Mục lục

1 Một số khái niệm

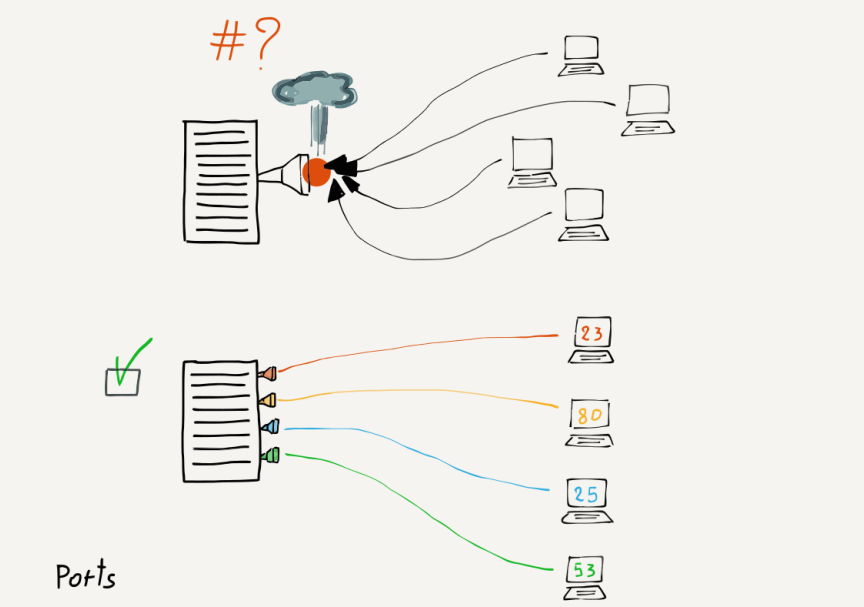
1.1 Khái niệm mô hình ứng dụng Clients-Servers

Mục đích của lớp Transport là cung cấp kết nối tin cậy giữa các ứng dụng trên network để ứng dụng có thể gửi và nhận dữ liệu.
Với một ứng dụng rất đơn giản trong việc yêu cầu lớp Transport tạo một kết nối đến ứng dụng đang chạy trên máy chủ ở xa. Ứng dụng khởi tạo kết nối trên máy cục bộ được gọi là "máy khách"(client) và ứng dụng phản hồi yêu cầu kết nối gọi là "máy chủ"(server). Đó là nguyên tắc cơ bản đằng sau sự duyệt web phải có cả hai thành phần client/server cùng làm việc với nhau.
Khi một ứng dụng máy khách muốn tạo một kết nối đến một máy tính ở xa, điều quan trọng là phải kết nối chính xác đến ứng dụng máy chủ ở xa đó. Ngược lại máy tính ở xa có thể có nhiều ứng dụng máy chủ đang được chạy ở cùng thời điểm, các ứng dụng đó có thể là Web Server, Video Server hoặc Mail Server.
Đơn cử một Web Client là một trình duyệt web muốn kết nối đến Web Server đang chạy ở một máy tính ở xa. Ở đây sử dụng một khái niệm gọi là "cổng"(port) cho phép ứng dụng máy khách lựa chọn ứng dụng tại máy chủ mà nó muốn trao đổi. Khi ứng dụng máy chủ được khởi động, nó sẽ luôn trong tình trạng "lắng nghe"(listens) một tín hiệu kết nối đến từ một cổng nhất định. Ngay khi ứng dụng máy chủ đã đăng ký sẵn sàng nhận các tín hiệu đến nó sẽ chờ đến khi kết nối đầu tiên được tạo.[1]
Ứng dụng máy khách biết trước cổng kết nối, danh sách các cổng kết nối mặc định cho nhiều ứng dụng máy chủ khác nhau:

- SSH (22): Secure Login
- HTTP (80): World Wide Web
- HTTPS (443): Secure Web
- SMTP (25): Incoming Mail
- IMAP (143/220/993): Mail Retrieval
- POP (109/110): Mail Retrieval
- DNS (53): Domain Name Resolution
- FTP (21): File Transfer

1.2 Khái niệm Thư điện tử

Thư điện tử là công cụ giao tiếp cá nhân cho phép người dùng gửi thông điệp dưới dạng tin nhắn đến một hoặc nhiều người nhận.



Hình 1: Minh hoạt mô hình Client-Server

Thư điện tử ngày nay cho phép gửi thông điệp đa phương tiện bao gồm: văn bản, hình ảnh, video, files, ...
Các thông điệp gửi đi trong Thư điện tử được mã hóa với mục đích bảo mật.[1]

2 Các thành phần chính của Thư điện tử

2.1 User Agent

User Agent là công cụ giao tiếp với người dùng hay còn gọi là "Trình đọc thư".
User Agent cho phép người dùng soạn, sửa, lưu, đọc nội dung và tìm kiếm Thư điện tử.

2.2 Mail Servers

Mail Servers là "Trình xử lý thư". Điều này có thể xem Mail Servers như một "bưu điện truyền thống", tại đó người dùng sẽ trao cho Mail Servers các thông điệp mà họ muốn gửi.
Tại điểm đến, Mail Servers bên đích sẽ nhận thông điệp được gửi đến và đặt nó trong hộp thư của người nhận. [2]

2.3 Simple Mail Transfer Protocol

SMTP là giao thức quyết định các thông điệp sẽ được trao đổi như thế nào giữa các Mail Servers với nhau.
SMTP quyết định Mail Server và có thể tự do cấu hình nên có rất nhiều sản phẩm Mail Servers trên thị trường.
Ứng dụng Thư điện tử sử dụng giao thức SMTP (RFC 2821) ở tầng Application

và giao thức TCP/IP ở tầng Transport.
Trong đó TCP/IP là một giao thức connection-oriented giúp kiểm tra các gói tin có lỗi hay không, đảm bảo cho việc gửi một thông điệp thư điện tử thành công.

3 Giao thức giữa Mail Servers và User Agent

Phần này tập trung làm rõ khái niệm các giao thức giữa Mail Servers và User Agent, các phiên làm việc của các giao thức này sẽ được trình bày ở các phần sau.

3.1 POP3

Post Office Protocol version 3 (RFC 1939) được User Agent sử dụng để lấy thư về từ hộp thư của nó trên Mail Servers.

3.2 SMTP

User Agent sử dụng SMTP để gửi thư ra Mail Servers.

3.3 IMAP

Với những người dùng có một tài khoản Thư điện tử trên một Internet Service Provider và người dùng này thường truy cập Thư điện tử trên một máy tính thì giao thức POP3 hoạt động tốt.
Tuy nhiên một sự thật trong lĩnh vực công nghệ thông tin là khi một thứ gì đó đã hoạt động tốt, người ta lập tức đòi hỏi thêm nhiều tính năng mới. Điều này cũng xảy ra đối với hệ thống Thư điện tử.
Ví dụ, người ta chỉ có một tài khoản Thư điện tử nhưng họ lại muốn ngồi đâu cũng truy cập được nó. POP3 cũng làm được chuyện này bằng cách đơn giản là tải hết các Thư điện tử xuống máy tính cá nhân mà người dùng đang trực tiếp làm việc. Và dĩ nhiên là thư từ của người dùng sẽ nằm rải rác khắp nơi.
Sự bất tiện này khơi mào cho sự ra đời của giao thức phân phối thư mới có tên là Internet Message Access Protocol (IMAP) được định nghĩa trong RFC 2060.

4 Các nguyên tắc cơ bản

Client muốn gửi thư sẽ mở kết nối TCP đến Server SMTP để gửi thư qua kết nối. Server SMTP luôn ở chế độ lắng nghe, ngay khi nó nghe kết nối TCP từ bất kỳ Client nào, quy trình SMTP sẽ bắt đầu kết nối trên port đó (thường là port 25). Sau khi thiết lập thành công kết nối TCP, tiến trình Client sẽ gửi thư ngay lập tức.
Từ nguyên tắc cơ bản hình thành 3 giai đoạn truyền:

- Thiết lập kết nối.

- Truyền thông điệp.
- Đóng kết nối.

5 Các lệnh SMTP cơ bản

5.1 HELO/EHLO

Lệnh khởi tạo cuộc trò chuyện phiên SMTP.
Máy khách chào máy chủ và giới thiệu về chính nó.
Theo quy tắc, HELO được gán với một đối số chỉ định tên miền hoặc địa chỉ IP của máy khách SMTP.

5.2 MAIL FROM

Lệnh khởi tạo một thư cần chuyển với một đối số, MAIL FROM bao gồm một hộp thư người gửi.

5.3 RCPT TO

Lệnh quy định người nhận cụ thể , trong trường hợp nhiều người nhận RCPT TO sẽ chỉ định từng người nhận cụ thể.

5.4 DATA

Máy client yêu cầu máy chủ cho phép chuyển dữ liệu thư.
Lệnh này có mã phản hồi là 354 cấp quyền và client bắt đầu khởi chạy việc gửi nội dung theo từng dòng. Dòng cuối cùng chứa dấu chấm "." để kết thúc quá trình truyền.

5.5 QUIT

Lệnh yêu cầu chấm dứt phiên SMTP.
Sau khi máy chủ phản hồi mã 211 máy client sẽ đóng kết nối SMTP.

6 Các lệnh SMTP mở rộng

Các lệnh SMTP mở rộng mà một số Server SMTP có thể hỗ trợ.

6.1 STARTTLS

Lệnh được sử dụng để bắt đầu một handshake đảm bảo một phiên SMTP an toàn.
STARTTLS đặt giao thức SMTP về trạng thái ban đầu.
Sau khi nhận mã phản hồi 220 từ máy chủ, ứng dụng client sẽ gửi HELO/EHLO để chạy phiên.

6.2 AUTH

Lệnh được sử dụng để xác thực client đến máy chủ, nó quy định một phương thức về bảo mật và đăng nhập với mã phản hồi từ máy chủ là 334 để xác thực phiên.

7 Mã phản hồi SMTP

Server SMTP phản hồi Client bằng một mã bao gồm 3 chữ số. Mỗi chữ số có một ý nghĩa riêng:

- Chữ số đầu tiên:
  Có giá trị từ 2 đến 5 biểu thị yêu cầu được chấp nhận, yêu cầu không đầy đủ hoặc yêu cầu bị từ chối.
- Chữ số thứ 2:
  Có giá trị từ 0 đến 5 biểu thị loại lỗi xảy ra bao gồm lỗi cú pháp, lỗi thông tin, lỗi kết nối, lỗi hệ thống hoặc lỗi không xác định.
- Chữ số thứ 3:
  Có giá trị từ 0 đến 5 cung cấp mô tả tốt nhất cùng với giải thích bằng văn bản.

Ngoài ra mỗi mã phản hồi còn đi kèm một văn bản giải thích cho người dùng. Các máy chủ khác nhau sử dụng mô tả phản hồi khác nhau tuy nhiên mã phản hồi thì không thay đổi.

Code	Description
101	Lỗi kết nối Server (tên Server sai hoặc port kết nối sai)
211	Trạng thái hệ thống (phản hồi với HELP)
214	Thông báo trợ giúp (phản hồi cho HELP)
220	Server đã sẵn sàng (phản hồi lại nỗ lực thiết lập kết nối TCP của Client)
221	Server đóng kênh truyền
235	Xác thực thành công (phản hồi cho AUTH)
250	Lệnh yêu cầu đã hoàn thành. (mã được theo sau bởi OK)
252	Server không thể xác minh người dùng (phản hồi với VRFY)
334	Phản hồi AUTH lệnh khi cơ chế bảo mật được chấp nhận
421	Server không khả dụng vì đóng kênh truyền
422	Hopo thư của người nhận đã vượt quá giới hạn lưu trữ
431	Quá tải tệp (quá nhiều thư được gửi đến một miền cụ thể)
441	Không có phản hồi từ Server của người nhận

442	Kết nối bị ngắt
446	Vòng lặp nội bộ đã xảy ra
450	Hộp thư không khả dụng (hộp thư bận hoặc tạm thời bị chặn)
452	Server đã huy lệnh do không đủ bộ nhớ hệ thống
454	TLS không khả dụng do một lý do tạm thời (phản hồi cho STARTTLS)
455	Các thông số không được cung cấp
471	Lỗi Server do bộ lọc thư rác cục bộ
500	Lỗi cú pháp (một dòng lệnh có thể quá dài)
501	Lỗi cú pháp trong các tham số hoặc đối số
502	Server chưa triển khai lệnh
503	Chuỗi lệnh không đúng
504	Server chwua triển khai tham số của lệnh
510	Địa chỉ Thư điện tử không hợp lệ
512	Lỗi DNS (kiểm tra lại địa chỉ người nhận)
523	Tổng kích thước thư vượt quá giới hạn máy chủ người nhận
530	Sự cố xác thực chủ yếu yêu cầu lệnh STARTTLS để chạy
535	Quá trình xác thực đã thất bại
538	Cần có mã hóa cho cơ chế xác thực được yêu cầu
541	Thư bị bộ lọc thư rác từ chối
550	Hộp thư không khả dụng. Server đã hủy lệnh vì không thể tìm thấy hộp thư hoặc vì lý do policy
551	Người dùng không phải là local user
552	Server đã hủy bỏ lệnh vì hộp thư đầy
553	Địa chỉ thư không chính xác về mặt cú pháp
554	Giao dịch không thành công do lỗi không xác định
555	Các thông số không được công nhận (phản hồi với MAIL FROM hoặc RCPT TO)

8 Lệnh phản hồi SMTP

Như đã trình bày ở phần trước, một số mã lệnh là cụ thể. Trên thực tế, chỉ có ba trong mã lệnh 500, 501 và 421 là có thể phản hồi với bất kỳ lệnh SMTP nào. Những lệnh khác có thể được phân loại là tích cực hoặc tiêu cực.

Command	Positive response	Negative response

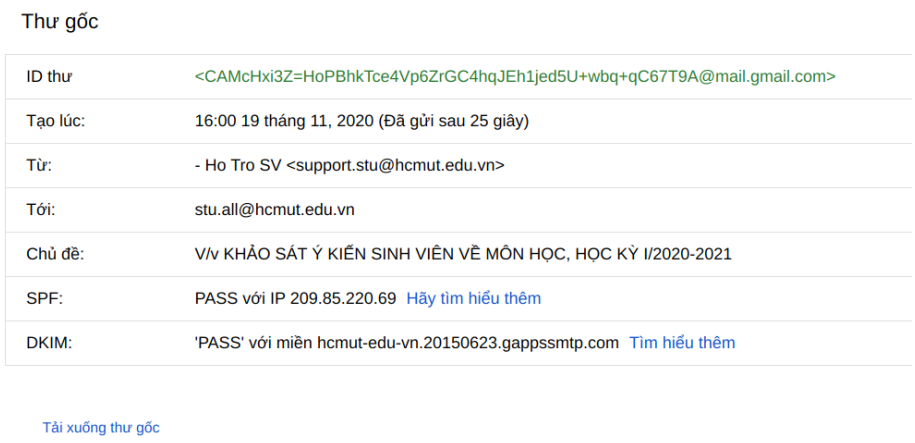
SMTP handshake	220	554
STARTTLS	220	454
EHLO or HELO	250	502,504,550
AUTH	235,334	530,535,538
MAIL FROM	250	451,452,455,503,550,552,553,555
RCPT TO	250,251	450,451,452,455,503,550,551,552,553,555
DATA	250,354	450,451,452,503,550,552,554
QUIT	221	-

9 Định dạng Thư điện tử

RFC 822 đã định nghĩa một Thư điện tử bao gồm phần tiêu đề (Header) và phần thân (Body).

9.1 Header

Phần tiêu đề bao gồm nhiều thông tin, mỗi dòng kết thúc bằng hai ký tự <CRLF>
Phần tiêu đề được chia khỏi phần thân bởi một hàng rỗng.
Mỗi hàng tiêu đề chưa một cặp "Tên": "Giá trị"ngăn bởi dấu ":"



Hình 2: Một ví dụ Header

9.2 Body

Lúc đầu phần thân được quy định có dạng khuôn văn bản đơn giản.
Sau này RFC 822 cập nhật năm 1996 cho phép phần thân mang nhiều loại dữ liệu hơn: audio, video, hình ảnh, tài liệu,... gọi chung là Multipurpose Internet Mail Extensions (MIME).

10 Mã hóa Thư điện tử

Ở phần đầu đã giới thiệu tính năng mã hoá nhằm mục đích bảo mật thông điệp người dùng.
Cả phần Header và Body đều mã hóa dưới dạng ASCII bởi vì để đi được đến đích, mỗi bức thư có thể phải trung chuyển qua nhiều trạm gateway mà các gateway này đều xem thư dưới dạng ASCII, nếu trong thư có bất kỳ kí tự nào khác ASCII thì bức thư sẽ bị đứt gãy nội dung.
Ý tưởng mã hóa sử dụng Base64 là ánh xạ 3 bytes (24 bits) dữ liệu nhị phân nguyên thủy thành 4 ký tự ASCII.
24 bits này được chia thành 4 cụm, mỗi cụm 6 bits được ánh xạ tiếp vào trong 64 ký tự ASCII hợp lệ.

11 Mã hóa TLS/SSL

11.1 Khái niệm

SSL (Secure Sockets Layer) hay TLS (Transport Layer Security) là kỹ thuật mã hóa truyền tin trên internet.
Sử dụng SSL/TLS bằng việc mã hóa dữ liệu truyền tin giữa máy tính và server thì có thể tránh bên thứ ba nghe trộm hoặc giả mạo dữ liệu.[3]

11.2 Sự khác nhau

Cả TLS và SSL đều là giao thức được sử dụng để cung cấp bảo mật giữa trình duyệt web và máy chủ web.
SSL sử dụng Message digest để tạo master secret, đồng thời nó cung cấp các dịch vụ bảo mật cơ bản là Authentication và Confidentiality.
Trong khi đó TLS sử dụng hàm pseudo-random để master secret.

11.3 Kiểm tra

Khi trao đổi thông tin thì việc kiểm tra xem site đó có đối ứng TLS/SSL hay không là rất cần thiết. Ở trang web được cài đặt TLS/SSL thì URL hiển thị ở thanh address được gắn thêm ký tự "s"và đuôi của "http://"thành "https://".

11.4 Giao thức của SSL

Các giao thức của SSL bao gồm:

- SSL record protocol
- Handshake protocol
- Change-cipher spec protocol

- Alert protocol

Giao thức SSL được đặt giữa tầng vận chuyển và tầng ứng dụng, giao thức SSL cung cấp giao thức bảo mật truyền thông có 3 điểm nổi bật:

- Các bên giao tiếp có thể xác thực nhau bằng cách sử dụng mật mã khóa chung.
- Sự bí mật của lưu lượng dữ liệu được bảo vệ vì kết nối được mã hóa trong suốt sau khi một sự thiết lập quan hệ ban đầu và sự thương lượng khóa phiên làm việc đã xảy ra.
- Tính xác thực và tính toàn vẹn của lưu lượng dữ liệu cũng được bảo vệ vì các thông báo được xác thực và được kiểm tra tính toàn vẹn một cách trong suốt bằng cách sử dụng MAC.

11.5 Giao thức TLS

Mục tiêu chính của giao thức TLS là cung cấp sự riêng tư và toàn vẹn dữ liệu giữa hai ứng dụng trong môi trường mạng.
Cũng như giao thức SSL thì giao thức TLS cũng theo mô hình client-server.
Trong mô hình TCP/IP thì giao thức TLS gồm có hai lớp là Record Layer và Handshake Layer. Record Layer là lớp thấp nhất của giao thức TLS nằm trên tầng giao vận như giao thức truyền tải TCP, giao thức truyền vận không tin cậy UDP. Handshake Layer nằm trên Record Layer dùng để định danh của điểm kết nối có thể xác thực bằng cách sử dụng mã hóa bất đối xứng hoặc khóa công khai, quá trình thỏa thuận khóa bí mật chia sẻ được an toàn, quá trình thỏa thuận đáng tin cậy.
Ứng dụng của giao thức TLS bao gồm:

- Đóng gói các giao thức như HTTP, FTP, SMTP, ...
- Cho phép trao đổi riêng tư trên mạng.
- Cho phép các ứng dụng client-server giao tiếp với nhau an toàn.

12 Phiên làm việc của SMTP

SMTP là một giao thức sử dụng các ký tự ASCII. Sau khi thiết lập kết nối TCP đến port 25 của máy đích (được xem là Server), máy nguồn (được xem là Client) chờ nhận kết quả trả về từ Server.
Server khởi đầu cuộc hội thoại bằng cách gửi một dòng văn bản đến Client thông báo danh tính của nó và khả năng nhận thư. Nếu Server không có khả năng nhận thư tại thời điểm hiện tại, Client sẽ hủy bỏ kết nối và thử thiết lập lại sau.
Trong trường hợp nếu Server sẵn sàng nhận thư, Client sẽ thông báo thư đó đến

từ đâu và ai sẽ là người nhận. Nếu người nhận đó tồn tại, Server sẽ thông báo cho Client tiếp tục gửi.
Sau đó Client gửi thư và Server phản hồi đã nhận thư đó. Khi cả hai bên hoàn tất phiên truyền nhận, kết nối sẽ đóng lại.

- Port 25: port không mã hóa
- Port 465/587: port SSL/TSL

12.1 Ví dụ một phiên SMTP thành công

S	220 hamburger.edu
C	EHLO crepes.fr
S	250 Hello crepes.fr
C	MAIL FROM: <>
S	250 ok
C	RCPT TO: <>
S	250 ok
C	DATA
S	354 Start mail input; end with <CRLF>.<CRLF>
S	250 ok
C	QUIT
S	221 hamburger.edu closing connection

12.2 Ví dụ một phiên SMTP bị hủy bỏ

S	220 hamburger.edu
C	EHLO crepes.fr
S	250 Hello crepes.fr
C	MAIL FROM: <>
S	250 ok
C	RCPT TO: <>
S	550 No such user here
C	RSET
S	250 ok
C	QUIT
S	221 hamburger.edu closing connection

13 Phiên làm việc của POP3

Một phiên làm việc theo giao thức POP3 bắt đầu tại User Agent. User Agent khởi tạo một kết nối TCP đến port 110 của Mail Server. Khi đã thực hiện xong kết nối,

phiên làm việc POP3 sẽ trải qua ba giai đoạn:

- Chứng thực
- Giao dịch dữ liệu
- Cập nhật

Giai đoạn chứng thực buộc người dùng phải thực hiện thủ tục đăng nhập bằng cách khai báo tên người dùng và mật khẩu.
Ở giai đoạn giao dịch, người dùng có thể xem danh sách thư chưa nhận về, nhận thư về và xóa thư trong hộp thư của mình khi cần thiết.
Mặc định port của POP3 là:

- Port 110: port không mã hóa
- Port 995: port SSL/TSL

14 Phiên làm việc của IMAP

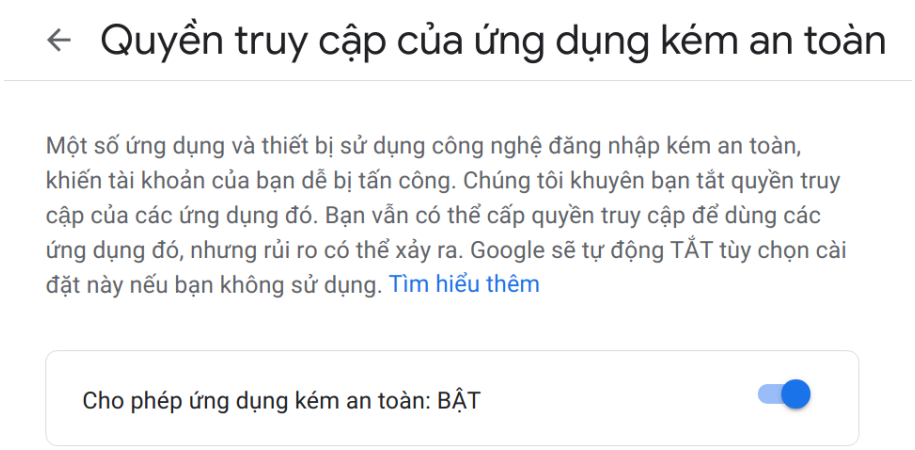
Vì máy tính cá nhân của người dùng không phải lúc nào cũng trong trạng thái mở, khi đó thư điện tử được gửi đến máy chủ và lưu trữ tại đó cho đến khi người dùng bật máy tính và nhận bất kỳ thư mới nào.
Giống như nhiều ứng dụng tiêu chuẩn, Internet Message Access Protocol được mô tả trong tập tài liệu Request For Comment (RFC).[4]
Cách làm việc của IMAP cũng giống như POP3.
IMAP Server lắng nghe trên port 143, cũng nên lưu ý rằng không phải mọi Internet Service Provider đều hỗ trợ cả POP3 và IMAP.
Mặc định port của IMAP là:

- Port 143: port không mã hóa
- Port 993: port SSL/TSL

15 Thiết lập ứng dụng Client cho Google Email

Một số ứng dụng và thiết bị công nghệ truy cập kém an toàn khiến tài khoản thư điện tử dễ bị tấn công, do đó Google khuyến nghị người dùng tắt quyền truy cập cho các ứng dụng đó. Tuy nhiên nếu muốn sử dụng SMTP để đăng nhập và gửi thư điện tử người dùng cần cấp quyền truy cập và chấp nhận rủi ro bằng cách "Cho phép ứng dụng kém an toàn: BẬT".
Để đọc thư Google Mail trên các trình thư điện tử máy khách bằng cách sử dụng POP, tuân thủ tài liệu hỗ trợ của Google:

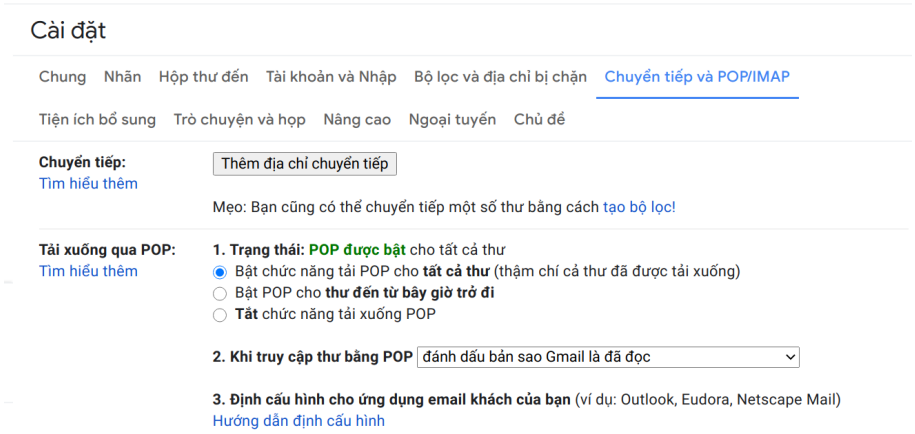
1. Xác nhận POP là cách tốt nhất để đọc thư điện tử:
   IMAP và POP là hai cách để đọc thư điện tử của Google trong ứng dụng



Hình 3: Cấp quyền truy cập

khác. IMAP có thể được sử dụng trên nhiều thiết bị và thư điện tử được đồng bộ hóa theo thời gian thực. POP chỉ có thể được sử dụng cho một máy tính duy nhất. Thư điện tử không được đồng bộ hóa theo thời gian thực, thay vào đó chúng được tải xuống và người dùng phải quyết định mức độ thường xuyên muốn tải thư điện tử mới xuống.

2. Thiết lập POP:
   Thiết lập POP trong Google mail.



Hình 4: Thiết lập POP Gmail

Tiếp theo thực thiện thiếp lập POP trên ứng dụng máy khách.

- Máy chủ thư đến (POP): Server (pop.gmail.com), Yêu cầu SSL (có), Port (995).
- Máy chủ thư đi (SMTP): Server (smtp.gmail.com), Yêu cầu SSL (có), Yêu cầu TLS (nếu có), Yêu cầu xác thực (có), Port (TLS/STARTTLS 587).
- Thời gian chờ máy chủ: nhiều hơn 1 phút (khuyến nghị là 5 phút).

- Tên đầy đủ hoặc tên hiển thị: tên người dùng
- Tài khoản hoặc địa chỉ email: địa chỉ email của người dùng.
- Mật khẩu: mật khẩu google mail của người dùng.

Để sử dụng IMAP, thiết lập tương tự như bước thiết lập POP trong tài khoản Google mail. Tiếp đến là thiết lập IMAP trong ứng dụng máy khách:

- Máy chủ thư đến (IMAP): Server (imap.gmail.com), Yêu cầu SSL (có), Port (993).
- Máy chủ thư đi (SMTP): Server (smtp.gmail.com), Yêu cầu SSL (có), Yêu cầu TLS (nếu có), Yêu cầu xác thực (có), Port SSL (465), Port TLS/STARTTLS (587)
- Tên đầy đủ hoặc tên hiển thị: tên người dùng
- Tài khoản hoặc địa chỉ email: địa chỉ email của người dùng.
- Mật khẩu: mật khẩu google mail của người dùng.

16 Lập trình mô phỏng với thư viện Python

Có nhiều ứng dụng có thể gửi và nhận dữ liệu trên network được viết trên nền một hoặc nhiều ngôn ngữ khác nhau. Theo đó có nhiều ngôn ngữ lập trình hỗ trợ thư viện để có thể dễ dàng viết một ứng dụng để gửi và nhận dữ liệu. Với một thư viện lập trình tốt, việc tạo một kết nối đến một ứng dụng đang chạy trên server, gửi dữ liệu đến server và nhận dữ liệu từ server cũng như đọc và ghi file rất đơn giản.

16.1 Thư viện socket

socket là một thư viện phần mềm có sẵn với nhiều ngôn ngữ lập trình cho phép tạo một kết nối trên network và trao đổi dữ liệu dễ dàng.
Thư viện socket trả về một đối tượng socket có các phương thức thực hiện các lệnh hệ thống socket khác nhau.
Lập trình socket là một cách kết nối hai nút trên mạng để giao tiếp với nhau. Một socket (nút) lắng nghe trên một cổng cụ thể tại một địa chỉ IP, trong khi socket khác kết nối với socket kia để tạo kết nối. Máy chủ tạo thành socket lắng nghe trong khi máy khách tiếp cận với máy chủ. Đó là nguyên tắc cơ bản đằng sau sự duyệt web.
Định dạng địa chỉ được yêu cầu bởi một đối tượng socket cụ thể được chọn tự động dựa trên họ địa chỉ được chỉ định khi đối tượng socket được tạo. Địa chỉ socket được biểu diễn như sau:

- AF_UNIX: socket liên kết với hệ thống được biểu diễn dưới dạng một chuỗi. Trước đây ở phiên bản 3.3 các đường dẫn AF_UNIX được giả định sử dụng mã hóa UTF-8. Phiên bản 3.5 hiện tại chấp nhận cả đối tượng byte.
- AF_INET: một cặp giá trị được sử dụng cho họ địa chỉ, trong đó máy chủ lưu trữ là một chuỗi đại diện cho tên máy chủ trong miền Internet hoặc như địa chỉ IPv4 và port là một số nguyên (host,port).

Lập trình socket được bắt đầu bằng cách nhập thư viện socket và tạo một socket đơn giản.
Trong đó, mặc định family là AF_INET đề cập đến địa chỉ IPv4 và type mặc định là SOCK_STREAM chỉ định giao thức connection-oriented TCP có điểm đầu và điểm cuối, theo trình tự và duy nhất, sẽ hủy kết nối cũng như phát hiện lỗi. Ngoài ra còn có giao thức SOCK_DGRAM là một loại mạng có kết nối ít điểm để gửi và nhận gói tin. Nó tương tự như hộp thư, các dữ liệu được gửi vào hộp thư gửi và truyền đến hộp thư nhận.
Trong kiến trúc máy chủ-máy khác, có một máy chủ tập trung cung cấp dịch vụ và nhiều máy khách nhận dịch vụ từ máy chủ tập trung đó. Các máy khách cũng thực hiện yêu cầu đối với máy chủ. Một số phương thức socket máy chủ quan trọng trong kiến trúc này như sau:

- socket.bind(): Phương thức này liên kết địa chỉ(tên host,port) với socket.
- socket.listen(): Phương thức này về cơ bản lắng nghe các kết nối được thực hiện với socket. Nó khởi động trình nghe của TCP.
- socket.accept(): Phương thức này chấp nhận kết nối máy khách TCP. Cặp (conn,address) là một cặp giá trị trả về của phương thức này. Ở đây, conn là một đối tượng socket mới được sử dụng để gửi và nhận dữ liệu trên kết nối và address là địa chỉ liên kết với socket. Trước khi sử dụng phương thức này, phương thức socket.bind() và socket.listen() phải được sử dụng.

Máy khách trong kiến trúc máy khách-máy chủ yêu cầu từ máy chủ và nhận các dịch vụ từ máy chủ. Do đó chỉ có một phương thức dành riêng cho máy khách:

- socket.connect(address): Phương thức này chủ động kết nối máy chủ, đối số đại diện cho địa chỉ của máy chủ cần kết nối.

Các phương thức socket chung của máy khách và máy chủ rất hữu ích:

- socket.recv(bufsize): Phương thức nhận thông điệp TCP từ socket. Đối số bufsize là viết tắt của kích thước bộ đệm và xác định dữ liệu tối đa mà phương thức này có thể nhận bất kỳ lúc nào đơn vị là byte.
- socket.send(byte): Phương thức này được sử dụng để gửi dữ liệu đến socket được kết nối với máy từ xa. Đối số byte sẽ cung cấp số byte được gửi đến socket.

- socket.close(): Phương thức này sẽ đóng socket.
- socket.sendall(data): Phương thức này gửi tất cả dữ liệu đến socket được kết nối với một máy từ xa. Nó sẽ chuyển dữ liệu một cách bất cẩn đến khi xảy ra lỗi và nếu xảy ra lỗi thì nó sẽ sử dụng phương thức socket.close() để đóng socket.

```
Server : 220 smtp.gmail.com ESMTP q200sm582358pfq.95 - gsmtp

Client : HELO ComputerNetwork
Server : 250 smtp.gmail.com at your service

Client : AUTH Login
Server : 334 VXNlcm5hbWU6

Client : MAIL FROM: <remitai1998@gmail.com>
Server : 250 2.1.0 OK q200sm582358pfq.95 - gsmtp

Client : RCPT TO: <remitai1998@gmail.com>
Server : 250 2.1.5 OK q200sm582358pfq.95 - gsmtp

Client : DATA
Server : 354  Go ahead q200sm582358pfq.95 - gsmtp

Client : I love Computer Network
.

Server : 250 2.0.0 OK  1606795339 q200sm582358pfq.95 - gsmtp

Client : QUIT
Server : 221 2.0.0 closing connection q200sm582358pfq.95 - gsmtp


Process finished with exit code 0
```

Hình 5: Mô phỏng phiên làm việc gửi thư điện tử

16.2 Thư viện ssl

Thư viện này cung cấp một trình bao bọc socket gồm chức năng mã hóa và giải mã dữ liệu đi qua socket bằng kỹ thuật SSL.
Đối với các ứng dụng phức tạp, ssl.SSLContext giúp quản lý và cấu hình, sau đó có thể được kế thừa bởi các socket SSL được tạo thông qua phương thức SSLContext.wrap_socket()

16.3 Thư viện poplib

Thư viện này định nghĩa một class POP3 đóng gói kết nối đến máy chủ POP3 và triển khai giao thức như được định nghĩa trong RFC năm 1939.

Ngoài ra thư viện này còn cung cấp class POP3_SSL cung cấp hỗ trợ kết nối với máy chủ POP3 sử dụng kỹ thuật SSL làm lớp giao thức cơ bản.
Một số phương thức trong thư viện:

- user: gửi lệnh cho người dùng, phản hồi cho biết bắt buộc mật khẩu.
- pass_: gửi mật khẩu, phản hồi bao gồm số lượng tin nhắn và kíhc thước hộp thư.
- list: yêu cầu danh sách tin nhắn.
- retr: lấy toàn bộ thông điệp đó.

```
Server : +OK Gpop ready for requests from 14.226.229.50 g26mb126015933ivp
Client : user remitai1998@gmail.com
Server : +OK message follows
Client : list
Server : 86
Client : retr 86
Server : Bcc: remitai1998@gmail.com
Return-Path: <remitai1998@gmail.com>
Received: from ComputerNetwork ([14.226.229.50])
        by smtp.gmail.com with SMTPSA id x28sm640839pfr.186.2020.11.30.20.27.31
        for <remitai1998@gmail.com>
        (version=TLS1_3 cipher=TLS_AES_256_GCM_SHA384 bits=256/256);
        Mon, 30 Nov 2020 20:27:32 -0800 (PST)
Message-ID: <5fc5c634.1c69fb81.7e33b.1e5b@mx.google.com>
Date: Mon, 30 Nov 2020 20:27:32 -0800 (PST)
From: remitai1998@gmail.com

I love Computer Network
Client : quit
Server : +OK POP3 server signing off

Process finished with exit code 0
```

Hình 6: Mô phỏng xem thư điện tử với poplib

16.4 Thư viện imaplib

Thư viện này bao gồm ba class là: IMAP4, IMAP4_SSL và IMAP4_stream.
Trong đó gói gọn một kết nối đến một máy chủ IMAP4 và thực hiện một tập hợp các giao thức được quy định tại RFC 2060 cho phép người dùng xem và thao tác với các thư.
Sử dụng classs IMAP4_SSL để kết nối an toàn, đây là class con có nguồn gốc từ class IMAP4 kết nối qua socket được mã hóa SSL (để sử dụng class này, cần có module socket được biên dịch với hỗ trợ SSL).
Nếu máy chủ không chỉ định thì máy chủ localhost sẽ được sử dụng. Nếu Port không chỉ định thì Port tiêu chuẩn là 993 được sử dụng.
Để đăng nhập vào hộp thư và có toàn quyền truy cập vào tất cả thư điện tử sử dụng phương thức "login".

Tiếp theo là thao tác chọn hộp thư cần đến với phương thức "select". Nếu người dùng chỉ định hộp thư đến thì câu lệnh sẽ là select("Inbox").
Sử dụng phương thức "search"để tìm kiếm thư, phương thức này trả về hai đối số là "type"phản hồi yêu cầu đó có "ok"hay không, nếu "ok"nghĩa là yêu cầu thành công. Trong khi đó đối số còn lại là "data"là id của tất cả các thư điện tử.
Phương thức "fetch"để tìm nạp thư cho id của thư, ở đó "RFC822"là giao thức truy cập tin nhắn trên Internet.

```
Bcc: remitai1998@gmail.com
Return-Path: <remitai1998@gmail.com>
Received: from ComputerNetwork ([14.226.229.50])
        by smtp.gmail.com with SMTPSA id 143sm307791pfc.119.2020.11.30.18.11.25
        for <remitai1998@gmail.com>
        (version=TLS1_3 cipher=TLS_AES_256_GCM_SHA384 bits=256/256);
        Mon, 30 Nov 2020 18:11:26 -0800 (PST)
Message-ID: <5fc5a64e.1c69fb81.8d0b9.0f65@mx.google.com>
Date: Mon, 30 Nov 2020 18:11:26 -0800 (PST)
From: remitai1998@gmail.com

I love Computer Network


Process finished with exit code 0
```

Hình 7: Mô phỏng xem thư điện tử với imaplib

Tài liệu

[1] Charles Severance. *Introduction to Networking*

[2] James F. Kurose, Keith W. Ross. *Computer Networking A Top Down Approach Featuring the Internet, 3rd edition*

[3] GeoTrust. https://www.geotrust.co.jp/ssl_guideline/ssl_beginners/

[4] RFC3501. https://tools.ietf.org/html/rfc3501