HTTP là chữ viết tắt của HyperText Transfer Protocol (giao thức truyền tải siêu văn bản). Đây là một giao thức [ứng dụng](http://www.techz.vn/C/ung-dung) trong bộ các giao thức TCP/IP (gồm một nhóm các giao thức nền tảng cho internet).

HTTP hoạt động dựa trên mô hình Client – Server. Trong mô hình này, các máy tính của người dùng sẽ đóng vai trò làm máy khách (Client). Sau một thao tác nào đó của người dùng, các máy khách sẽ gửi yêu cầu đến máy chủ (Server) và chờ đợi câu trả lời từ những máy chủ này. Để có thể nói chuyện được với nhau, các máy chủ và máy khách phải thực hiện việc trao đổi thông qua các giao thức. Một trong những giao thức được sử dụng thường xuyên nhất chính là HTTP.

Khi bạn gõ một địa chỉ Web URL vào trình duyệt Web, một lệnh HTTP sẽ được gửi tới Web server để ra lệnh và hướng dẫn nó tìm đúng trang Web được yêu cầu. Trang Web này sau đó sẽ được kéo về và mở trên trình duyệt Web. Nói đơn giản hơn, HTTP là giao thức giúp cho việc truyền tải file từ một Web server vào một trình duyệt Web để người dùng có thể xem một trang Web đang hiện diện trên trình duyệt.

**ẨN HỌA TỪ HTTP VÀ SỰ CẦN THIẾT CỦA HTTPS**

Được sử dụng một cách rộng rãi là vậy nhưng HTTP chứa đựng trong nó không ít những điểm hạn chế. Khi bạn tiến hành thực hiện việc truy nhập vào một trang Web thông 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.

**Trong quá trình kết nối và trao đổi thông tin, trình duyệt của bạn sẽ mặc nhiên thừa nhận địa chỉ IP đó đến từ Server của chính Website mà bạn muốn truy nhập mà không hề có biện pháp xác thực nào. Các thông tin được chuyển đi qua giao thức HTTP (bao gồm địa chỉ IP của bạn, các thông tin mà bạn nhập liệu trên Website…) cũng không hề được mã hóa và bảo mật.**

Điều này dẫn đến những nguy cơ về việc phiên kết nối của bạn tới máy chủ của Website có thể bị “nghe lén”, hoặc việc truy nhập của bạn bị chuyển hướng đến một trang Web giả danh với thiết kế giống hệt Website gốc mà người sử dụng không hề hay biết.

Nếu bạn thường xuyên sử dụng các dịch vụ ngân hàng trực tuyến, bạn sẽ thấy khi truy nhập vào địa chỉ tên miền của ngân hàng, giao thức mà nó sử dụng sẽ là HTTPS thay vì HTTP như ở những trang web thông thường. Điều này là bởi, phiên bản nâng cấp HTTPS của HTTP được sử dụng nhằm tăng cường khả năng bảo mật thông tin sau mỗi lần truy nhập.

HTTPS là tên viết tắt của "Hypertext Transfer Protocol Secure". Đây là một sự kết hợp giữa giao thức HTTP và giao thức bảo mật SSL hay TLS**. HTTPS giúp cho việc trao đổi thông tin một cách bảo mật trên nền Internet.**

Khác với HTTP, HTTPS sẽ hỗ trợ việc xác thực tính chính danh của các Website mà người dùng truy nhập 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) có uy tín. Với các xác thực từ CA, người sử dụng có thể biết rằng mình đã truy nhập đúng vào Website cần truy nhập chứ không phải một Website giả danh bất kỳ nào khác.

Bên cạnh đó, **các phiên kết nối giữa trình duyệt của bạn đến Server đều sẽ được** **mã hóa**. Điều này sẽ giúp che giấu địa chỉ IP của bạn và những thông tin nhập liệu về tài khoản của bạn trên Website khỏi sự nhòm ngó của các hacker. HTTPS không đem đến sự an toàn 100%. Tuy vậy, đây là biện pháp bảo mật hữu hiệu thay vì việc sử dụng giao thức HTTP truyền thống vốn đầy rủi ro sẵn có.

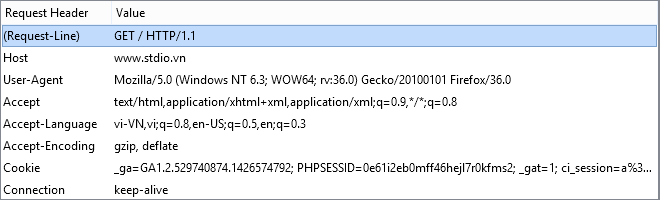
Để kiểm tra một Website có sử dụng giao thức HTTPS hay không cũng vô cùng đơn giản. Các bạn chỉ cần để ý đến phần link đường dẫn khi truy nhập vào Website mở đầu bằng http:// hay https://. Bên cạnh đó, link đường dẫn của các Website có sử dụng giao thức **HTTPS thường đi kèm với một biểu tượng nhỏ hình ổ khóa**. Khi đưa con trỏ chuột hướng vào biểu tượng này, trên đó sẽ hiện ra tên của đơn vị xác thực (CA) như đã nói ở trên. Đây là dấu hiệu cho thấy Website mà bạn đang truy nhập không phải là giả mạo.

-----------------------

Như chúng ta đã biết, HTTP là giao thức được thiết kế theo kiểu client – server, giao tiếp giữa client và server dựa vào một cặp request – response, client đưa ra các request và server trả lời các request này.

### HTTP Request

Để bắt đầu trao đổi dữ liệu, phía client khởi tạo một HTTP session bằng cách mở một kết nối TCP đến HTTP server sau đó gửi request đến server này. Request có thể được tạo bằng nhiều cách, trực tiếp khi người dùng nhấp vào một liên kết trên trình duyệt hoặc gián tiếp, ví dụ như một video được đính kèm trong một website và việc request đến website này sẽ dẫn đến một request tới video ấy.

  
Ví dụ một HTTP Request

Bắt đầu của HTTP Request sẽ là dòng Request-Line bao gồm 3 thông tin đó là:

* Method: là phương thức mà HTTP Request này sử dụng, thường là GET, POST, ngoài ra còn một số phương thức khác như HEAD, PUT, DELETE, OPTION, CONNECT. Trong ví dụ trên là GET
* URI: là địa chỉ định danh của tài nguyên. Trong tường hợp này URI là / - tức request cho tài nguyên gốc, nếu request không yêu cầu một tài nguyên cụ thể, URI có thể là dấu \*.
* HTTP version: là phiên bản HTTP đang sử dụng, ở đây là HTTP 1.1.

Tiếp theo là các trường request-header, cho phép client gửi thêm các thông tin bổ sung về thông điệp HTTP request và về chính client. Một số trường thông dụng như:

* Accept: loại nội dung có thể nhận được từ thông điệp response. Ví dụ: text/plain, text/html…
* Accept-Encoding: các kiểu nén được chấp nhận. Ví dụ: gzip, deflate, xz, exi…
* Connection: tùy chọn điều khiển cho kết nối hiện thời. Ví dụ: keep-alive, Upgrade…
* Cookie: thông tin HTTP Cookie từ server.
* User-Agent: thông tin về user agent của người dùng.

#### Phương thức GET và POST

Hai phương thức được sử dụng nhiều nhất trong HTTP request là GET và POST

Với GET, câu truy vấn sẽ được đính kèm vào đường dẫn của HTTP request. Ví dụ: **/?username=”abc”&password=”def”**

Một số đặc điểm của phương thức GET:

* GET request có thể được cached, bookmark và lưu trong lịch sử của trình duyệt.
* GET request bị giới hạn về chiều dài, do chiều dài của URL là có hạn.
* GET request không nên dùng với dữ liệu quan trọng, chỉ dùng để nhận dữ liệu.

Ngược lại, với POST thì câu truy vấn sẽ được gửi trong phần message body của HTTP request, một số đặc điểm của POST như:

* POST không thể, cached, bookmark hay lưu trong lịch sử trình duyệt.
* POST không bị giới hạn về độ dài.

Các phương thức khác

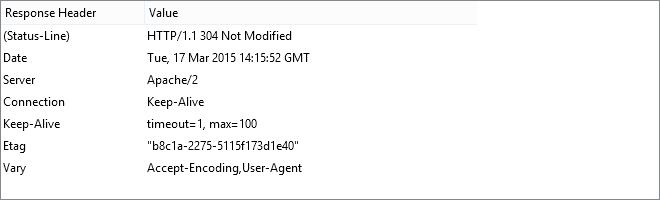
Ngoài GET và POST, HTTP request còn có thể có một số phương thức khác như:

* HEAD:; giống như GET nhưng chỉ gửi về HTTP header.
* PUT: tải lên một mô tả về URI định trước.
* DELETE: xóa một tài nguyên định trước.
* OPTIONS: trả về phương thức HTTP mà server hỗ trợ.
* CONNECT: chuyển kết nối của HTTP request thành một kết nối HTTP tunnel.

### HTTP Response

Cấu trúc HTTP response gần giống với HTTP request, chỉ khác nhau là thay vì Request-Line, thì HTTP có response có Status-Line. Và giống như Request-Line, Status-Line cũng có ba phần như sau:

* HTTP-version: phiên bản HTTP cao nhất mà server hỗ trợ.
* Status-Code: mã kết quả trả về.
* Reason-Phrase: mô tả về Status-Code.

  
Ví dụ một HTTP Response

#### HTTP Status Codes

Một số loại Status-Code thông dụng mà server trả về cho client như sau:

**1xx: information Message:**các status code này chỉ có tính chất tạm thời, client có thể không quan tâm.

**2xx Successful:** khi đã xử lý thành công request của client, server trả về status dạng này:

* 200 OK: request thành công.
* 202 Accepted: request đã được nhận, nhưng không có kết quả nào trả về, thông báo cho client tiếp tục chờ đợi.
* 204 No Content: request đã được xử lý nhưng không có thành phần nào được trả về.
* 205 Reset: giống như 204 nhưng mã này còn yêu câu client reset lại document view.
* 206 Partial Content: server chỉ gửi về một phần dữ liệu, phụ thuộc vào giá trị range header của client đã gửi.

**3xx Redirection:**server thông báo cho client phải thực hiện thêm thao tác để hoàn tất request:

* 301 Moved Permanently: tài nguyên đã được chuyển hoàn toàn tới địa chỉ Location trong HTTP response.
* 303 See other: tài nguyên đã được chuyển tạm thời tới địa chỉ Location trong HTTP response.
* 304 Not Modified: tài nguyên không thay đổi từ lần cuối client request, nên client có thể sử dụng đã lưu trong cache.

**4xx Client error:**lỗi của client:

* 400 Bad Request: request không đúng dạng, cú pháp.
* 401 Unauthorized: client chưa xác thực.
* 403 Forbidden: client không có quyền truy cập.
* 404 Not Found: không tìm thấy tài nguyên.
* 405 Method Not Allowed: phương thức không được server hỗ trợ.

**5xx Server Error:**lỗi của server:

* 500 Internal Server Error: có lỗi trong quá trình xử lý của server.
* 501 Not Implemented: server không hỗ trợ chức năng client yêu cầu.
* 503: Service Unavailable: Server bị quá tải, hoặc bị lỗi xử lý.