

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN

CHỦ ĐỀ: XÂY DỰNG PROXY SERVER

Môn học: Mạng máy tính
GV hướng dẫn: Đỗ Hoàng Cường
Lớp: 23TNT1TN

TP. Hồ Chí Minh, tháng 12 năm 2024

Mục lục

Danh mục hình	2
Danh mục bảng	2
1 Giới thiệu	3
1.1 Tổng quan	3
1.2 Mục tiêu của đồ án môn học	3
1.3 Phạm vi thực hiện	4
2 Phân tích và thiết kế hệ thống	4
2.1 Yêu cầu chức năng	4
2.2 Luồng hoạt động của hệ thống	5
2.2.1 Máy chủ Transparent Proxy	5
2.2.2 Máy chủ Man-in-the-Midle Proxy (MITM)	5
2.3 Công nghệ sử dụng	8
2.4 Cài đặt công cụ	8
3 Cấu trúc mã nguồn và cài đặt	11
3.1 Cấu trúc mã nguồn	11
3.1.1 Phân tích các lớp chính	11
3.1.2 Lớp ProxyServer	12
3.1.3 Lớp ClientHandler	12
3.1.4 Lớp SocketHandler	12
3.1.5 Lớp ThreadPool	13
3.1.6 Lớp PUI	13
3.1.7 Các lớp khác	13
3.2 Cách biên dịch và chạy chương trình	13
3.2.1 Cài đặt OpenSSL và thiết lập thư viện:	13
3.2.2 Dịch chương trình	15
3.2.3 Chạy chương trình và thử nghiệm:	15
3.2.4 Kiểm tra:	15
3.2.5 Cài đặt chứng chỉ lên máy	16
4 Kết quả đạt được	18
4.1 Những chức năng đã hoàn thành:	18
4.2 Hình ảnh minh họa:	18
5 Kết luận	18
6 Báo cáo công việc	20
Tài liệu tham khảo	21

Danh mục hình

1	Sơ đồ khái ý tưởng của máy chủ Transparent Proxy	6
2	Sơ đồ ý tưởng Man-in-the-Middle Proxy Server	7
3	Trang web dự án MSYS2	8
4	Cửa sổ console của MSYS2	9
5	Cài đặt chương trình dịch g++	9
6	Cài đặt make	10
7	Trang web cài đặt thư viện OpenSSL	10
8	Cài đặt chương trình đóng gói phần mềm	11
9	Cấu trúc mã nguồn đồ án	12
10	Trang web tải thư viện Openssl	14
11	Tìm kiếm Edit environment variables for your account	14
12	Cửa sổ Environment Variables	14
13	Cửa sổ Edit environment variables	15
14	Cài đặt Proxy lên máy tính	16
15	Thiết lập ProxyServer trên máy tính	16
16	Mở ProxyServer	17
17	Tìm thư mục chứa chứng chỉ	17
18	Tìm vị trí cài đặt chứng chỉ vào hệ thống	17
19	Cài đặt chứng chỉ vào hệ thống	18
20	Cho phép truy cập chatgpt.com	19
21	Chặn truy cập của chatgpt.com trường hợp Transparent	19
22	Chặn truy cập của chatgpt.com trường hợp MITM	19

Danh mục bảng

1	Phân chia công việc	20
---	-------------------------------	----

Tóm tắt nội dung

Báo cáo này trình bày quá trình nghiên cứu và triển khai của nhóm về Proxy Server. Nội dung bao gồm tổng quan về proxy, phân tích các chức năng chính của hệ thống và một số đoạn mã nguồn tiêu biểu trong quá trình thực hiện. Toàn bộ mã nguồn của dự án đã được đăng tải trên kho lưu trữ GitHub.¹

1 Giới thiệu

1.1 Tổng quan

Máy chủ Proxy

Máy chủ Proxy—hay là web cache—là một thực thể mạng có nhiệm vụ đáp ứng các yêu cầu kết nối tới máy chủ web. Trình duyệt được cấu hình để gửi tất cả yêu cầu thông qua Proxy. Proxy thực hiện kết nối tới máy chủ, là trung gian gửi và nhận dữ liệu giữa trình duyệt (khách hay client) và máy chủ (server).

Proxy có vai trò vừa là trình chủ vừa là trình khách. Trình chủ được thể hiện ở chỗ Proxy thay mặt xử lý các yêu cầu từ trình duyệt. Trình khách được thể hiện ở Proxy kết nối đến máy chủ, gửi yêu cầu nhận được từ trình duyệt đến máy chủ và nhận kết quả.

Proxy có nhiệm vụ quản lý số lượng kết nối, lưu trữ (caching), và kiểm soát truy cập.

Quản lý lượng kết nối Proxy giới hạn số lượng xử lý đồng thời để tránh trường hợp quá tải bộ nhớ và bộ xử lý. Điều này giúp máy tính tránh bị tình trạng treo, cũng như giảm lưu lượng trên đường mạng.

Lưu trữ (caching) Proxy lưu những bản sao kết quả của các yêu cầu trước đó. Giúp việc xử lý yêu cầu từ trình duyệt được nhanh hơn.

Kiểm soát truy cập Proxy có thể lọc và chặn các kết nối tới máy chủ không an toàn hoặc không mong muốn.

Socket

Ứng dụng sử dụng socket hướng kết nối và socket không hướng kết nối. Dựa trên giao thức TCP, việc truyền dữ liệu chỉ thực hiện giữa hai quá trình đã thiết lập kết nối. Giao thức này đảm bảo dữ liệu được truyền đến nơi nhận một cách đáng tin cậy, đúng thứ tự nhờ vào cơ chế quản lý luồng lưu thông trên mạng và cơ chế chống tắc nghẽn. Đồng thời, mỗi thông điệp gửi phải có xác nhận trả về và các gói tin chuyển đi tuần tự.

1.2 Mục tiêu của đồ án môn học

Mục tiêu của đồ án môn học mạng máy tính là xây dựng một **Máy chủ Proxy** với các chức năng chính sau:

- Tiếp nhận và chuyển tiếp các *request* từ trình duyệt (trình duyệt) đến máy chủ đích.
- Chuyển tiếp thông tin phản hồi từ máy chủ đích đến trình duyệt.

¹Xem chi tiết tại [1].

- Xử lý và phân tích các *request* và *response* trong quá trình truyền tải.
- Hỗ trợ giao thức **HTTP** và **HTTPS**.
- Đọc và xử lý được thông tin mã hóa của trình duyệt đối với các yêu cầu kết nối HTTPS.
- Chặn các kết nối không mong muốn của trình duyệt đến máy chủ đích.

Thông qua quá trình triển khai, nhóm đã nắm vững các nguyên lý hoạt động của máy chủ Proxy, giao thức mạng và lập trình socket, đồng thời hiểu rõ hơn về cách xử lý dữ liệu mã hóa trong kết nối bảo mật HTTPS.

1.3 Phạm vi thực hiện

Trong khuôn khổ đồ án môn học này, nhóm thực hiện xây dựng một máy chủ Proxy hoạt động trên hệ điều hành Windows, nhằm xử lý các yêu cầu kết nối và truyền tải dữ liệu giữa trình duyệt và máy chủ. Để hiện thực hóa các chức năng trên, nhóm sử dụng ngôn ngữ lập trình C++, kết hợp với thư viện OpenSSL để đảm bảo các yêu cầu về bảo mật, cùng với API của Windows để triển khai các chức năng liên quan đến kết nối và xử lý dữ liệu.

2 Phân tích và thiết kế hệ thống

2.1 Yêu cầu chức năng

Với mục tiêu đã được đề cập ở mục trước, chúng ta có các chức năng cho máy chủ proxy như sau:

Chức năng chính:

- Lắng nghe và nhận *request* từ client.
- Chuyển tiếp *request* đến máy chủ đích.
- Nhận *response* từ máy chủ đích và trả lại cho tiến trình khách - trình duyệt.
- Chặn các kết nối không mong muốn từ tiến trình khách đến máy chủ đích.

Chức năng bổ sung:

- Hỗ trợ kết nối **HTTPS** thông qua mã hóa **SSL/TLS** để kiểm soát truy cập.
- Ghi *log* thông tin *request* và *response* và ghi rõ các lỗi xuất hiện trong khi vận hành máy chủ Proxy để dễ dàng kiểm tra và sửa lỗi.
- Xử lý nhiều kết nối đồng thời và chia luồng gửi và nhận dữ liệu để đảm bảo tốc độ truyền tải.
- Xây dựng hệ thống duy trì và kiểm soát luồng để không làm cho máy tính quá tải.

2.2 Luồng hoạt động của hệ thống

2.2.1 Máy chủ Transparent Proxy

Luồng hoạt động của hệ thống được mô tả chi tiết như sau:

1. Máy chủ Proxy hoạt động trên một máy và lắng nghe các kết nối từ tiến trình khách trên một địa chỉ IP và cổng đã chỉ định.
2. Khi tiến trình khách gửi *request*, máy chủ Proxy nhận và xử lý. Có hai trường hợp có thể xảy ra như sau:
 - **Kết nối HTTP (port 80):** máy chủ Proxy đọc trực tiếp dữ liệu từ trình duyệt vì thông tin không bị mã hóa.
 - **Kết nối HTTPS (port 443):** trình duyệt gửi yêu cầu sử dụng phương thức CONNECT để thiết lập một kết nối “tunnel” đến máy chủ đích. Lúc này, máy chủ Proxy chỉ chuyển tiếp các thông tin đã được mã hóa giữa tiến trình khách và máy chủ đích.
3. Máy chủ Proxy thiết lập kết nối đến máy chủ đích và chuyển tiếp *request*.
4. Máy chủ đích phản hồi lại *response*.
5. Máy chủ Proxy nhận *response* từ máy chủ đích và gửi lại cho trình duyệt.
6. Máy chủ Proxy ghi lại log thông tin *request* và *response* (nếu cần thiết).
7. Hệ thống đảm bảo xử lý được nhiều kết nối đồng thời một cách hiệu quả.

Hình 1 mô tả ý tưởng chính của máy chủ Transparent Proxy. Với thiết kế này, máy chủ Proxy có khả năng nhận và chuyển tiếp dữ liệu giữa tiến trình khách (trình duyệt) và máy chủ đích. Khi giao tiếp sử dụng giao thức HTTP, máy chủ Proxy có thể đọc và xử lý trực tiếp dữ liệu truyền giữa trình duyệt web và máy chủ đích, vì dữ liệu không được mã hóa.

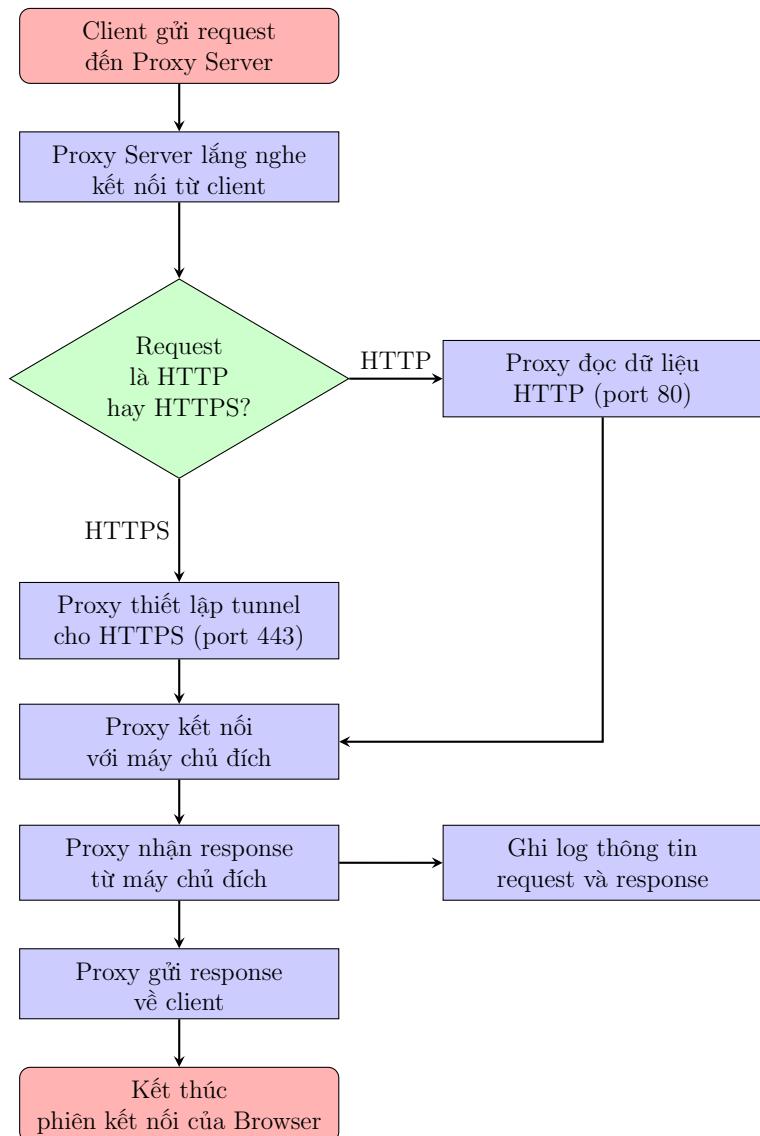
Tuy nhiên, với giao thức HTTPS, máy chủ Proxy phải thiết lập một đường hầm (tunnel) giữa trình duyệt và máy chủ đích thông qua phương thức CONNECT. Trong trường hợp này, máy chủ Proxy chỉ chuyển tiếp dữ liệu đã được mã hóa mà không thể đọc hoặc chỉnh sửa nội dung truyền đi.

Để có thể đọc được dữ liệu truyền giữa trình duyệt và máy chủ khi sử dụng HTTPS, máy chủ Proxy cần hoạt động như một *Man-in-the-Middle Proxy*.

2.2.2 Máy chủ Man-in-the-Middle Proxy (MITM)

Với ý tưởng Man-in-the-Middle Proxy, máy chủ Proxy sẽ đóng vai trò như một máy chủ trung gian, nhận và gửi dữ liệu giữa trình duyệt và máy chủ đích.

- Proxy giả lập thành máy chủ đích đối với trình duyệt và nhận các request từ trình duyệt. Lưu ý là lúc này máy chủ Proxy sẽ nhận những thông tin header giống với thông tin mà trình duyệt gửi đến máy chủ đích.
- Đồng thời, Proxy giả lập thành trình duyệt đối với máy chủ đích để chuyển tiếp request và nhận phản hồi từ máy chủ đích.

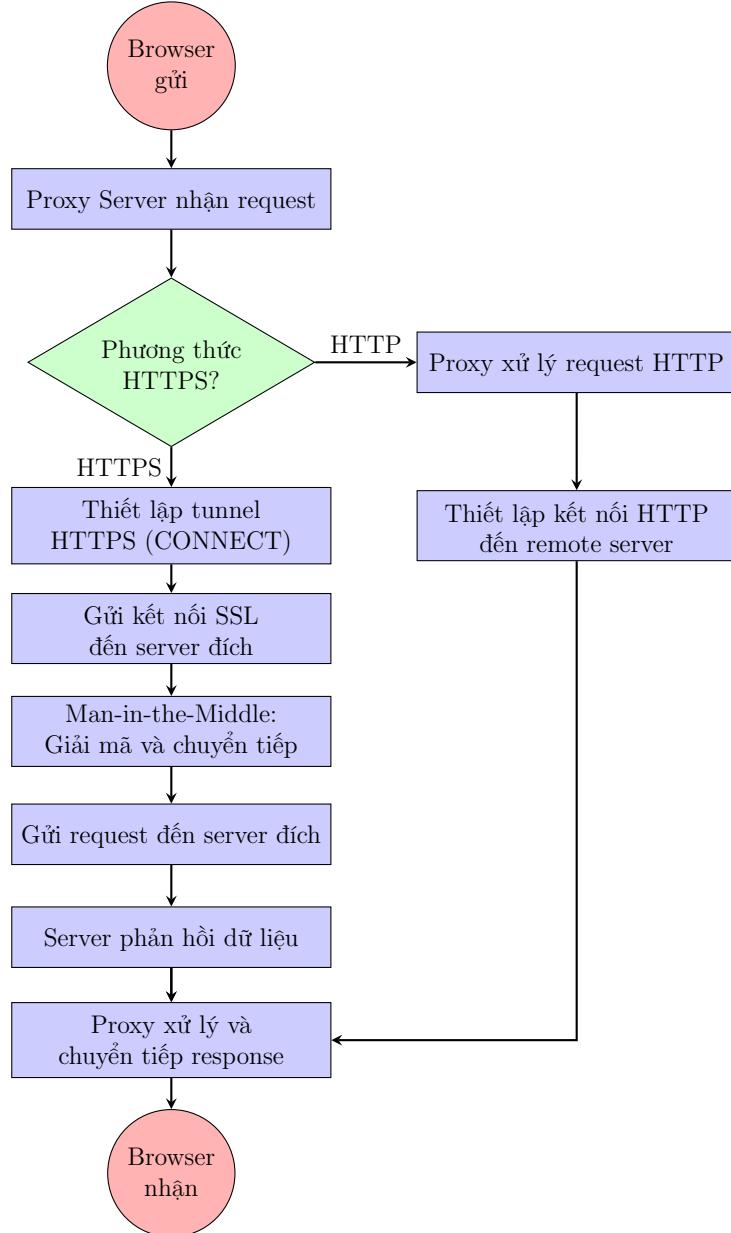


Hình 1: Sơ đồ khái ý tưởng của máy chủ Transparent Proxy

- Nhờ đó, máy chủ Proxy có thể giải mã và đọc dữ liệu trước khi tiếp tục mã hóa và gửi dữ liệu đến trình duyệt hoặc là máy chủ đích.

Ý tưởng này cho phép máy chủ Proxy can thiệp vào luồng dữ liệu mã hóa giữa trình duyệt và máy chủ, từ đó đọc, ghi log hoặc xử lý dữ liệu khi cần thiết. Đồng thời, với cách làm này giúp phát triển ứng dụng để hiện thực hóa tính năng caching của Proxy.

Sơ đồ thể hiện ý tưởng thiết kế Man-in-the-Middle Proxy được mô tả ở hình 2.



Hình 2: Sơ đồ ý tưởng Man-in-the-Middle Proxy Server

Dựa trên ý tưởng trên, máy chủ Proxy có thể gửi và nhận các yêu cầu (request) cũng như phản hồi (response) từ HTTP và HTTPS. Tuy nhiên, để đảm bảo hiệu suất cao, máy chủ cần quản lý việc xử lý các yêu cầu trên nhiều luồng song song. Điều này giúp tối ưu hóa tốc độ xử lý và duy trì sự kết nối ổn định giữa trình duyệt và máy chủ đích.

Một vấn đề quan trọng cần giải quyết là việc kiểm soát tài nguyên hệ thống. Nếu Proxy

tạo một tiến trình mới cho mỗi yêu cầu từ trình duyệt, tài nguyên hệ thống sẽ nhanh chóng cạn kiệt, đặc biệt khi số lượng yêu cầu tăng cao. Vì vậy, việc triển khai cần phải bao gồm một cơ chế giới hạn số lượng tiến trình hoặc luồng tối đa được tạo ra, đảm bảo máy chủ Proxy duy trì hiệu suất ổn định và thời gian phản hồi tối ưu trong mọi điều kiện hoạt động. Vấn đề này sẽ được trình bày ở phần sau.

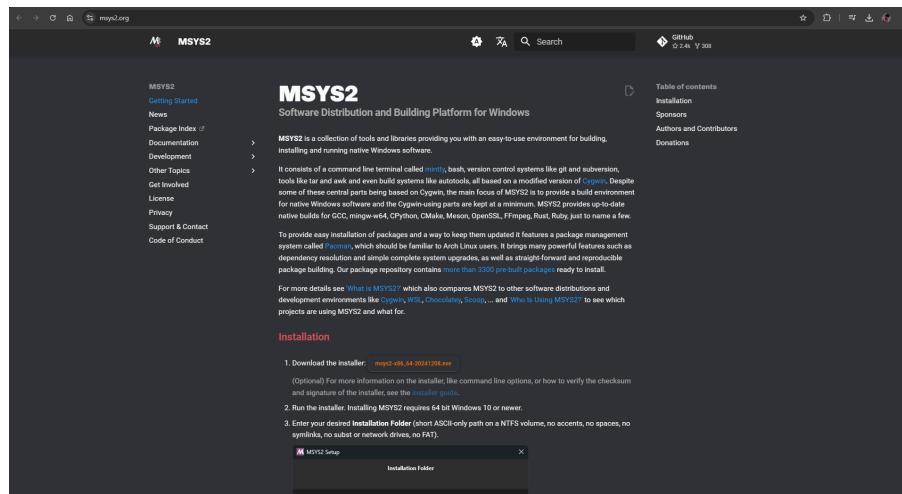
2.3 Công nghệ sử dụng

- **Ngôn ngữ lập trình:** C++.
- **Thư viện:**
 - **OpenSSL:** Xử lý mã hóa SSL/TLS cho kết nối HTTPS.
 - **Windows API:** Lập trình socket để lắng nghe và chuyển tiếp dữ liệu.
- **Công cụ phát triển:**
 - **Visual Studio Code:** Môi trường viết và phát triển mã nguồn.
 - **Complier g++:** Công cụ biên dịch mã nguồn. Với mã nguồn này cần được biên dịch bằng bộ dịch được tải từ MSYS2 để tương thích với một số header mà trên Mingw không hỗ trợ.
 - **Make và Makefile:** Hỗ trợ xây dựng và biên dịch chương trình.
 - **Inno Setup:** Đóng gói và tạo trình cài đặt phần mềm.

2.4 Cài đặt công cụ

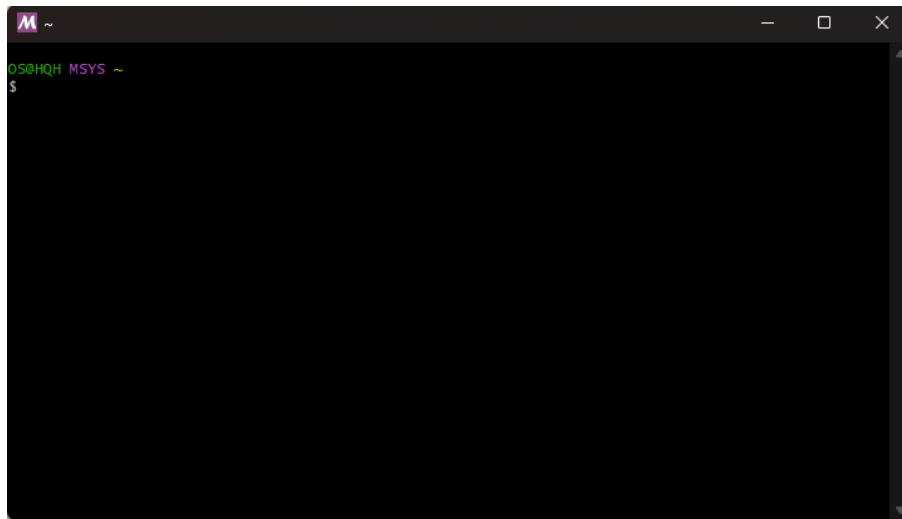
Cài đặt trình quản lý gói MSYS2

Trong đồ án này, chương trình dịch được cài đặt thông qua trình quản lý gói MSYS2. Bộ cài đặt MSYS2 có thể được tải từ trang web chính thức như minh họa trong hình 3.



Hình 3: Trang web dự án MSYS2

Sau khi cài đặt, cần thiết lập biến môi trường để trỏ đến thư mục `C:/msys64/usr/bin`. Tiếp theo, khởi động chương trình MSYS2. Giao diện dòng lệnh MSYS2 được minh họa trong hình 4.



Hình 4: Cửa sổ console của MSYS2

Cài đặt chương trình dịch g++

Bộ dịch g++ trong trình quản lý gói MSYS2 được sử dụng để biên dịch mã nguồn của đồ án. Quy trình cài đặt g++ được trình bày trong hình 5.

```
OS@HQH MSYS ~
$ pacman -S mingw-w64-x86_64-gcc
resolving dependencies...
looking for conflicting packages...

Packages (2) mingw-w64-x86_64-gcc-libs-14.2.0-2 mingw-w64-x86_64-gcc-14.2.0-2

Total Download Size: 44.50 MiB
Total Installed Size: 228.00 MiB
Net Upgrade Size: 74.80 MiB

:: Proceed with installation? [Y/n] |
```

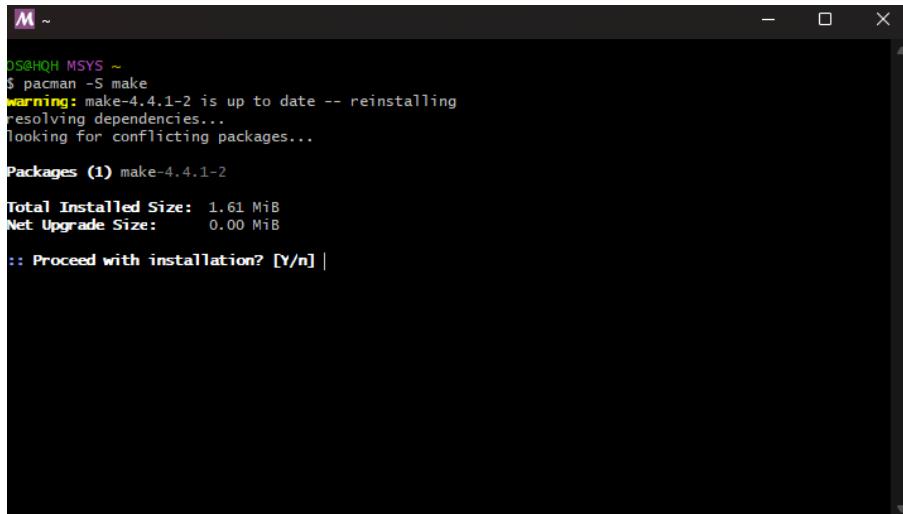
Hình 5: Cài đặt chương trình dịch g++

Cài đặt công cụ hỗ trợ dịch mã nguồn

Để hỗ trợ việc biên dịch mã nguồn dễ dàng và thuận tiện hơn, công cụ `make` cũng được cài đặt thông qua MSYS2. Hình 6 minh họa các bước cài đặt công cụ này.

Cài đặt thư viện ngoài

Để triển khai các giao thức bảo mật như SSL (Secure Sockets Layer) và TLS (Transport Layer Security), nhóm thực hiện đã sử dụng thư viện mã nguồn mở OpenSSL. Trang web tải bộ cài đặt OpenSSL được minh họa trong hình 7.



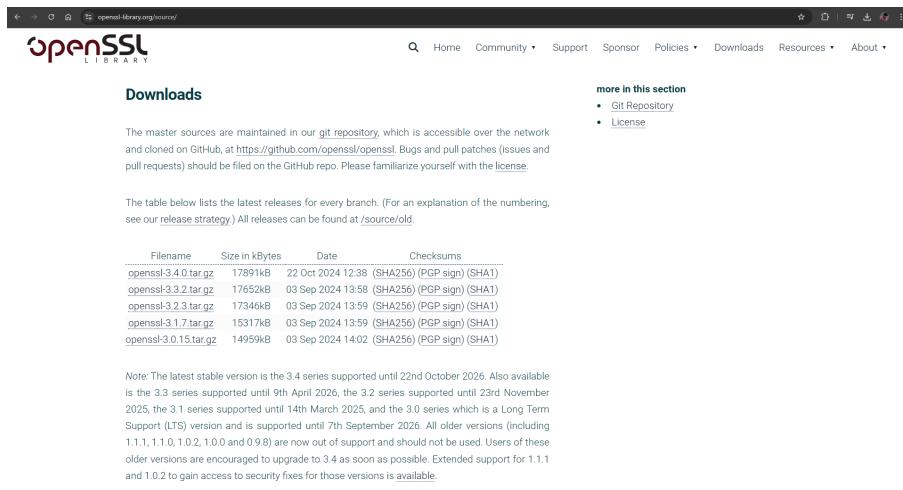
```
05@HQH MSYS ~
$ pacman -S make
warning: make-4.4.1-2 is up to date -- reinstalling
resolving dependencies...
looking for conflicting packages...

Packages (1) make-4.4.1-2

Total Installed Size: 1.61 MiB
Net Upgrade Size: 0.00 MiB

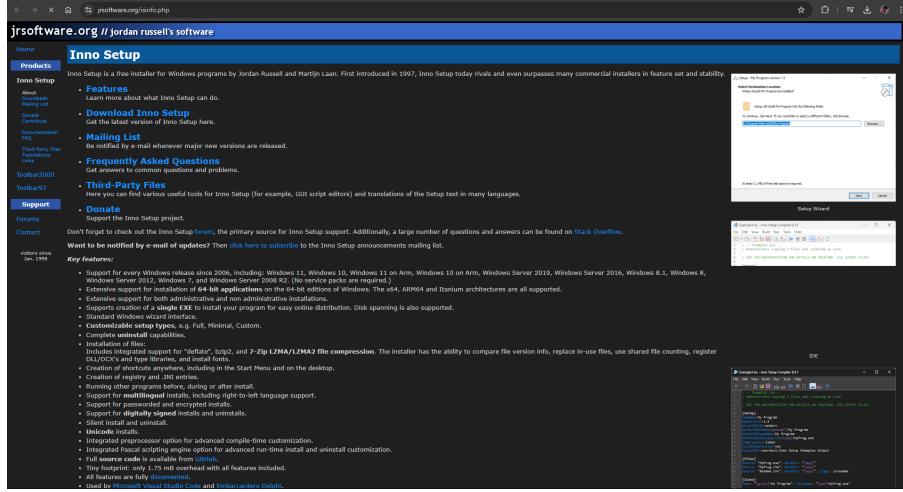
:: Proceed with installation? [Y/n] |
```

Hình 6: Cài đặt make



Hình 7: Trang web cài đặt thư viện OpenSSL

Công cụ khác



Hình 8: Cài đặt chương trình đóng gói phần mềm

3 Cấu trúc mã nguồn và cài đặt

3.1 Cấu trúc mã nguồn

Phần này trình bày cấu trúc tổ chức mã nguồn của đồ án Proxy Server. Mã nguồn được chia thành các thành phần chính như sau:

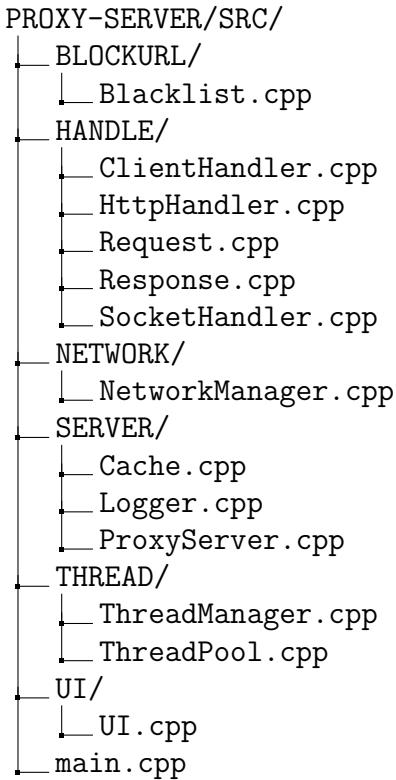
- **BLOCKURL:** Xử lý chức năng chặn các URL không mong muốn, quản lý danh sách đen.
- **HANDLE:** Chứa các lớp xử lý liên quan đến đản bảo kết nối - (SSL) socket - và xử lý các yêu cầu từ trình duyệt cũng như các phản hồi từ máy chủ đích.
- **NETWORK:** Khởi tạo và quản lý các kết nối socket của máy chủ Proxy, đảm bảo thông nhất phiên bản các API window liên quan đến kết nối mạng trên toàn dự án.
- **SERVER:** Xử lý thiết lập server và quản lý logger để ghi log thông tin.
- **THREAD:** Xử lý vấn đề đa luồng để quản lý nhiều kết nối đồng thời, đảm bảo tối ưu tài nguyên hệ thống.
- **UI:** Quản lý giao diện người dùng của Proxy Server, giúp thao tác thân thiện và dễ sử dụng.

Hình 9 thể hiện cấu trúc thư mục mã nguồn của đồ án.

3.1.1 Phân tích các lớp chính

Lớp NetworkManager

Lớp **NetworkManager** chứa thông tin về việc khởi tạo winsock và địa chỉ IP. Ngoài ra **NetworkManager** còn có phương thức cho việc gửi và nhận dữ liệu từ socket.



Hình 9: Cấu trúc mã nguồn đồ án

3.1.2 Lớp ProxyServer

Lớp `ProxyServer` kế thừa từ lớp `NetworkManager`. Thuộc tính chính của `ProxyServer` là `type`, `port` và `localSocket`. Trong đó, biến `type` cho sẽ quyết định loại Proxy được chọn là Man-in-the-Middle hay là Transparent Proxy, biến `localSocket` sẽ kết nối với `port` (cổng) để có thể nhận kết nối từ client qua cổng đó. Hai phương thức chính của lớp này là `waitingClient` và `acceptClient` để lắng nghe và chấp nhận kết nối từ client.

3.1.3 Lớp ClientHandler

Khi proxy chấp nhận kết nối từ client. Đối tượng `ClientHandler` được tạo ra. Chức năng của đối tượng này là thiết lập kết nối đến máy chủ đích mà trình duyệt muốn và xử lý việc gửi nhận thông tin giữa tiến trình khách và máy chủ đích.

3.1.4 Lớp SocketHandler

Lớp `SocketHandler` tạo và quản lý kết nối socket. Khi một đối tượng `ClientHandler` được tạo ra, `socket` trao đổi dữ liệu giữa trình duyệt và proxy sẽ được truyền vào `SocketHandler`. Sau đó, `SocketHandler` sẽ đảm nhiệm việc đọc thông tin yêu cầu kết nối và kết nối đến với máy chủ đích. Nếu yêu cầu kết nối là HTTP thì sẽ được kết nối trực tiếp đến máy chủ đích, còn nếu là yêu cầu kết nối CONNECT để truyền dữ liệu HTTPS thì tùy vào loại proxy được chọn, nó sẽ tiến hành thiết lập SSL hay không.

3.1.5 Lớp ThreadPool

Lớp `ThreadPool` tạo và quản lý một số lượng luồng (thread) nhất định. Khi một đối tượng `ClientHandler` được tạo ra, công việc xử lý của những yêu cầu tiến trình khách sẽ được đẩy vào danh sách công việc và chờ một trong các luồng xử lý. Ngoài ra, luồng này sẽ nhận thêm các công việc như gửi và nhận các thông tin tiến trình khách.

3.1.6 Lớp PUI

Lớp `PUI` quản lý và xử lý các thao tác trên cửa sổ giao diện đồ họa (UI). Cửa sổ này sẽ cung cấp cho người dùng một số thao tác như:

- **start:** khởi chạy máy chủ Proxy.
- **stop:** tạm dừng chạy máy chủ Proxy.
- **mode:** cung cấp cho người dùng 2 loại proxy là *Man-in-the-Middle Proxy* và *Transparent Proxy* để chọn lựa và sử dụng.
- **blacklist:** để người dùng thêm các trang web cần bị chặn.
- **log:** hiển thị các lỗi để người quan trị dễ dàng theo dõi.

3.1.7 Các lớp khác

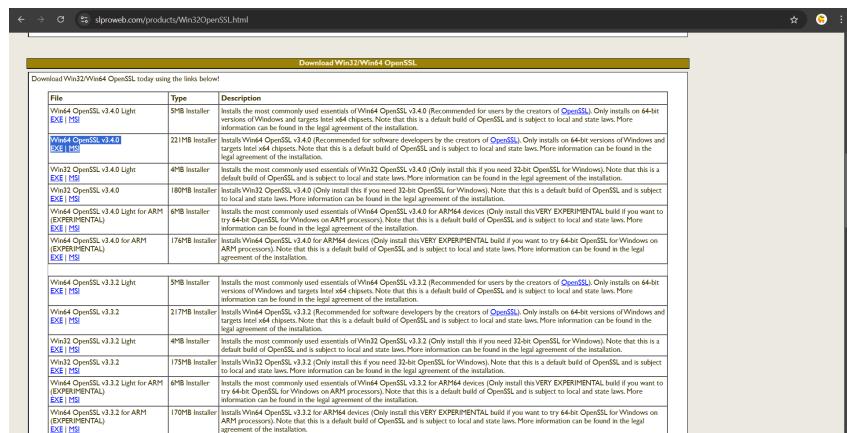
Ngoài các lớp chính được mô tả ở trên, đồ án còn có thêm các lớp như: `HTTPHandler`, `RequestHandler`, `ResponseHandler`, nhằm để xử lý request và response giữa trình duyệt và máy chủ đích một cách chính xác, những class này có tác dụng sẽ phân tích gói HTTP và lấy các thông tin từ header và body.

3.2 Cách biên dịch và chạy chương trình

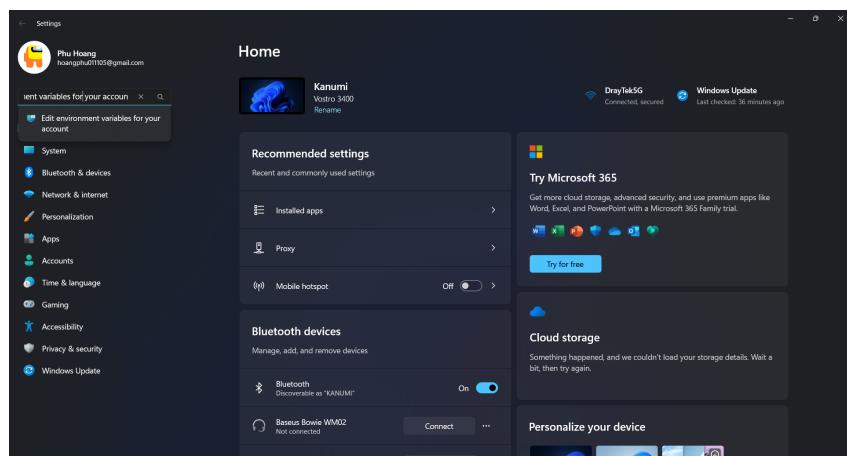
Để thực hiện quá trình dịch chương trình thành file chương trình chạy, các bước sau đây cần được tiến hành theo thứ tự:

3.2.1 Cài đặt OpenSSL và thiết lập thư viện:

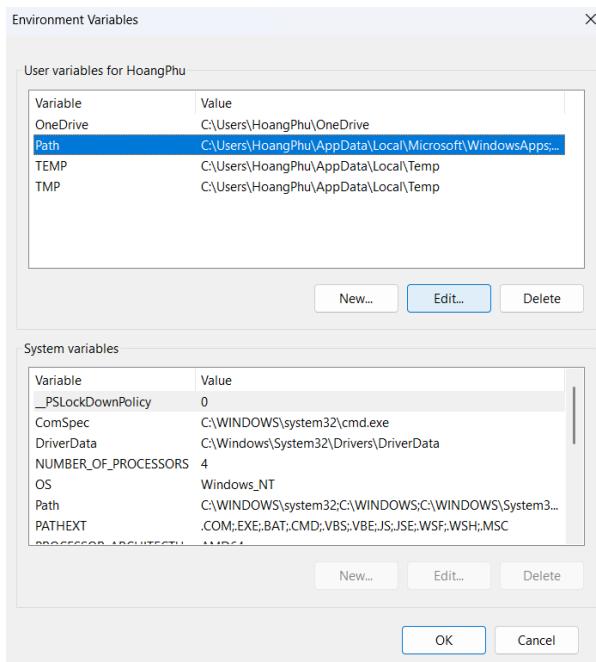
- Tải và cài đặt OpenSSL phiên bản phù hợp với hệ điều hành (có thể tải tại [đường dẫn này](#), hình 10). Sau đó cài đặt OpenSSL vào máy. Mở Settings, tìm kiếm và chọn “Edit environment variables for your account” (hình 11). Tại cửa sổ mới Environment Variables, tìm biến Path và chọn Edit (hình 12). Tại cửa sổ Edit environment variables, chọn New và thêm đường dẫn thư mục bin—nếu trong lúc cài đặt có chọn tùy chọn tạo thư mục bin cho các tệp dll—của thư viện OpenSSL (hình 13).
- Đảm bảo liên kết các file thư viện như `applink.c`, `libssl.lib` và `libcrypto.lib` trong Visual Studio Code để có thể kiểm lỗi dễ dàng hơn tránh trường hợp bộ dịch và chương trình kiểm lỗi không giống nhau.
- Sửa lại đường dẫn của OpenSSL trong Makefile để có thể chương trình chính xác.



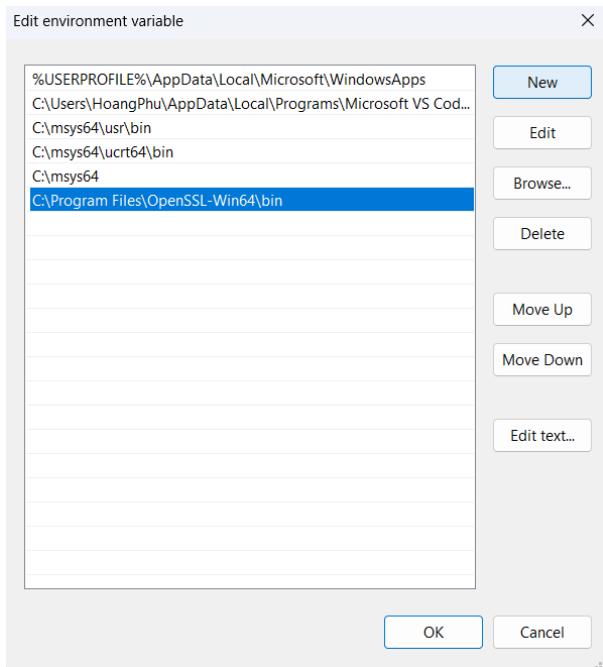
Hình 10: Trang web tải thư viện Openssl



Hình 11: Tìm kiếm Edit environment variables for your account



Hình 12: Cửa sổ Environment Variables



Hình 13: Cửa sổ Edit environment variables

3.2.2 Dịch chương trình

- Cần cài đặt make và bộ dịch g++ để tiến hành dịch mã nguồn thành tệp proxy.exe. (Lưu ý: một bộ dịch g++ từ MSYS2 có khả năng là cần thiết vì nếu bộ dịch không tương thích có khả năng không dịch được chương trình.) Chi tiết cài đặt đã được trình bày ở trên.
- Sử dụng file makefile đã được đính kèm trong sourcode để biên dịch chương trình thành tệp proxy.exe.

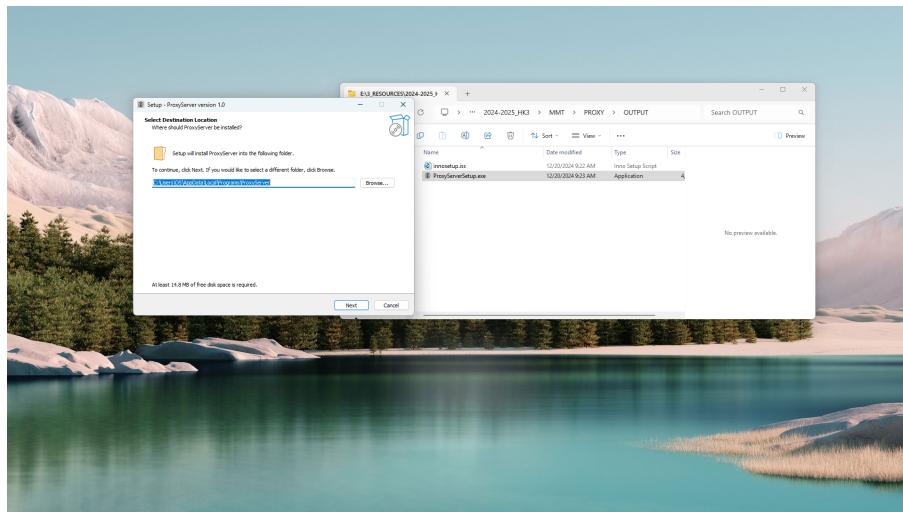
3.2.3 Chạy chương trình và thử nghiệm:

- Chạy file thực thi proxy.exe trên terminal hoặc cmd.
- Làm theo hướng dẫn để thiết lập chứng chỉ đối với việc sử dụng Man-in-the-Middle Proxy.
- Cấu hình trình duyệt (như Firefox, Chrome) để sử dụng địa chỉ IP và cổng của Proxy Server.
- Thử nghiệm gửi các request HTTP và HTTPS qua trình duyệt.

3.2.4 Kiểm tra:

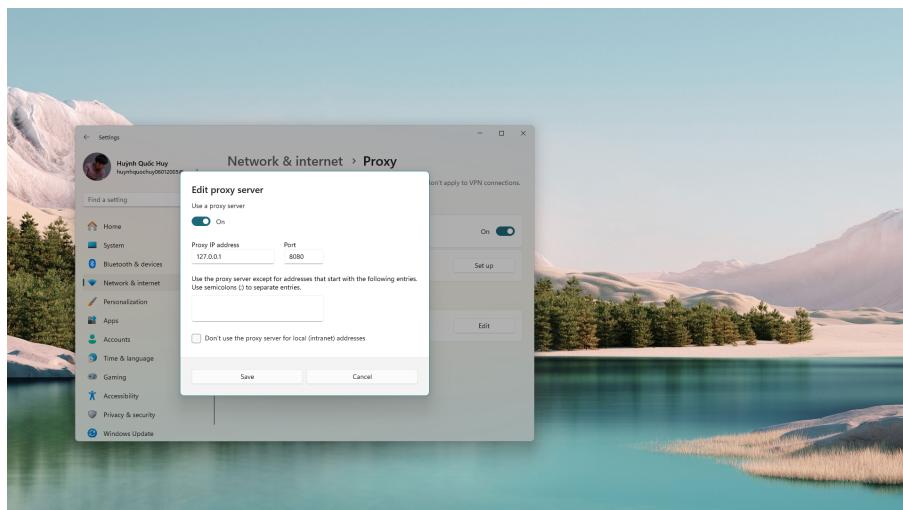
Ngoài phương pháp dịch lại chương trình để chạy, nhóm còn cung cấp file cài đặt ProxySetup.exe để có thể cài đặt dễ dàng hơn. Chương trình này sẽ tự động cài đặt vào máy, người dùng chỉ cần mở lên và sử dụng.

- Chạy tệp cài đặt ProxySetup.exe trên terminal hoặc cmd. (Hình 14)
- Làm theo hướng dẫn trên cửa sổ giao diện.



Hình 14: Cài đặt Proxy lên máy tính

- Thiết lập ProxyServer trên máy tính. (Hình 15)



Hình 15: Thiết lập ProxyServer trên máy tính

- Mở ProxyServer trên máy và chạy thử. (Hình 16)

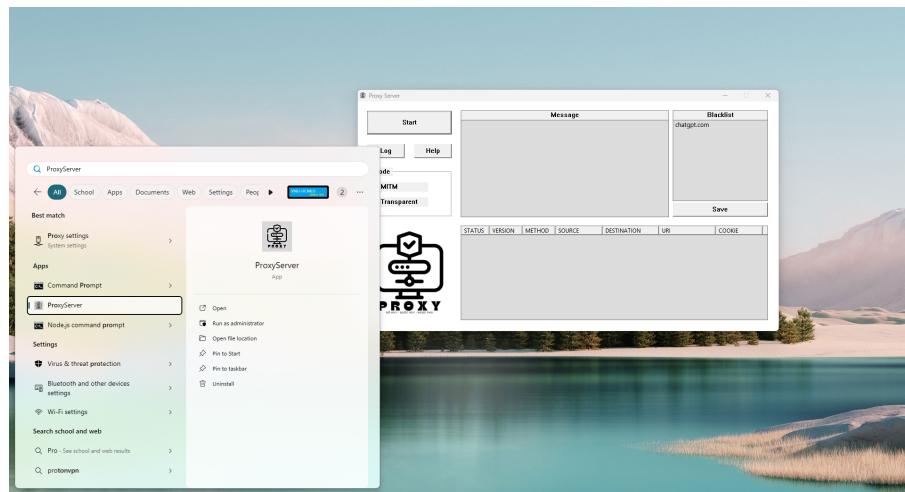
Với các bước làm trên, chúng ta có thể sử dụng được chế độ transparent vì đây là chế độ máy chủ Proxy chỉ nhận và chuyển tiếp các request và response và không đọc được thông tin được mã hóa ở dưới. Để sử dụng được chế độ MITM, chúng ta phải cài chứng chỉ tự ký cho máy.

3.2.5 Cài đặt chứng chỉ lên máy

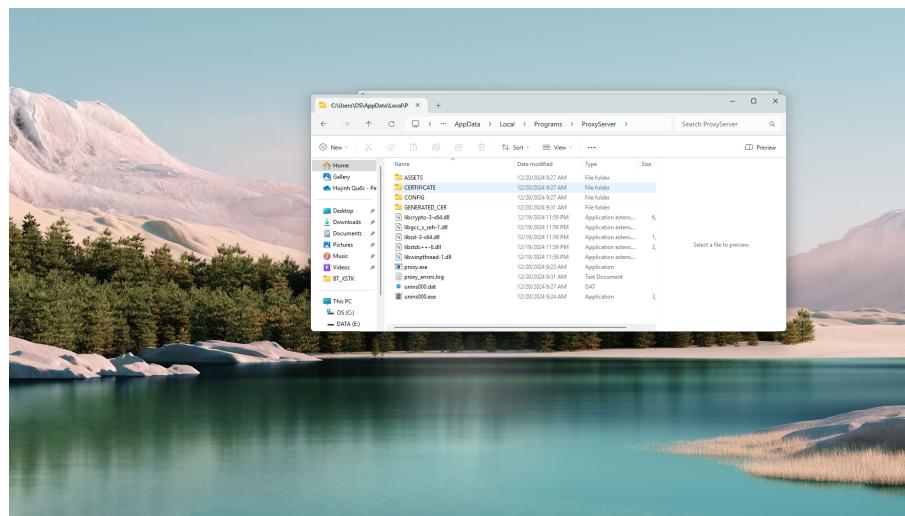
Bước 1: Tìm thư mục chứa chứng chỉ CERTIFICATE trong thư mục cài đặt của hệ thống như hình 17.

Bước 2: Tìm vị trí để cài đặt chứng chỉ tự ký vào hệ thống của máy như hình 18.

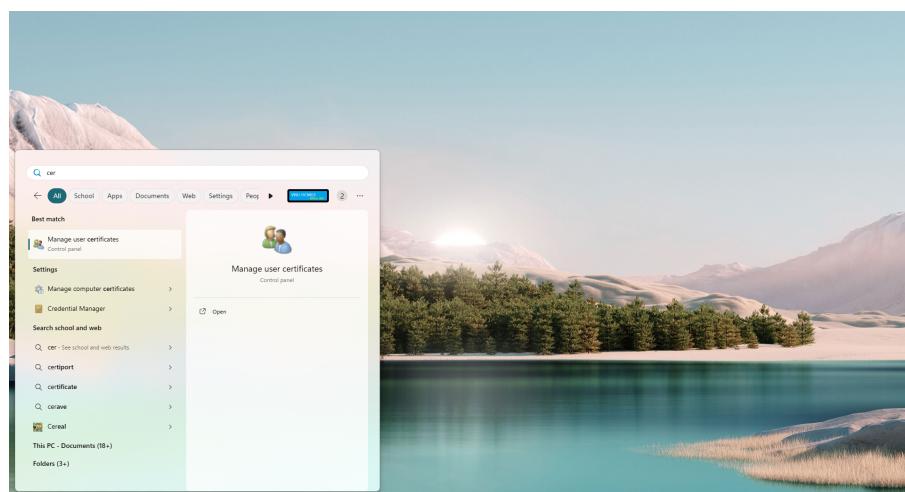
Bước 3: Cài đặt chứng chỉ tự ký vào hệ thống của máy như hình 19.



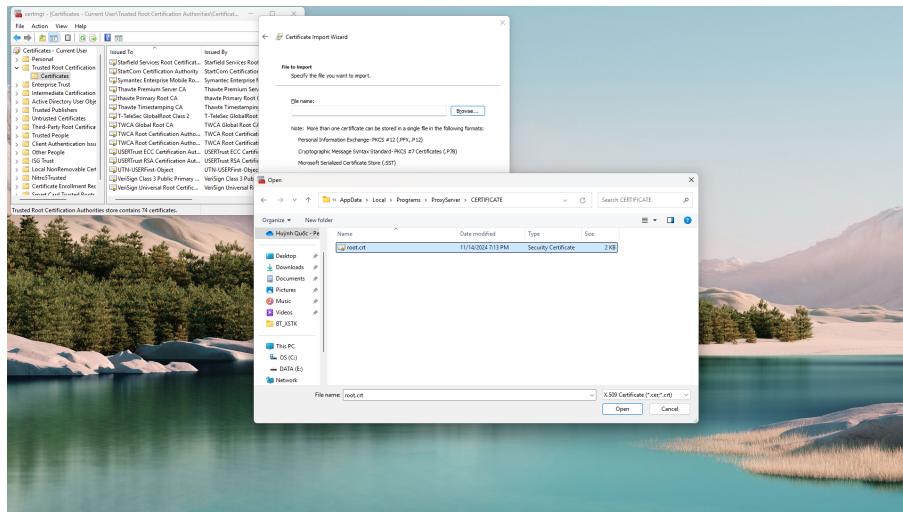
Hình 16: Mở ProxyServer



Hình 17: Tìm thư mục chứa chứng chỉ



Hình 18: Tìm vị trí cài đặt chứng chỉ vào hệ thống



Hình 19: Cài đặt chứng chỉ vào hệ thống

4 Kết quả đạt được

4.1 Những chức năng đã hoàn thành:

- Proxy Server đã nhận và chuyển tiếp các request từ client đến server.
- Hỗ trợ giao thức HTTP và HTTPS.
- Đã thực hiện thành công Man-in-the-Middle, Proxy có khả năng giải mã các thông tin giao tiếp giữa máy chủ đích và trình duyệt web kết nối vào Proxy Server.
- Đã ghi lại được các loại lỗi phát sinh trong quá trình chạy server bằng file logger.
- Xử lý được đồng thời nhiều kết nối đồng thời.

4.2 Hình ảnh minh họa:

Cho phép truy cập

Hình 20 mô tả trường hợp cho phép truy cập vào chatgpt.com.

Không phép truy cập đối với chế độ transparent

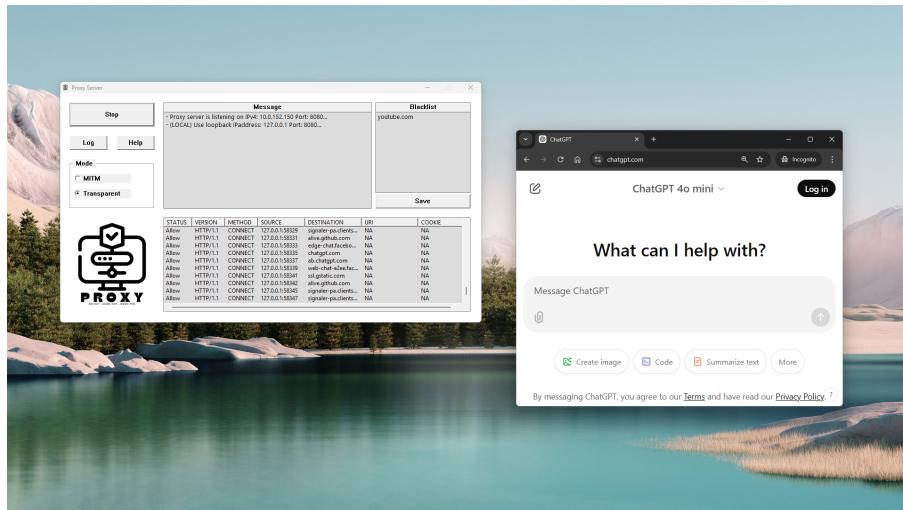
Hình 21 mô tả trường hợp chặn truy cập vào chatgpt.com với mode transparent.

Không phép truy cập đối với chế độ MITM

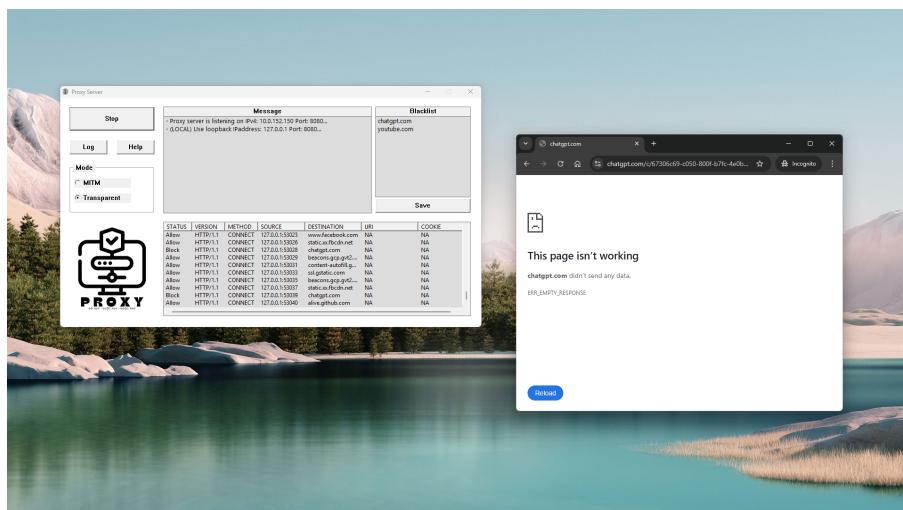
Hình 22 mô tả trường hợp chặn truy cập vào chatgpt.com với mode mitm.

5 Kết luận

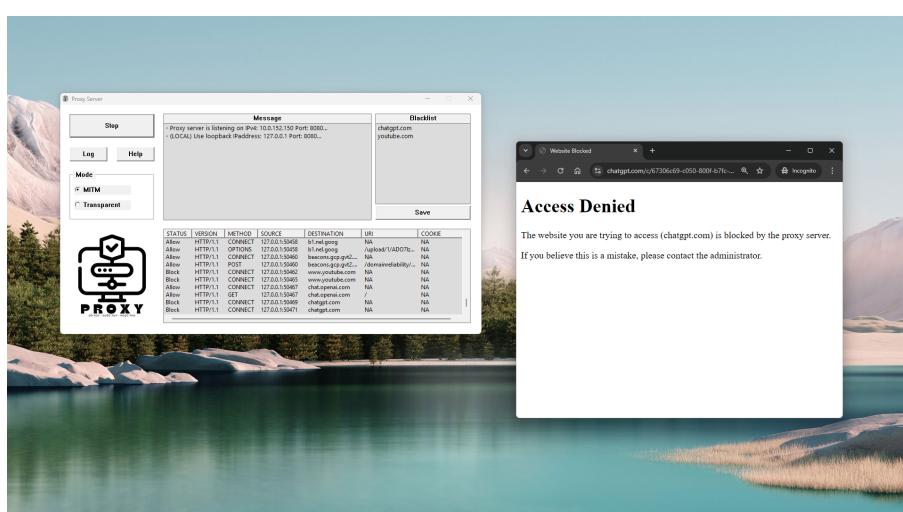
Trong đồ án này, nhóm chúng em đã xây dựng thành công một máy chủ Proxy cơ bản với các chức năng chính như tiếp nhận và chuyển tiếp request/response giữa tiến trình khách (trình duyệt) và máy chủ đích, hỗ trợ giao thức HTTP và HTTPS, cũng như xử



Hình 20: Cho phép truy cập chatgpt.com



Hình 21: Chặn truy cập của chatgpt.com trường hợp Transparent



Hình 22: Chặn truy cập của chatgpt.com trường hợp MITM

lý nhiều kết nối đồng thời. Qua quá trình thực hiện, nhóm đã áp dụng và củng cố kiến thức về lập trình socket, giao thức HTTP/HTTPS, và các khái niệm bảo mật cơ bản.

Bên cạnh đó, việc tích hợp các thư viện như OpenSSL để xử lý kết nối HTTPS và xây dựng hệ thống đa luồng thông qua ThreadPool đã giúp nhóm hiểu rõ hơn về việc tối ưu hóa tài nguyên và xử lý đồng thời trong một hệ thống mạng. Giao diện người dùng (UI) được thiết kế thân thiện, hỗ trợ quản lý hoạt động của Proxy Server một cách dễ dàng, cũng là một điểm nổi bật của đồ án.

Trong quá trình thực hiện, nhóm đã gặp một số khó khăn như xử lý mã hóa SSL/TLS, quản lý kết nối HTTPS dưới dạng Man-in-the-Middle Proxy, và đảm bảo tính ổn định khi xử lý đồng thời nhiều kết nối. Tuy nhiên, nhờ sự phối hợp của các thành viên trong nhóm và sự hỗ trợ từ tài liệu tham khảo, các vấn đề này đã được giải quyết một cách hợp lý.

Đồ án không chỉ giúp nhóm đạt được mục tiêu ban đầu mà còn mang lại nhiều bài học thực tiễn, đặc biệt là về cách thiết kế, xây dựng và triển khai một hệ thống mạng phức tạp. Đây là tiền đề quan trọng để nhóm tiếp tục nghiên cứu sâu hơn về kiến thức mạng, tối ưu hóa hệ thống, và ứng dụng các công nghệ hiện đại trong tương lai.

Nhóm nhận thấy rằng máy chủ Proxy được nhóm xây dựng vẫn còn một số hạn chế và có tiềm năng cải tiến, chẳng hạn như bổ sung chức năng lưu cache hiệu quả hơn, hỗ trợ thêm các giao thức khác hoặc triển khai cơ chế bảo mật nâng cao. Đây sẽ là định hướng để nhóm hoàn thiện hơn trong các nghiên cứu tiếp theo.

Nhóm xin gửi lời cảm ơn chân thành đến thầy vì đã tận tình hướng dẫn và tạo điều kiện thuận lợi để nhóm hoàn thành đồ án này. Trong quá trình thực hiện, mặc dù nhóm đã nỗ lực hết mình để đảm bảo chất lượng tốt nhất, nhưng vẫn có thể không tránh khỏi những sai sót trong quá trình kiểm thử. Nhóm rất mong nhận được sự góp ý từ thầy để hoàn thiện hơn trong tương lai.

6 Báo cáo công việc

Thành viên nhóm:

1. Huỳnh Quốc Huy, MSSV: 23120015. (Nhóm trưởng.)
2. Đỗ Trọng Huy, MSSV: 23120007.
3. Hoàng Ngọc Phú, MSSV: 23120010.

Bảng 1 mô tả phần trăm (%) thực hiện các công việc:

Công việc \ Thành viên	H. Q. Huy	D. T. Huy	H. N. Phú
Proxy server	90	10	
UI			100
Báo cáo	50		50
Video demo		100	

Bảng 1: Phân chia công việc

Tài liệu tham khảo

- [1] Kho GitHub của đồ án. Xem tại đây.
- [2] Tài liệu tham khảo của OpenSSL. Xem tại đây.
- [3] Mai Văn Cường, Trần Trung Dũng, Trần Hồng Ngọc, Lê Ngọc Sơn, Lê Giang Thanh, Trương Thị Mỹ Trang, and Dào Anh Tuấn. *Giáo trình Mạng máy tính*. Nhà xuất bản Khoa học và kỹ thuật, 2020.
- [4] Nguyễn Thành Sơn, Trần Thanh Hoàng, Lê Thị Minh Châu, and Nguyễn Trần Thị Vân. *Lập trình truyền thông LINUX*. Nhà xuất bản Đại học Quốc gia thành phố Hồ Chí Minh, 2016.