truy xuất nhanh bằng toán tử [] của map.
- Blacklist 2 trường hợp là www. và không có www.



IV. DÙNG WIRESHARK BẮT GÓI TIN TẠI PROXY SERVER

- Thử với trang web www.example.com và connect bằng cURL
  Lệnh : curl http://example.com --proxy 127.0.0.1:8888
- Ta có được các gói tin http bắt tại proxy server như sau:



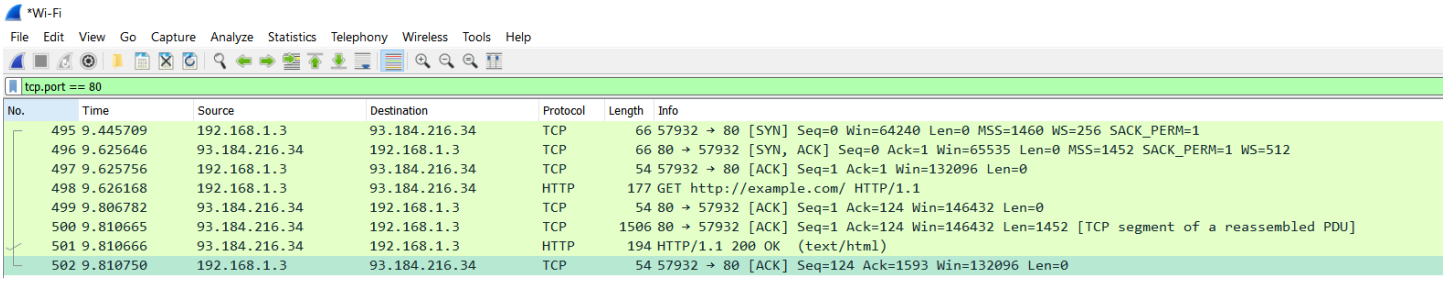
Hình 1 – Minh họa quá trình giữa Client và PS

- Theo như các gói tin trên, port của Client là 55415, và port của PS là 8888.

- Với 3 gói tin đầu tiên : ta thấy Client gửi gói SYNchronize tới ProxyServer. PS nhận và trả về gói SYN-ACK. Client nhận gói SYN-ACK và trả về gói ACKnowledge.

⇨ Hoàn thành việc cài đặt TCP Socket connection giữa Client và PS

- Sau đó Client gửi HTTP Request đến ProxyServer. (gói 664)



Hình 2 – Minh họa quá trình giữa PS và WS

- Tương tự với hình 2, 3 gói tin đầu chính là quá trình khởi tạo TCP socket connection giữa ProxyServer và WebServer, với địa chỉ của ProxyServer là địa chỉ mạng của máy, còn địa chỉ của Webserver chính là địa chỉ được phân giải từ domain example.com.
- Proxy Server sẽ gửi request nhận được từ Client đến webserver. (gói 498)
- Sau đó server sẽ trả về response lần lượt qua 3 gói tin 499 500 và 501. Response này gồm phần header và phần html.

- 3 gói tin tiếp theo của hình 1 gồm các gói ACK – PSH,ACK – ACK (665->667) tương ứng quá trình Proxy Server gửi response header nhận được từ webserver đến Client. Sau đó là gửi phần content html. (668)

- 5 Gói tin cuối gồm các gói tin ACK – FIN,ACK – ACK được gửi qua lại giữa Client và PS thể hiện việc close connection của SOCKET.

- Trong trường hợp có cache hoặc website nằm trong blacklist, thì quá trình giữa PS và WS không xảy ra. Chỉ có quá trình giữa Client và PS.

V. TẠI SAO LẠI CẦN PROXY SERVER

1. Do mọi thông tin truy xuất phải thông qua Proxy nên chúng ta có thể quản lý được mọi thông tin ra và vào ví dụ: Mọi yêu cầu của máy khách phải qua Proxy server, nếu địa chỉ IP có trên proxy, nghĩa là website này được lưu trữ cục bộ, trang này sẽ được truy cập mà không cần phải kết nối Internet, nếu không có trên Proxy server và trang này không bị cấm, yêu cầu sẽ được chuyển đến server thật, DNS server… và ra Internet.

2. Các dịch vụ proxy đều có lợi trong việc logging :Vì các proxy server hiểu các giao thức cơ bản, chúng cho phép logging đạt hiệu quả. Ví dụ, thay vì logging tất cả những dữ liệu đã truyền, một FTP (File Transfer Protocol) proxy server chỉ ghi lại những lệnh đã tạo và những đáp ứng của remote server, điều này giúp việc logging ít và hữu dụng hơn.

3. Đáp ứng được nhu cầu truy xuất của cá nhân và vừa đảm bảo an toàn cho hệ thống cục bộ do chúng ta sử dụng địa chỉ ẩn danh ,và mọi truy xuất đều thông qua proxy nên thông tin cục bộ không trực tiếp tương tác với bên ngoài.

4. Các dịch vụ proxy cho phép người dùng truy cập các dịch vụ Internet "trực tiếp". Với các dịch vụ Proxy, các người dùng luôn nghĩ rằng họ đang tương tác trực tiếp với các dịch vụ Internet. Ví dụ các người dùng chỉ cần gõ vào địa chỉ của một trang web nào đó thì trang web được trình duyệt hiển thị lên cho người dùng. Dĩ nhiên là có nhiều công việc phải làm ở bên trong nhưng nó là trong suốt đối với người dùng. Người dùng truy cập các dịch vụ Internet từ chính những hệ thống riêng của họ, mà không cần cho phép các gói tin truyền trực tiếp giữa hệ thống của người dùng và Internet đảm bảo an toàn cho hệ thống.

5. Proxy server tích lũy và cứu file , những file mà thường được yêu cầu bởi ngàn người dùng trên internet trong dữ liệu đặc biệt , gọi là cache . Do đó , proxy server chúng có thể tăng tốc độ truy nhập internet. Cache của proxy server có thể đã sẵn chứa thông tin bạn cần trong thời gian bạn yêu cầu , làm cho proxy server có thể phân phối thông tin ngay lập tức mà không cần phải truy tìm thông tin ngoài internet.

6. Một Proxy Server thường nằm bên trong tường lửa , giữa trình duyệt web và server thật , làm chức năng tạm giữ những yêu cầu Internet của các máy khách để chúng không giao tiếp trực tiếp Internet .Người dùng sẽ không truy cập được những trang web không cho phép (bị công ty cấm). Vd :Admin không muốn nhân viên của mình đọc báo hay chơi game online trong giờ làm việc , bằng cách dùng proxy server admin có thể khóa một số site được chỉ định.

7. Proxy server làm cho việc sử dụng băng thông có hiệu quả do chúng ta quản lý được các hoạt động của người dùng.Nên có thể giới hạn thông tin nào được dùng và không dùng tránh được việc nghẽn băng thông.

VI. TỔNG KẾT

- Kiến thức thu được
- Cách lập trình socket
- Bản chất, vai trò và cách thực hiện proxy server
- Các kiến thức về giao thức HTTP
- Quá trình truyền nhận dữ liệu giữa Proxy-Client và Proxy-Web Server
- Kỹ năng làm việc nhóm và tra tài liệu
- Tài liệu tham khảo

- https://docs.microsoft.com/en-us/windows/desktop/winsock/windows-sockets-start-page-2
- https://github.com/
- https://vinahost.vn/proxy-server.html
- https://quan-cam.com/posts/http-caching
- https://www.inetdaemon.com/tutorials/internet/tcp/3-way_handshake.shtml
- https://viblo.asia/p/tim-hieu-ve-http-caching-djeZ1BRJlWz?fbclid=IwAR2FiVXgI7x2osaJMqCXr77WLD7pfNAHRZCYiFg4UYhhEKUrB4-2kdVQv-8