

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO PROJECT II
KỲ 2023.2

**Đề tài: Tìm hiểu về giao thức HTTP/HTTPS và
triển khai SSL/TLS cho HTTPS Server**

Họ và tên - MSSV

Trần Hoàng Sơn: 20210744

Mã lớp: 738757

GVHD: TS. Đỗ Tiến Dũng

Mục lục

I, Giới thiệu đề tài	3
II, Nội dung	3
1. Bài toán thực tế	3
2. Các khái niệm	3
2.1. Giao thức HTTP.....	3
2.2. Giao thức HTTPS.....	5
2.3. Sự khác nhau giữa HTTP và HTTPS	6
2.4. Tìm hiểu về SSL/TLS	6
2.4.1. Khái niệm chứng chỉ SSL/TLS.....	6
2.4.2. PKI	6
2.4.3. Cách hoạt động của TLS/SSL.....	7
3. Phân tích TLS handshake	8
4. Tự triển khai HTTP/HTTPS bằng cách tạo https server.	11
4.1. Tạo chứng chỉ cho server bằng OpenSSL.....	11
4.2. Cài đặt SSL/TLS cho server bằng OpenSSL	12
4.3. Cài đặt https_server để xử lý yêu cầu của client.....	12
4.4. Kết quả chạy server:	14

I, Giới thiệu đề tài

Giao thức HTTP/HTTPS đóng vai trò quan trọng trong việc truyền tải dữ liệu trên mạng internet. HTTP là viết tắt của Hypertext Transfer Protocol, là giao thức truyền tải siêu văn bản được sử dụng để truyền tải dữ liệu giữa máy chủ web và trình duyệt web. HTTPS là giao thức HTTP được tích hợp thêm lớp bảo mật SSL/TLS, giúp bảo mật dữ liệu truyền tải giữa máy chủ và trình duyệt.

Bài báo cáo này sẽ trình bày về các vấn đề sau:

- Định nghĩa và cách thức hoạt động của 2 giao thức.
- Phân tích sự khác nhau của 2 giao thức.
- Lí do phải thay thế HTTP bằng HTTPS.
- Tìm hiểu về chứng chỉ SSL/TLS và cách hoạt động của SSL/TLS.
- Hướng dẫn cách sử dụng và triển khai HTTP/HTTPS.

Phương pháp:

- Đối với việc tìm hiểu hoạt động của SSL/TLS: Sử dụng Wireshark để bắt các gói tin, từ đó phân tích và chỉ ra cách Server và Client thiết lập kênh truyền an toàn.
- Đối với việc triển khai HTTP/HTTPS: Xây dựng 1 https server có cài đặt SSL/TLS bằng OpenSSL.

II, Nội dung

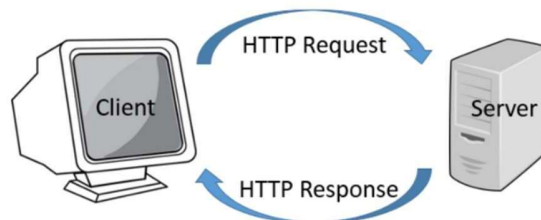
1. Bài toán thực tế

Trong thời đại của kỷ nguyên số, công nghệ hiện đại, các trang web cần đáp ứng những tiêu chuẩn bảo mật ngày càng nghiêm ngặt hơn, nhằm bảo mật thông tin người dùng và các dữ liệu quan trọng. Đối với giao thức HTTP, các thông tin được truyền qua giao thức này không hề được mã hóa và bảo mật. Đây chính là kẽ hở để những hacker dễ lợi dụng để đánh cắp thông tin người dùng. Một hình thức tấn công thường gặp là tấn công man-in-the-middle. Để khắc phục điều đó, giao thức HTTPS đã được tạo ra, nhằm bảo mật dữ liệu truyền tải, giúp ngăn chặn các hành vi tấn công và giả mạo. Việc thay đổi từ HTTP sang HTTPS là rất quan trọng để bảo mật dữ liệu, tăng cường sự tin tưởng của người dùng và cải thiện hiệu suất website.

2. Các khái niệm

2.1. Giao thức HTTP

HTTP (Hypertext Transfer Protocol): giao thức truyền tải siêu văn bản. HTTP là giao thức tiêu chuẩn cho www (World Wide Web) để truyền tải dữ liệu dưới dạng văn bản, hình ảnh, video, âm thanh từ web server đến trình duyệt web của người dùng và ngược lại.

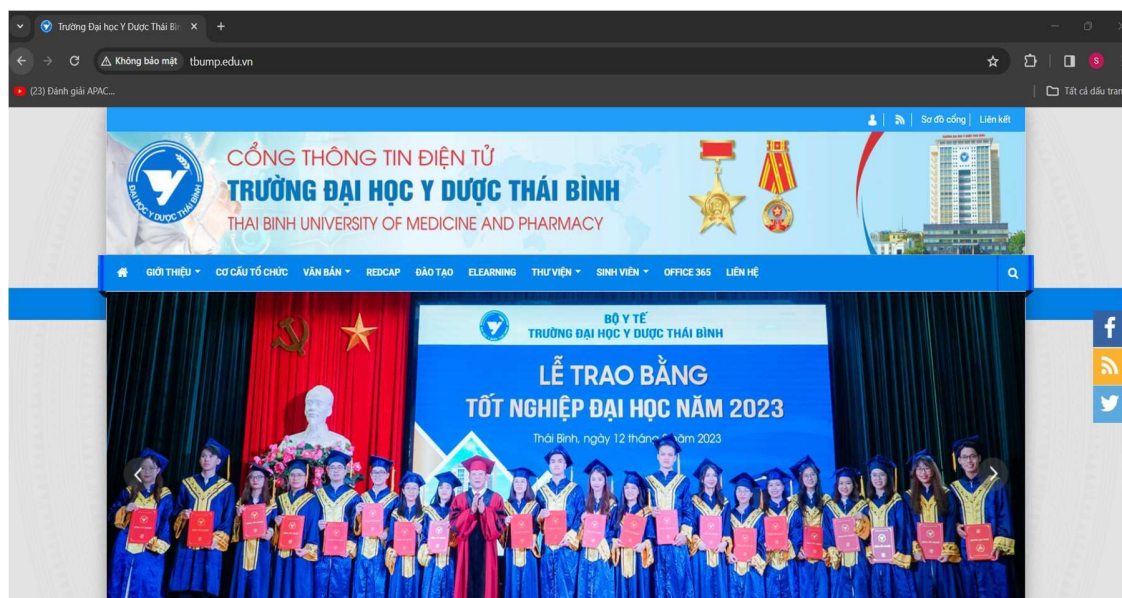


Hình 1. Mô hình giao thức HTTP

HTTP hoạt động theo mô hình Client (máy khách) – Server (máy chủ). Việc truy cập website được tiến hành dựa trên các giao tiếp giữa 2 đối tượng trên. Khi bạn truy cập một trang web qua giao thức HTTP, trình duyệt sẽ thực hiện các phiên kết nối đến server của trang web đó thông qua địa chỉ IP do hệ thống phân giải tên miền DNS cung cấp. Máy chủ sau khi nhận lệnh, sẽ trả về lệnh tương ứng giúp hiển thị website, bao gồm các nội dung như: văn bản, ảnh, video, âm thanh, ...

Trong quá trình kết nối và trao đổi thông tin, tính xác thực và toàn vẹn của dữ liệu sẽ không được bảo đảm. Địa chỉ IP của server không có biện pháp xác thực nào cả. Bên cạnh đó, các thông tin được gửi qua giao thức này cũng không được mã hóa và bảo mật.

Ví dụ: Bắt gói tin trên trang web <http://tbump.edu.vn/> (Địa chỉ IP: 116.104.51.204) bằng wireshark:



Hình 2. Trang web tbump.edu.

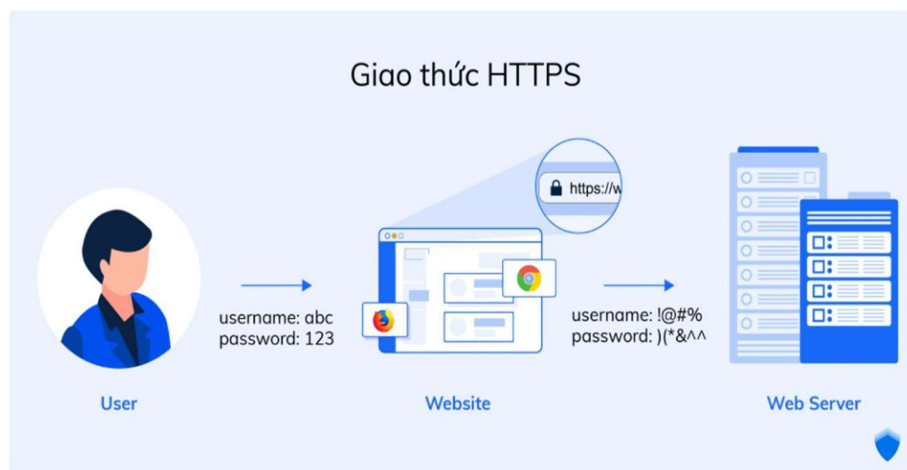
Khi truy cập vào trang web, Wireshark sẽ bắt được những gói tin được trao đổi giữa người dùng và server. Do trang web đang sử dụng giao thức http nên nội dung của các gói tin có thể dễ dàng bị xem bởi wireshark. Phân tích gói tin http, ta sẽ có được nội dung mà bên client và server đang trao đổi. Điều này tạo điều kiện cho kẻ xấu dễ dàng nắm bắt được các thông tin nhạy cảm mà 2 bên đang trao đổi và đánh cắp những thông tin đó.

```
> Frame 54: 290 bytes on wire (2320 bits), 290 bytes captured (2320 bits) on interface \Device\NPF_{10381317-87BA-4F63-BA6F-45050DA4167E}, id 0
> Ethernet II, Src: VietnamPostA_50:14:a8 (cc:71:90:50:14:a8), Dst: Intel_cd:8f:4d (28:7f:cf:cd:8f:4d)
> Internet Protocol Version 4, Src: 116.104.51.204, Dst: 192.168.1.6
▼ Transmission Control Protocol, Src Port: 80, Dst Port: 65528, Seq: 21781, Ack: 798, Len: 236
  Source Port: 80
  Destination Port: 65528
  [Stream index: 6]
  > [Conversation completeness: Incomplete (12)]
  [TCP Segment Len: 236]
  Sequence Number: 21781 (relative sequence number)
  Sequence Number (raw): 1553003196
  [Next Sequence Number: 22017 (relative sequence number)]
  Acknowledgment Number: 798 (relative ack number)
  Acknowledgment number (raw): 1004694345
  0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
  Window: 501
  [Calculated window size: 501]
  [Window size scaling factor: -1 (unknown)]
  Checksum: 0xff53 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  > [Timestamps]
  > [SEQ/ACK analysis]
  TCP payload (236 bytes)
  TCP segment data (236 bytes)
  > [16 Reassembled TCP Segments (22016 bytes): #36(1452), #37(1452), #38(1452), #39(1452), #40(1452), #41(1452), #42(1452), #43(1452), #44(1452), #45(1452), #48(1452)]
  > Hypertext Transfer Protocol
  ▼ Line-based text data: text/html (1896 lines)
    <!DOCTYPE html>\n
    \t<html lang="vi" xmlns="http://www.w3.org/1999/xhtml" prefix="og: http://ogp.me/ns#">\n
    \t<head>\n
    <title>Trường Đại học Y Dược Thái Bình</title>\n
    <meta name="description" content="Tin Tức - Tin Tức - http&#x3A;&#x002F;&#x002F;tump.edu.vn&#x002F;">\n
```

Hình 3. Gói tin bắt được trên Wireshark

2.2. Giao thức HTTPS

HTTPS (Hypertext Transfer Protocol Secure) là giao thức truyền tải siêu văn bản an toàn. Thực chất, đây chính là giao thức HTTP nhưng tích hợp thêm Chứng chỉ bảo mật SSL/TLS nhằm mã hóa các thông điệp giao tiếp để tăng tính bảo mật. Có thể hiểu, HTTPS là phiên bản HTTP an toàn, bảo mật hơn.



Hình 4. Giao thức https

HTTPS hoạt động tương tự như HTTP, tuy nhiên được bổ sung thêm chứng chỉ **SSL (Secure Sockets Layer – tầng ổ bảo mật)** hoặc **TLS (Transport Layer Security – bảo mật tầng truyền tải)**. Hiện tại, đây là các tiêu chuẩn bảo mật hàng đầu cho hàng triệu website trên toàn thế giới.

Với việc bổ sung SSL/TLS, trước khi diễn ra quá trình trao đổi dữ liệu, phía client và phía Server sẽ diễn ra TLS handshake, nhằm xác minh danh tính server và trao đổi các khóa để mã hóa và giải mã liệu. Dữ liệu sau TLS handshake sẽ được mã hóa bằng các **khóa đã được trao đổi** và chỉ có bên server và client dùng các **khóa đã được trao đổi** đã giải mã dữ liệu đó. Từ đó, tính toàn vẹn, tính xác thực, tính bảo mật của thông tin trao đổi giữa 2 bên sẽ luôn được đảm bảo.

2.3. Sự khác nhau giữa HTTP và HTTPS

Port trên HTTP và HTTPS:

HTTP sử dụng Port 80 trong khi HTTPS là Port 443.

Mã hóa trên HTTP và HTTPS:

HTTPS được mã hóa thông tin, sử dụng SSL/ TLS tiêu chuẩn công nghệ bảo mật, truyền thông mã hóa giữa máy chủ Web server và trình duyệt. Với HTTP thì hoàn toàn không.

Mức độ bảo mật HTTP vs HTTPS:

HTTPS hỗ trợ việc xác thực tính đích danh của website mà máy khách truy cập thông qua việc kiểm tra xác thực bảo mật (Security Certificate). Các xác thực bảo mật này được cung cấp và xác minh bởi các CA (Certificate Authority) uy tín. Khi được xác thực từ CA người dùng sẽ biết được mình đang truy cập vào đúng website cần tìm thay vì một web mạo danh nào đó. Việc bảo mật HTTPS không phải là 100% an toàn nhưng tốt hơn HTTP rất nhiều. Tất nhiên với HTTP không được mã hóa thông tin nên rất dễ bị Hacker tấn công.

2.4. Tìm hiểu về SSL/TLS

2.4.1. Khái niệm chứng chỉ SSL/TLS

Chứng chỉ SSL/TLS là đối tượng kỹ thuật số cho phép các hệ thống xác minh danh tính và sau đó thiết lập kết nối mạng được mã hóa với một hệ thống khác bằng giao thức Lớp công bảo mật/Bảo mật lớp truyền tải (SSL/TLS). Các chứng chỉ được sử dụng trong một hệ thống mật mã được gọi là cơ sở hạ tầng khóa công khai (PKI). PKI cung cấp phương thức để một bên thiết lập nhận dạng của một bên khác bằng cách sử dụng các chứng chỉ nếu cả hai đều tin cậy bên thứ ba - được gọi là cơ quan cấp chứng chỉ. Do đó, chứng chỉ SSL/TLS đóng vai trò là thẻ định danh kỹ thuật số để bảo mật hoạt động giao tiếp mạng, thiết lập danh tính của các trang web qua Internet cũng như tài nguyên trên các mạng riêng.

2.4.2. PKI

Trong mật mã học, **hạ tầng khóa công khai (Public key infrastructure)** là một cơ chế để cho một bên thứ 3 (thường là nhà cung cấp chứng thực số) cung cấp và xác thực định danh các bên tham gia vào quá trình trao đổi thông tin. Cơ chế này cũng cho phép gán cho mỗi người sử dụng trong hệ thống một cặp khóa công khai/ khóa bí mật. Các quá trình này thường được thực hiện bởi một phần mềm đặt tại trung tâm và các phần mềm phối hợp

khác tại các địa điểm của người dùng. Khóa công khai thường được phân phối trong chứng thực khóa công khai.

2.4.3. Cách hoạt động của TLS/SSL

Đây là những nguyên tắc cần nắm để hiểu rõ cách hoạt động của TLS:

- Giao tiếp an toàn giữa 2 bên phải bắt đầu bằng TLS handshake, để mở một kết nối an toàn và trao đổi khóa chung.
- Trong TLS handshake, hai bên tạo ra khóa phiên (khóa đối xứng) và khóa phiên sẽ mã hóa và giải mã tất cả thông tin liên lạc sau cái bắt tay TLS.
- Các khóa phiên là khác nhau trong mỗi phiên liên lạc khác nhau.
- TLS đảm bảo rằng bên phía máy chủ hoặc trang web mà người dùng đang tương tác thực sự chính là người mà họ xác nhận.
- TLS bảo đảm tính toàn vẹn của dữ liệu vì mã xác thực tin nhắn (MAC) được bao gồm trong quá trình truyền.

Với TLS, cả dữ liệu HTTP mà người dùng gửi đến một trang web (bằng cách nhấp vào, điền vào biểu mẫu, v.v.) và dữ liệu HTTP mà trang web gửi cho người dùng đều được mã hóa. Dữ liệu được mã hóa phải được người nhận giải mã bằng khóa.

TLS handshake sử dụng cơ chế mã hóa khóa bất đối xứng, với 2 bên đều có 2 loại khóa riêng (khóa chung và khóa riêng) của mình. Server và Client sử dụng khóa chung và khóa riêng để trao đổi dữ liệu được tạo ngẫu nhiên và dữ liệu ngẫu nhiên này được sử dụng để tạo khóa mới để mã hóa, được gọi là khóa phiên. Bằng kỹ thuật trao đổi khóa công khai (RSA, Diffie Hellman, Elliptic Curve,...) nên 2 bên có thể trao đổi 1 cách an toàn trên 1 kênh truyền không tin cậy.

Trong giao tiếp TLS, phía server sẽ gửi kèm một mã xác thực tin nhắn (message authentication code - MAC), đóng vai trò như chữ ký số điện tử. MAC xác nhận rằng luồng giao tiếp thực sự bắt nguồn từ website đích thực. Bằng cách này, TLS giúp xác thực server, ngăn chặn các tấn công trung gian (on-path attacks) và giả mạo tên miền (domain spoofing). Ngoài ra, MAC còn đảm bảo dữ liệu không bị can thiệp trên đường truyền.

3. Phân tích TLS handshake

Phân tích TLS handshake khi kết nối với trang web: <https://www.neu.edu.vn/>



Hình 5. Trang web www.neu.edu.vn

Kết quả trên Wireshark như sau:

87	1.639224	192.168.1.3	171.244.50.30	TLSv1.2	571 Client Hello (SNI=www.neu.edu.vn)
97	1.647550	171.244.50.30	192.168.1.3	TLSv1.2	493 Server Hello, Certificate, Certificate Status, Server Key Exchange, Server Hello Done
100	1.651741	192.168.1.3	171.244.50.30	TLSv1.2	212 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
102	1.652105	192.168.1.3	171.244.50.30	TLSv1.2	153 Application Data
103	1.652233	192.168.1.3	171.244.50.30	TLSv1.2	557 Application Data
105	1.657517	171.244.50.30	192.168.1.3	TLSv1.2	105 Change Cipher Spec, Encrypted Handshake Message
120	1.698575	192.168.1.3	171.244.50.30	TLSv1.2	92 Application Data
163	2.272306	171.244.50.30	192.168.1.3	TLSv1.2	1506 Application Data, Application Data
180	2.290478	171.244.50.30	192.168.1.3	TLSv1.2	1506 Application Data, Application Data
181	2.292781	192.168.1.3	171.244.50.30	TLSv1.2	234 Application Data
200	2.302516	171.244.50.30	192.168.1.3	TLSv1.2	1506 Application Data, Application Data
203	2.304977	192.168.1.3	171.244.50.30	TLSv1.2	138 Application Data
204	2.305145	192.168.1.3	171.244.50.30	TLSv1.2	141 Application Data
205	2.305225	192.168.1.3	171.244.50.30	TLSv1.2	152 Application Data
206	2.305279	192.168.1.3	171.244.50.30	TLSv1.2	146 Application Data
207	2.305340	192.168.1.3	171.244.50.30	TLSv1.2	143 Application Data

Hình 6. Các gói tin bắt được trên Wireshark khi kết nối đến trang web

Khởi đầu TLS Shakedown là gói tin Client Hello:

87	1.639224	192.168.1.3	171.244.50.30	TLSv1.2	571 Client Hello (SNI=www.neu.edu.vn)
97	1.647550	171.244.50.30	192.168.1.3	TLSv1.2	493 Server Hello, Certificate, Certificate Status, Server Key Exchange, Serv
100	1.651741	192.168.1.3	171.244.50.30	TLSv1.2	212 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message

```

v TLSv1.2 Record Layer: Handshake Protocol: Client Hello
  Content Type: Handshake (22)
  Version: TLS 1.0 (0x0301)
  Length: 512
v Handshake Protocol: Client Hello
  Handshake Type: Client Hello (1)
  Length: 508
  Version: TLS 1.2 (0x0303)
v Random: c158bcb765ceb4661458ed978ea9e9ae5e733f02e20072551a865f48d1203102
  GMT Unix Time: Oct 16, 2072 11:36:07.000000000 SE Asia Standard Time
  Random Bytes: 65ceb4661458ed978ea9e9ae5e733f02e20072551a865f48d1203102
  Session ID Length: 32
  Session ID: 4e9cab91777aa1b9c575a48fda02cd1c78f47f83182c4ba902af3581b76e88d
  Cipher Suites Length: 32
v Cipher Suites (16 suites)
  Cipher Suite: Reserved (GREASE) (0x8a8a)
  Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
  Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
  Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)
  Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
  Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
  Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xc03a)
  Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xc03b)
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
  Cipher Suite: TLS_RSA_WITH_AES_128_GCM_SHA256 (0x009c)
  Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
  Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)

```

Hình 7. Chi tiết gói tin Client Hello

Trong gói tin Client Hello, Client sẽ gửi cho bên Server một số thông tin về:

- version TLS khả dụng (TLS 1.2)
- Session ID
- bộ Cipher Suites: là bộ các thuật toán được sử dụng để thiết lập bảo mật. Trong ảnh có 16 suites, Server sẽ chọn một trong số các suites đó để thống nhất các quy tắc bảo mật giữa cả 2 bên.

Tiếp đó, Server sẽ trả lời Client bằng gói tin Server Hello:

```

v TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
  Content Type: Handshake (22)
  Version: TLS 1.2 (0x0303)
  Length: 4790
v Handshake Protocol: Server Hello
  Handshake Type: Server Hello (2)
  Length: 94
  Version: TLS 1.2 (0x0303)
v Random: 65fc871d2413684de376ffed911a379e6981d846bdf1097ff5cdea0f7fc349a0
  GMT Unix Time: Mar 22, 2024 02:14:37.000000000 SE Asia Standard Time
  Random Bytes: 2413684de376ffed911a379e6981d846bdf1097ff5cdea0f7fc349a0
  Session ID Length: 32
  Session ID: ad3b0000250583644f128d0fa741c652a44d0faea40164b635ede651a31a1964
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
  Compression Method: null (0)
  Extensions Length: 22
  > Extension: status_request (len=0)
  > Extension: application_layer_protocol_negotiation (len=5)
  > Extension: extended_master_secret (len=0)
  > Extension: renegotiation_info (len=1)
  [JA3S Fullstring: 771,49200,5-16-23-65281]
  [JA3S: 67bfe5d15ae567fb35fd7837f0116eec]
v Handshake Protocol: Certificate
  Handshake Type: Certificate (11)
  Length: 2879
  Certificates Length: 2876
  Certificates (2876 bytes)
  > Certificate Length: 1764
  > Certificate [truncated]: 308206e0308205c8a003020102020c184b98f2f55319f64fcb4e5300d06092a864886f70d01010b05003050310b3009060355040613024245311
  > Certificate Length: 1106
  > Certificate [truncated]: 3082044e30820336a003020102020d01ee5f221dfc623bd4333a8557300d06092a864886f70d01010b0500304c3120301e060355040b1317476c6
v Handshake Protocol: Certificate Status
  Handshake Type: Certificate Status (22)
  Length: 4436

```

```

    Handshake Protocol: Server Key Exchange
      Handshake Type: Server Key Exchange (12)
      Length: 361
    EC Diffie-Hellman Server Params
      Curve Type: named_curve (0x03)
      Named Curve: secp384r1 (0x0018)
      Pubkey Length: 97
      Pubkey: 0496e2730e9a17a2aaa4a5d75ef5cefd1e1a8cf2bd9a8775ddc1396ef1bf6063f0e341405398d2d642ef327a79f2b1a9dfe39a9db54044f7fbb60c40b0e69232d8ee6c
    Signature Algorithm: rsa_pkcs1_sha256 (0x0401)
      Signature Hash Algorithm Hash: SHA256 (4)
      Signature Hash Algorithm Signature: RSA (1)
      Signature Length: 256
      Signature [truncated]: 6e147e0228bcfecf45413468e95f925705afbce311e9f47b344163c12a04db8a7e5f2d5af653b4974d839b06fa6e86d57eda23c62031defbd364094
    Handshake Protocol: Server Hello Done
      Handshake Type: Server Hello Done (14)
      Length: 0

```

Hình 8. Gói tin Server Hello

Trong gói tin này, Server sẽ gửi cho bên Client những thông tin như:

- Bộ Cipher Suite mà bên Server chọn để thống nhất các giao thức trao đổi khóa, mã hóa và bảo vệ tính toàn vẹn của dữ liệu là: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384.
- Session ID.
- Certificates: xác minh Server này là Server thật qua chuỗi các chứng chỉ từ các bên cung cấp chứng chỉ. Bên Client sẽ xác minh chữ ký của các chứng chỉ và nếu chữ ký hợp lệ thì sẽ trao đổi khóa cho bên Server.
- Do Server chọn giao thức trao đổi khóa ECDHE, nên bên Server sẽ gửi các tham số (Pubkey) của đường cong Elliptic curve để thực hiện trao đổi khóa trên kênh truyền không tin cậy.
- Server Hello Done: thông báo đã đưa hết thông tin từ bên Server.

Sau khi nhận được Server Hello, bên người dùng sẽ gửi Client Key Exchange:

87	1.639224	192.168.1.3	171.244.50.30	TLSv1.2	571 Client Hello (SNI=www.neu.edu.vn)
97	1.647550	171.244.50.30	192.168.1.3	TLSv1.2	493 Server Hello, Certificate, Certificate Status, Server Key Exchange
100	1.651741	192.168.1.3	171.244.50.30	TLSv1.2	212 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
102	1.652105	192.168.1.3	171.244.50.30	TLSv1.2	153 Application Data
103	1.652233	192.168.1.3	171.244.50.30	TLSv1.2	557 Application Data
105	1.657517	171.244.50.30	192.168.1.3	TLSv1.2	105 Change Cipher Spec, Encrypted Handshake Message
120	1.698575	192.168.1.3	171.244.50.30	TLSv1.2	92 Application Data
163	2.272306	171.244.50.30	192.168.1.3	TLSv1.2	1506 Application Data, Application Data
180	2.290478	171.244.50.30	192.168.1.3	TLSv1.2	1506 Application Data, Application Data
181	2.292781	192.168.1.3	171.244.50.30	TLSv1.2	234 Application Data

```

> Frame 100: 212 bytes on wire (1696 bits), 212 bytes captured (1696 bits) on interface \Device\NPF_{10381317-87BA-4F63-BA6F-45050DA4167E}, id 0
> Ethernet II, Src: Intel_cd:8f:4d (28:7f:cf:cd:8f:4d), Dst: VietnamPostA_50:14:a8 (cc:71:90:50:14:a8)
> Internet Protocol Version 4, Src: 192.168.1.3, Dst: 171.244.50.30
> Transmission Control Protocol, Src Port: 58320, Dst Port: 443, Seq: 518, Ack: 4796, Len: 158
▼ Transport Layer Security
  ▼ TLSv1.2 Record Layer: Handshake Protocol: Client Key Exchange
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 102
    ▼ Handshake Protocol: Client Key Exchange
      Handshake Type: Client Key Exchange (16)
      Length: 98
      ▼ EC Diffie-Hellman Client Params
        Pubkey Length: 97
        Pubkey: 04dcad420754ac6ca6ae96c188a41856abb679a2ab085da6694a81cb67c18e230b941080ed13391cac346c05815a51353a23c3376c51034b3101149df46e33f
  ▼ TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
    Content Type: Change Cipher Spec (20)
    Version: TLS 1.2 (0x0303)
    Length: 1
    Change Cipher Spec Message
  ▼ TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 40
    Handshake Protocol: Encrypted Handshake Message

```

Hình 9. gói tin Client Key Exchange

Trong gói tin Client Key Exchange, Client sẽ gửi nốt các tham số còn lại cho giao thức mã hóa ECDHE (Pubkey) để hoàn tất quá trình trao đổi khóa. Từ đó, 2 bên Client và Server đã tạo được ra khóa phiên giống nhau (Khóa đối xứng) bằng thuật toán Diffie Hellman và Elliptic Curve. Vì vậy, từ sau gói tin này, tất cả các gói tin khác đã bắt đầu được mã hóa.

102	1.652105	192.168.1.3	171.244.50.30	TLSv1.2	153 Application Data
103	1.652233	192.168.1.3	171.244.50.30	TLSv1.2	557 Application Data
105	1.657517	171.244.50.30	192.168.1.3	TLSv1.2	105 Change Cipher Spec, Encrypted Handshake Message
120	1.698575	192.168.1.3	171.244.50.30	TLSv1.2	92 Application Data
163	2.272306	171.244.50.30	192.168.1.3	TLSv1.2	1506 Application Data, Application Data
180	2.290478	171.244.50.30	192.168.1.3	TLSv1.2	1506 Application Data, Application Data
181	2.292781	192.168.1.3	171.244.50.30	TLSv1.2	234 Application Data

> Frame 102: 153 bytes on wire (1224 bits), 153 bytes captured (1224 bits) on interface \Device\NPF_{10381317-87BA-4F63-BA6F-450500A4167E}, id 0	0000	cc 71 90 50 14 a8 28 7f cf
> Ethernet II, Src: Intel_cd:8f:4d (28:7f:cf:cd:8f:4d), Dst: VietnamPostA_50:14:a8 (cc:71:90:50:14:a8)	0010	00 8b 24 f7 40 00 80 06 00
> Internet Protocol Version 4, Src: 192.168.1.3, Dst: 171.244.50.30	0020	32 1e e3 d0 01 bb d6 32 8a
> Transmission Control Protocol, Src Port: 58320, Dst Port: 443, Seq: 676, Ack: 4796, Len: 99	0030	02 01 a0 3b 00 00 17 03 03
> Transport Layer Security	0040	00 00 01 94 d9 f6 16 ed cd
▼ TLSv1.2 Record Layer: Application Data Protocol: HyperText Transfer Protocol 2	0050	21 f5 2a 5f fe 32 cf c2 a3
Content Type: Application Data (23)	0060	d6 74 7a 66 01 5f 14 56 49
Version: TLS 1.2 (0x0303)	0070	b9 35 9c cc 30 b5 61 40 bd
Length: 94	0080	30 f8 27 7a 0a 89 cf c4 c2
Encrypted Application Data: 0000000000000000194d9f616edcd0f63c92531b18e21f52e5ffe32cfc2a3cc23bfe7dc09d6747a66015f14564949277926b897dfb9359cc	0090	68 98 19 12 6d 4f 9c d9 2e
[Application Data Protocol: HyperText Transfer Protocol 2]		

Hình 10. Các gói tin sau đã bắt đầu được mã hóa bởi khóa phiên

4. Tự triển khai HTTP/HTTPS bằng cách tạo https server.

4.1. Tạo chứng chỉ cho server bằng OpenSSL.

Để tạo ra server được bảo mật với SSL/TLS, đầu tiên phải tạo ra chứng chỉ cho server. Trong giới hạn của project, server chỉ cần chứng chỉ tự ký (self-signed certificate). Trước hết, chúng ta phải tạo ra khóa bí mật gốc (root-level private key) và yêu cầu chứng chỉ gốc (root certificate request). Dùng lệnh sau để tạo ra khóa và yêu cầu trên:

```
$ openssl req -newkey rsa:2048 -keyout root_key.pem -out root_request.pem
```

Sau khi dùng lệnh trên thì khóa bí mật được mã hóa sẽ ở trong root_key.pem và yêu cầu chứng chỉ gốc sẽ ở trong root_request.pem. Tiếp theo, chúng ta sẽ tạo ra chứng chỉ cho gốc (root_certificate) bằng lệnh sau:

```
$ openssl x509 -req -in root_request.pem -signkey root_key.pem -out root_certificate
```

Sau đó, để thuận tiện chúng ta sẽ gộp root_certificate và root_key vào chung 1 gói root.pem:

```
$ cat root_certificate.pem root_key.pem > root.pem
```

Bước tiếp theo là tạo ra chứng chỉ cho CA(Certificate Authority). Đầu tiên chúng ta cần tạo ra CA private key và certificate request:

```
$ openssl req -newkey rsa:2048 -keyout CA_key.pem -out CA_request.pem
```

Tạo CA certificate được ký bằng root.pem:

```
$ openssl x509 -req -in CA_request.pem -CA root.pem -CAkey root.pem -CAcreateserial -out CAcert.pem
```

Gộp CAcert.pem , CA_key.pem và root_certificate.pem vào 1 file CA.pem:

```
$ cat CAcert.pem CA_key.pem root_certificate.pem > CA.pem
```

Sau khi đã có CA certificate thì chúng ta đã có thể tạo ra self-signed certificate cho server. Đầu tiên, chúng ta cần tạo khóa cho server:

```
$ openssl genrsa 2048 > server_key.pem
```

Tạo server request từ server_key.pem:

```
$ openssl req -new -key server_key.pem -out server_request.pem
```

Tạo server_certificate được ký bởi CA:

```
$ openssl x509 -req -in server_request.pem -CA CA.pem -CAcreateserial -CAkey CA.pem -out server_certificate.pem
```

Gộp server_key.pem, server_certificate.pem, CAcert.pem, root_certificate.pem thành 1 file server.pem (tạo thành Chain of Trust)

```
$ cat server_certificate.pem server_key.pem CAcert.pem \ root_certificate.pem > server.pem
```

4.2. Cài đặt SSL/TLS cho server bằng OpenSSL

SSL_CTX: chứa những thông tin về SSL/TLS hiện tại của server

```
SSL_CTX *my_ssl_ctx;
```

```
my_ssl_ctx = SSL_CTX_new(my_ssl_method);
```

Cung cấp private key và chứng chỉ của server (đã tạo ở 4.1) cho SSL_CTX:

```
SSL_CTX_use_certificate_file(my_ssl_ctx,"server.pem",SSL_FILETYPE_PEM);  
SSL_CTX_use_PrivateKey_file(my_ssl_ctx,"server.pem",SSL_FILETYPE_PEM);
```

Liên kết SSL với một kết nối mới:

```
...
```

```
client_fd = accept(my_fd, (sockaddr *)&client, (socklen_t *)&client_size);
```

```
...
```

```
if((my_ssl = SSL_new(my_ssl_ctx)) == NULL)
```

```
{
```

```
    ERR_print_errors_fp(stderr);
```

```
    exit(-1);
```

```
}
```

```
SSL_set_fd(my_ssl,client_fd);
```

4.3. Cài đặt https_server để xử lý yêu cầu của client

Khởi tạo SSL với khóa đã tạo ra trước và cấu trúc địa chỉ server:

```

SSL_library_init();
OpenSSL_add_all_algorithms();
SSL_load_error_strings();
my_ssl_ctx = SSL_CTX_new(TLS_server_method());
SSL_CTX_use_certificate_file(my_ssl_ctx, "server.pem", SSL_FILETYPE_PEM);
SSL_CTX_use_PrivateKey_file(my_ssl_ctx, "server.pem", SSL_FILETYPE_PEM);
if( !SSL_CTX_check_private_key(my_ssl_ctx) )
{
    fprintf(stderr, "Private key does not match certificate\n");
    exit(-1);
}

my_fd = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
server.sin_family = AF_INET;
server.sin_port = htons(5353);
server.sin_addr.s_addr = htonl(INADDR_ANY);
bind(my_fd, (struct sockaddr *)&server, sizeof(server));
listen(my_fd, 5);

```

Hình 11. Khởi tạo SSL và cấu hình server

Liên kết SSL với mỗi kết nối mới:

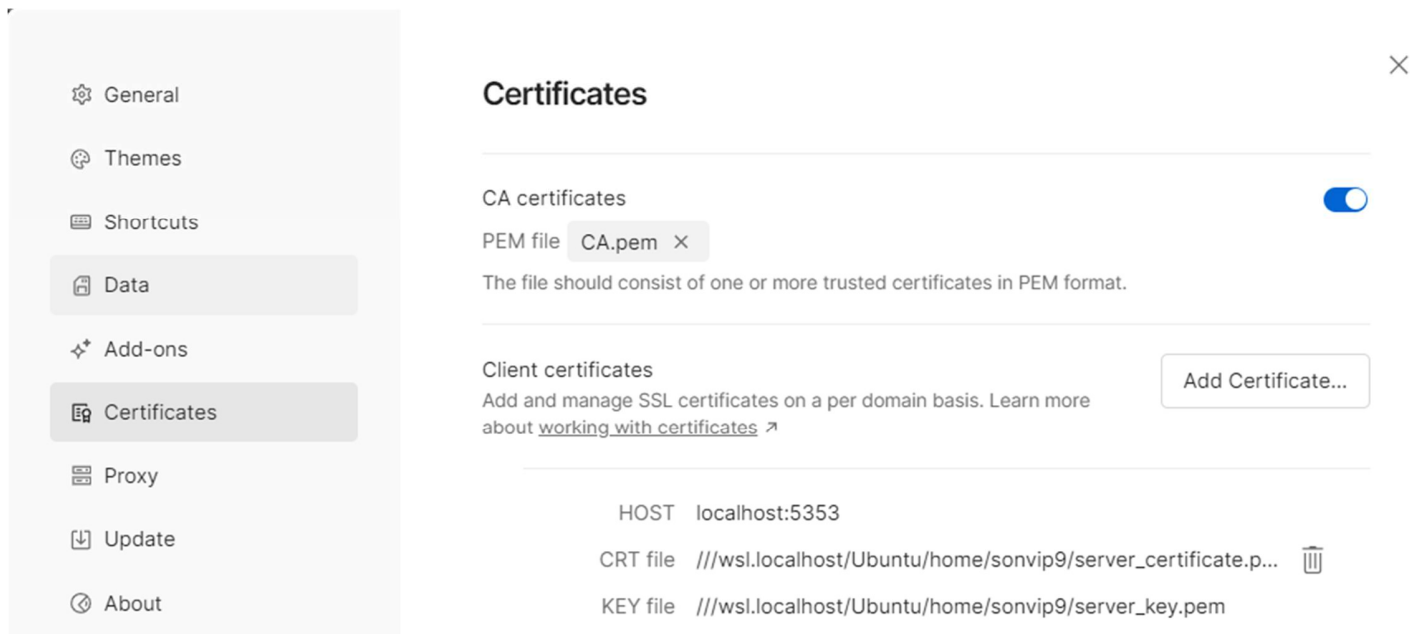
```

void *client_proc(void *arg)
{
    int m_client = *(int *)arg;
    if((my_ssl = SSL_new(my_ssl_ctx)) == NULL)
    {
        ERR_print_errors_fp(stderr);
        exit(-1);
    }
    SSL_set_fd(my_ssl, m_client);
    if(SSL_accept(my_ssl) <= 0)
    {
        ERR_print_errors_fp(stderr);
        exit(-1);
    }
}

```

Hình 12. Liên kết SSL với mỗi kết nối mới

Cài đặt môi trường dùng Postman để gửi những thông điệp https, cài đặt trước khóa và chứng chỉ để Postman có thể kiểm tra trước khi thiết lập kết nối với server:

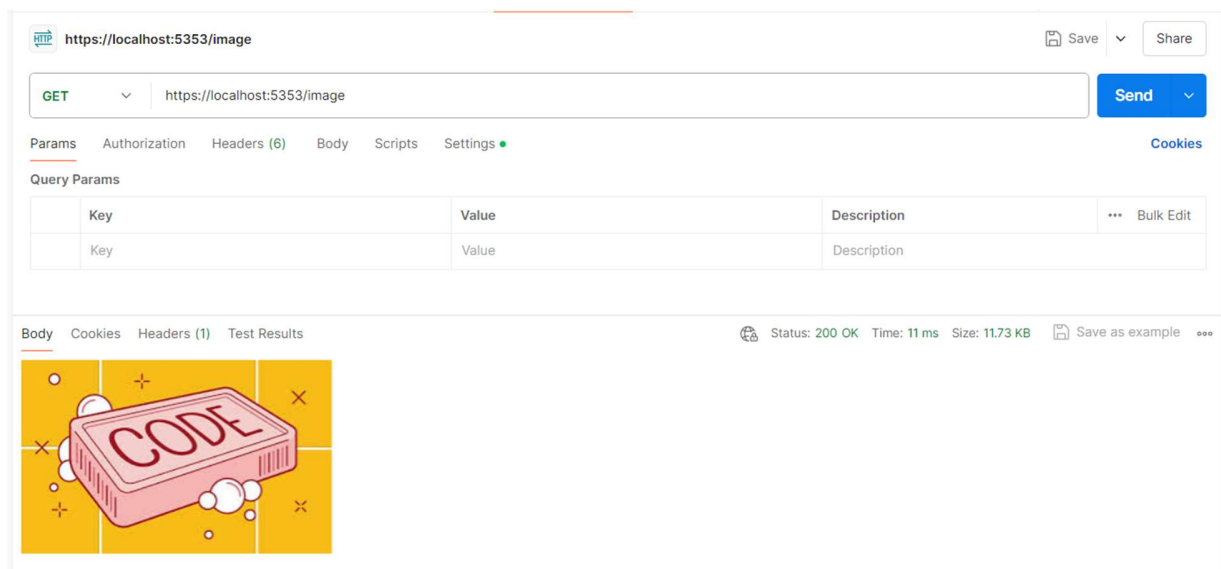


Hình 13. Khai báo chứng chỉ của server cho Postman

Code hoàn chỉnh: https://github.com/SonProCode/Project---II/blob/main/https_server.c

4.4. Kết quả chạy server:

- Xử lý các thông điệp GET:



Hình 14. Server phản hồi lại ảnh khi nhận thông điệp GET: `https://localhost:5353/image` từ Postman

HTTP <https://localhost:5353/video1> Save Share


GET <https://localhost:5353/video1> Send

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (2) Test Results Status: 200 OK Time: 26 ms Size: 772.89 KB Save as example



Hình 15. Server phản hồi lại video khi nhận thông điệp GET: <https://localhost:5353/video1> từ Postman

HTTP <https://localhost:5353/video2> Save Share


GET <https://localhost:5353/video2> Send

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (2) Test Results Status: 200 OK Time: 62 ms Size: 1.15 MB Save as example



Hình 16. Server phản hồi lại video khi nhận thông điệp GET: <https://localhost:5353/video2> từ Postman

GET https://localhost:5353/index.html

Status: 200 OK Time: 9 ms Size: 3.25 KB

Body

Pretty Raw Preview Visualize

Các Sản Phẩm Của Công Ty

Công ty chúng tôi tự hào giới thiệu đến quý khách hàng các sản phẩm chất lượng với mức giá cạnh tranh. Dưới đây là một số sản phẩm tiêu biểu của chúng tôi:

- Sản phẩm A: Sản phẩm A là một sản phẩm công nghệ tiên tiến, giúp người dùng tối ưu hóa công việc hàng ngày.
- Sản phẩm B: Sản phẩm B là một dòng sản phẩm thời trang cao cấp, đáp ứng mọi nhu cầu của khách hàng.
- Sản phẩm C: Sản phẩm C là một giải pháp phần mềm hiệu quả cho các doanh nghiệp vừa và nhỏ.

Bảng Thống Kê Doanh Thu

Dưới đây là bảng thống kê doanh thu các sản phẩm trong 12 tháng qua:

Sản Phẩm	Tháng 1	Tháng 2	Tháng 3	Tháng 4	Tháng 5	Tháng 6	Tháng 7	Tháng 8	Tháng 9	Tháng 10	Tháng 11	Tháng 12
Sản phẩm A	5000	5500	6000	6500	7000	7500	8000	8500	9000	9500	10000	10500

Hình 17. Server phản hồi lại index.html khi nhận thông điệp GET: <https://localhost:5353/index.html> từ Postman

GET https://localhost:5353/audio

Status: 200 OK Time: 144 ms Size: 4.3 MB

Body

0:00 / 1:52

Hình 18. Server phản hồi lại file audio khi nhận thông điệp GET: <https://localhost:5353/audio> từ Postman

- Xử lý các thông điệp POST: Ở đây giả sử client sẽ gửi tài khoản và mật khẩu để đăng nhập. Nếu client chưa có tài khoản thì người dùng có thể đăng ký tài khoản.

https://localhost:5353/login?username=test_user&pass=12

POST https://localhost:5353/login?username=test_user&pass=12

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

<input checked="" type="checkbox"/>	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	username	test_user			
<input checked="" type="checkbox"/>	pass	12			
	Key	Value	Description		

Body Cookies Headers (1) Test Results

Pretty Raw Preview Visualize

Tài khoản hoặc mật khẩu sai!

Hình 19. Đăng nhập không thành công

https://localhost:5353/register?username=test_user&pass=12

POST https://localhost:5353/register?username=test_user&pass=12

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

<input checked="" type="checkbox"/>	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	username	test_user			
<input checked="" type="checkbox"/>	pass	12			
	Key	Value	Description		

Body Cookies Headers (1) Test Results

Pretty Raw Preview Visualize

Đăng ký thành công

Hình 20. Đăng ký tài khoản

The screenshot shows a REST client interface with the URL `https://localhost:5353/login?username=test_user&pass=12`. The request method is **POST**. The query parameters are:

Key	Value	Description
username	test_user	
pass	12	

The response status is **200 OK** with a time of 12 ms and a size of 74 B. The response body is displayed in the "Preview" tab:

Đăng nhập thành công!

Hình 21. Đăng nhập thành công sau khi đã đăng ký tài khoản

The screenshot shows a REST client interface with the URL `https://localhost:5353/audio2`. The request method is **POST**. The response status is **404 Not Found** with a time of 12 ms and a size of 78 B. The response body is displayed in the "Preview" tab:

Không tìm thấy tài nguyên

Hình 22. Truy cập vào tài nguyên không tồn tại

Tài liệu tham khảo:

1. The definitive guide to linux network programming – Keir Davis
2. Slide môn học lập trình mạng IT4060: [network_programming/materials/Lap_trinh_mang_IT4060.pdf at master · lebavui/network_programming \(github.com\)](#)
3. <https://cystack.net/blog/http-va-https-la-gi>
4. <https://www.cloudflare.com/learning/ssl/how-does-ssl-work/>