|  |  |
| --- | --- |
| logo (CMYK)-01 | **BỘ GIÁO DỤC VÀ ĐÀO TẠO** **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM** |

**ĐỒ ÁN AN TOÀN MÁY CHỦ WINDOW**

**TÌM HIỂU VÀ TRIỂN KHAI**

**GIAO THỨC HTTP & HTTPS**

Ngành: CÔNG NGHỆ THÔNG TIN

Chuyên ngành: CÔNG NGHỆ PHẦN MỀM

Giảng viên hướng dẫn: Tống Thanh Văn

Sinh viên thực hiện:

Trần Chấn Hưng (nhóm trưởng) MSSV:1911065247

Ngô Nguyễn Ngọc Thành MSSV:1911160680

Lê Hoàng Minh Tuấn MSSV:1911065880

Võ Minh Phi MSSV:1911065836

TP. Hồ Chí Minh, 2022

**NHẬN XÉT CỦA GIẢNG VIÊN**

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

**MỤC LỤC**

[LỜI MỞ ĐẦU 1](#_Toc1889)

[Chương I. GIAO THỨC HTTP 2](#_Toc14173)

[1. Khái niệm. 2](#_Toc70)

[1.1. HTTP là gì? 2](#_Toc1505)

[1.2. Cấu trúc của HTTP. 3](#_Toc21311)

[1.3. Đặc trưng cơ bản của HTTP. 8](#_Toc4875)

[1.4. Cách thức hoạt động của HTTP. 9](#_Toc21667)

[1.5. Công cụ kiểm tra API với giao thức HTTP. 10](#_Toc25301)

[Chương II. GIAO THỨC HTTPS 11](#_Toc32130)

[1. Khái niệm. 11](#_Toc25379)

[1.1. HTTPS là gì? 11](#_Toc16631)

[1.2. HTTPS hoạt động như thế nào? 12](#_Toc11331)

[1.3. Bảo mật PKI – Public Key Infrastructure 13](#_Toc2356)

[2. Chứng chỉ SSL/TLS 14](#_Toc20055)

[2.1. Cơ chế hoạt động của SSL/TLS 15](#_Toc443)

[2.2. Các thuật toán dùng trong SSL 16](#_Toc23317)

[2.3. Lợi ích của SSL 16](#_Toc16550)

[2.4. Kiểm tra SSL trên trang web 17](#_Toc5763)

[2.5. Thiết lập SSL trên trang web 17](#_Toc12015)

[2.6. Chi phí SSL 19](#_Toc20119)

[Chương III. Sự khác biệt của HTTP và HTTPs 20](#_Toc20488)

[1. Sự khác biệt 20](#_Toc6378)

[2. Chứng chỉ SSL 20](#_Toc19283)

[3. Port trên HTTP và HTTPS 21](#_Toc28376)

[4. Mức độ bảo mật của HTTP và HTTPS 21](#_Toc28215)

[4.1. Có nên sử dụng HTTPS cho website của bạn? 21](#_Toc9154)

[Chương IV. Tiến hành đánh giá và triển khai 25](#_Toc21466)

[1. Sơ đồ, kịch bản 25](#_Toc20052)

[1.1. Sơ đồ cấu hình bằng Cisco 25](#_Toc2006)

[1.2. Kịch bản 27](#_Toc17652)

[2. Công cụ VMware 29](#_Toc25436)

[2.1. Cài đặt máy chủ hệ điều hành ảo trong VMware 29](#_Toc3791)

[2.2. Cài đặt dịch vụ trong Windows Server 2012 31](#_Toc8352)

[3. Các bước thực hiện 32](#_Toc30237)

[4. Kết quả 35](#_Toc16011)

[4.1. Kết quả sau khi tấn công Http dos flood 35](#_Toc3460)

[4.2. Thử nghiệm phân tích mạng qua Wireshark 36](#_Toc24316)

[5. Kết luận và phát triển 36](#_Toc18904)

[Chương V. TÀI LIỆU KHAM KHẢO 37](#_Toc23592)

# LỜI MỞ ĐẦU

Nhóm chúng em xin được gửi lời cảm ơn sâu sắc đến thầy Tống Thanh Văn- là giảng viên môn An Toàn Máy Chủ Window đã trực tiếp giảng dạy, luôn giúp đỡ, giải đáp, truyền đạt để nhóm em hoàn thành bài báo cáo tốt nhất.

Nhóm em xin chân thành cảm ơn sự chỉ bảo của thầy trong suốt quá trình thực hiện bài môn học. Khi bắt tay vào thực hiện bài báo cáo thì kiến thức và sự hiểu biết của em về đề tài này còn rất hạn hẹp. Với những kiến thức và sự nhiệt tình, thầy đã chỉ bảo chúng em đi đến những bước cuối cùng để hoàn thành được bài báo cáo đảm bảo nội dung và đúng thời hạn đề ra.

Và nhóm em cũng không quên cảm ơn đến những người bạn, những người đã gắn bó suốt quá trình làm đề tài, những người đã giúp đỡ em rất nhiều trong quá trình thực hiện hoàn thành bài đồ án này.

Một lần nữa nhóm chúng em xin được gửi lời cảm ơn sâu sắc đến thầy và bạn bè đã giúp đỡ chúng em hoàn thành bài báo cáo này!

Chúng em chân thành cảm ơn!

1. GIAO THỨC HTTP
   1. Khái niệm.
      1. HTTP là gì?

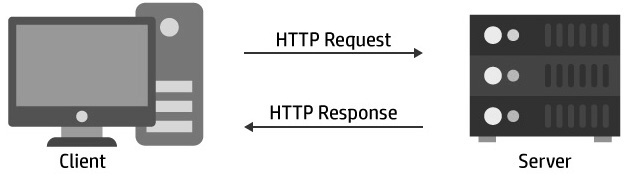


HTTP là chữ viết tắt từ HyperText Transfer Protocol (giao thức truyền tải siêu văn bản). Nó là giao thức cơ bản mà World Wide Web sử dụng. HTTP xác định cách các thông điệp (các file văn bản, hình ảnh đồ hoạ, âm thanh, video, và các file multimedia khác) được định dạng và truyền tải ra sao, và những hành động nào mà các Web server (máy chủ Web) và các trình duyệt Web (browser) phải làm để đáp ứng các lệnh rất đa dạng. Chẳng hạn, 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 và kéo về mở trên trình duyệt Web. Nói nôm na hơn, HTTP là giao thức truyền tải các file từ một Web server vào một trình duyệt Web để người dùng có thể xem một trang Web đang hiện diện trên Internet. HTTP là một giao thức ứng dụng của bộ giao thức TCP/IP (các giao thức nền tảng cho Internet).

HTTP lần đầu được giới thiệu vào những năm 90. Cho đến ngày nay, nó không ngừng được mở rộng và chiếm một vị trí rất quan trọng trong thế giới Internet. HTTP được coi như là một giao thức ứng dụng của bộ các giao thức nền tảng cho Internet TCP/IP. Nó cũng có thể được gửi thông qua kết nối TCP được mã hóa TLS.

Nhờ vào đặc tính mở rộng không giới hạn của nó, ngoài các tài liệu siêu văn bản, HTTP còn được dùng để tìm nạp hình ảnh và video. Thậm chí là đăng tải cả nội dung lên server. Chính vì thế, HTTP đóng vai trò quan trọng trong quy trình cập nhật website thông qua khả năng nạp dữ liệu cho các doc theo yêu cầu.

* + 1. Cấu trúc của HTTP.



Hình 1.2: Sơ đồ của HTTP.

Cấu trúc của HTTP bao gồm 2 đối tượng là Client và Server. Có thể coi HTTP như giao thức gửi các yêu cầu và phản hồi giữa Client - Server. Tại giao thức này, mọi thiết bị tìm kiếm hay trình duyệt web sẽ đóng vai trò như máy khách, còn máy chủ web có vai trò như Server.

* **Client**: Client (máy khách) gửi yêu cầu cụ thể đến Server theo mẫu phương thức yêu cầu -> Các phiên bản giao thức cùng với URI gửi thông báo MIME (gồm thông tin máy khách, nội dung của đối tượng, bộ chỉnh sửa) đến server qua kết nối TCP/IP.
* **Server**: Server nhận được yêu cầu -> Phản hồi lại bằng một dòng trạng thái qua thông báo MIME có chứa thông tin máy chủ, thông tin về nội dung của đối tượng và thực thể của đa phương tiện.

1.2.1 HTTP Request.

1.2.1.1 Về khái niệm.

HTTP Request là thông tin được gửi từ client lên server, để yêu cầu server tìm hoặc xử lý một số thông tin, dữ liệu mà client muốn. HTTP request có thể là một file text dưới dạng XML hoặc Json mà cả hai đều có thể hiểu được. Request URL: Điều này trỏ đến tài nguyên trên web.

1.2.1.2 Các phương thức của HTTP Request

**Phương thức GET**

* Get là phương thức được Client gửi dữ liệu lên Server thông qua đường dẫn URL nằm trên thanh địa chỉ của Browser. Server sẽ nhận đường dẫn đó và phân tích trả về kết quả cho bạn. Hơn nữa, nó là một phương thức được sử dụng phổ biến mà không cần có request body.
* Một số đặc điểm chính của phương thức Get là:
* Giới hạn độ dài của các giá trị là 255 kí tự.
* Chỉ hỗ trợ các dữ liệu kiểu String.
* Có thể lưu vào bộ nhớ cache.
* Các tham số truyền vào được lưu trữ trong lịch sử trình duyệt.
* Có thể được bookmark (đánh dấu rồi xem lại sau) do được lưu trong lịch sử trình duyệt.

**Phương thức POST**

* Phương thức Post là phương thức gửi dữ liệu đến server giúp bạn có thể thêm mới dữ liệu hoặc cập nhật dữ liệu đã có vào database.
* Chúng ta sẽ gửi thông tin cần thêm hoặc cập nhật trong phần body request.
* Một số đặc điểm chính của Post là:
  + Dữ liệu cần thêm hoặc cập nhật không được hiển thị trong URL của trình duyệt.
  + Dữ liệu không được lưu trong lịch sử trình duyệt.
  + Không có hạn chế về độ dài của dữ liệu.
  + Hỗ trợ nhiều kiểu dữ liệu như: String, binary, integers,…

**Phương thức PUT**

* Cách hoạt động tương tự như Post nhưng nó chỉ được sử dụng để cập nhật dữ liệu đã có trong database. Khi sử dụng nó, bạn phải sửa toàn bộ dữ liệu của một đối tượng.

**Phương thức PATCH**

* Tượng tự như Post và Put, nhưng Patch được sử dụng khi phải cập nhật một phần dữ liệu của đối tượng.

**Phương thức DELETE**

* Giống như tên gọi, khi sử dụng phương thức Delete sẽ xoá các dữ liệu của server về tài nguyên thông qua URI. Cũng giống như GET, phương thức này không có body request.

**Phương thức HEAD**

* HEAD gần giống giống với lại GET, tuy nhiên nó không có response body.
* Nói một cách khác, nếu sử dụng phương thức GET tới đường dẫn /Books thì sẽ trả về danh sách các sản phẩm, còn khi sử dụng HEAD tới đường dẫn /Books nhưng không nhận được danh sách các sản phẩm.
* Truy vấn HEAD hữu ích khi chúng ta sử dụng nó để kiểm tra API có hoạt động không do không có response body nên thời gian phản hồi nhanh hơn so với phương thức Get. Và thường được sử dụng để kiếm tra trước khi download file do cứ gọi đến api dowload sẽ download file nên thêm phương thức head vào nó kiểm tra xem api có đang hoạt động tốt không tránh down nhiều.

1.2.1.3 Về Cấu trúc

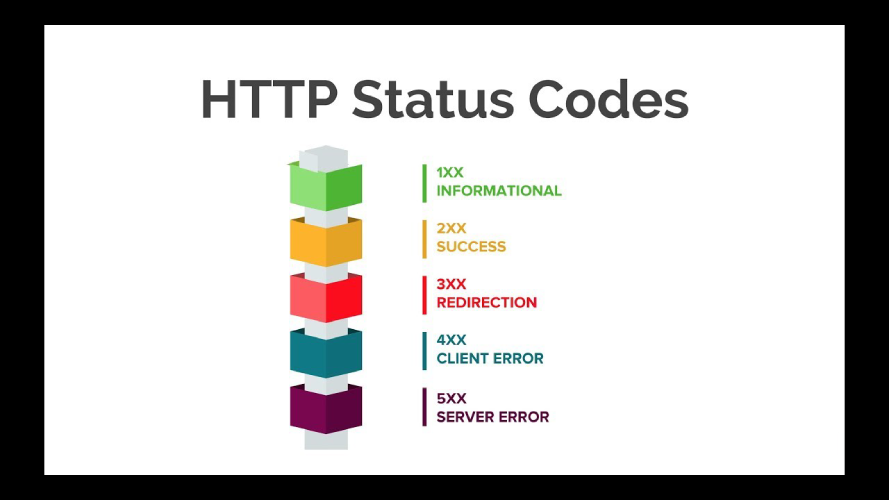
Cấu trúc của HTTP Request bao gồm:

* Request Method: Điều này cho biết hành động cụ thể mà Request mong đợi nhận được từ máy chủ trong Response của nó.
* Request Header: Điều này bao gồm dữ liệu như loại trình duyệt đang được sử dụng và dữ liệu mà Request đang tìm kiếm từ máy chủ. Nó cũng có thể bao gồm cookie, hiển thị thông tin được gửi trước đó từ máy chủ xử lý Request.
* Request Body: Đây là thông tin tùy chọn mà máy chủ cần từ Request, chẳng hạn như biểu mẫu người dùng – đăng nhập tên người dùng / mật khẩu, Response ngắn và tải lên tệp – đang được gửi đến trang web.

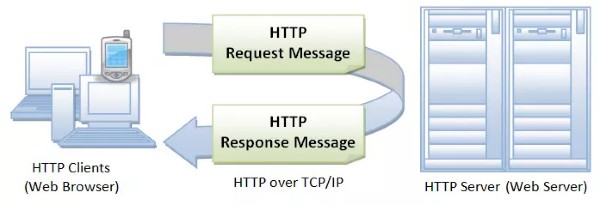
1.2.2 HTTP Response.

HTTP Reponse làgói thông tin được gửi bởi máy chủ đến máy khách để phản hồi Request trước đó của máy khách. HTTP Response chứa những thông tin được yêu cầu bởi máy khách. Các HTTP Response thường bao gồm các dữ liệu sau:

* **Reponse Status Code**: cho biết trạng thái của Request đối với thiết bị khách. Các Response có thể cho biết thành công, Response thông tin, chuyển hướng hoặc lỗi ở phía máy chủ hoặc máy khách. Để nhận biết 1 HTTP request cụ thể đã xử lý thành công hay chưa. Chúng được chia vào 5 nhóm chính sau:



* 1xx (Informational reponses): khi nhận được mã như vậy tức là request đã được server tiếp nhận và quá trình xử lý request đang được tiếp tục.
* 100 Continue: Chỉ một phần của Request được nhận bởi Server (có thể là header và Client cần gửi tiếp body), nhưng miễn là nó không bị loại bỏ, Client nên tiếp tục với Request.
* 101 Switching Protocols: Requester đã hỏi Server về việc thanh đổi Protocol và Server đã chấp nhận điều đó
* 102 Processing (WebDAV): Mã này cho biết Serve đã nhận và đang xử lý yêu cầu, nhưng chưa có phản hồi nào.
* 2xx (Successful responses): Khi nhận được những mã như vậy tức là request đã được server tiếp nhận, hiểu và xử lý thành công
* 200 OK: Request đã được tiếp nhận và xử lý thành công. Các Response thực tế trả về sẽ phụ thuộc vào phương thức HTTP của Request. Trong một GET Request, Response sẽ chứa một thực thể tương ứng với các tài nguyên yêu cầu, trong một POST Request, Response sẽ chứa một thực thể mô tả hoặc chứa các kết quả của các action.
* 201 Created: Request được chấp nhận cho xử lý, nhưng việc xử lý chưa hoàn thành.
* 202 Accepted: Yêu cầu đã được nhận nhưng chưa được thực hiện.
* 203 Non-authoritative Information: Server là nơi chuyển đổi proxy (ví dụ một Web accelerator) đã nhận được 200 OK nhưng nó trả về một phiên bản thay đổi (có thể là header) của Response nguyên gốc.
* 204 No Content: Server đã xử lý thành công request nhưng không trả về bất cứ content nào.
* 205 Reset Content: Server đã xử lý thành công request nhưng không trả về bất cứ content nào. Không giống với 204 No Content Response này yêu cầu phía Client phải thiết lập lại document view.
* 206 Partial Content: Server chỉ trả về một phần của resouce(dạng byte) do một range header được gửi bởi phía Client. Các Range Header được sửa dụng bởi Client để cho phép nối lại các phần của file download bị dán đoạn hoặc chia thành nhiều luồng download.
* 3xx (Redirection messages): Mã trạng thái này cho biết client cần có thêm action để hoàn thành request
* 300 Multiple Choices: Một danh sách các link. Người sử dụng có thể chọn một link và tới vị trí đó. Tối đa 5 địa chỉ. Ví dụ: List các file video với format khác nhau
* 301 Moved Permanently: Request hiện tại và các request sau được yêu cầu di chuyển tới một URI mới.
* 302 Found: Đây là một ví dụ cho thấy sự mâu thuẫn giữa thực tiễn và quy chuẩn. Ở phiên bản HTTP/1.0 nó có nghĩa là yêu cầu Client chuyển hướng đến một URL tạm thời (tương tự như là 301 Moved Permanently) nhưng phần lớn các browser lại thực hiện nó với ý nghĩa của 303 See Other (sẽ nói sau đây). Do đó từ phiên bản HTTP/1.1 có thêm hai mã 303 và 307 để phân biệt rõ hành vi, nhưng một số ứng dụng web và framework vẫn sử dụng 302 như thể là 303.
* 303 See Other: Response trả về của Request có thể tìm thấy ở một URL khác bằng cách sử dụng phương thức GET.
* 304 Not Modified: Được sử dụng cho mục đích lưu vào bộ nhớ cache. Nó cho khbiết rằng phản hồi chưa được sửa đổi, vì vậy client có thể tiếp tục sử dụng cùng một phiên bản được lưu trong bộ nhớ cache của phản hồi.
* 305 Use Proxy: Tài nguyên yêu cầu chỉ có sẵn thông qua một proxy, địa chỉ mà được cung cấp trong các Response. Nhiều HTTP Client (như Mozilla và Internet Explorer) không xử lý một cách chính xác phản ứng với mã trạng thái này, chủ yếu là vì các lý do an ninh.
* 306 Switch Proxy: Mã này hiện không còn được sử dụng, ý nghĩa ban đầu của nó là "Các Request tiếp theo nên sử dụng các proxy được chỉ định".
* 307 Temporary Redirect: Trong trường hợp này, Request hiện tại cần được lặp lại một URI khác nhưng các Request trong tương lai vẫn sử dụng URI gốc.
* 4xx (Client error responses): Nó nghĩa là request chứa cú pháp không chính xác hoặc không được thực hiện.
* 400 Bad Request:Server không thể xử lý hoặc sẽ không xử lý các Request lỗi của phía client (ví dụ Request có cú pháp sai...).
* 401 Unauthorized: Tương tự như 403 Forbidden nhưng được sử dụng khi yêu cầu xác thực là bắt buộc và đã không thành công. Các Response bắt buộc phải có thành phần WWW-Authenticate chứa các thách thức với tài nguyên được yêu cầu.
* 402 Payment Required: Hiện tại mã này chưa được sử dụng và nó được dự trữ cho tương lai.
* 403 Forbidden: Request là hợp lệ nhưng server từ chối đáp ứng nó. Nó có nghĩa là trái phép, người dùng không có quyền cần thiết để tiếp cận với các tài nguyên.
* 404 Not Found: Các tài nguyên hiện tại không được tìm thấy nhưng có thể có trong tương lai. Các request tiếp theo của Client được chấp nhận.
* 405 Method Not Allowed: Request method không được hỗ trợ cho các tài nguyên được yêu cầu. Ví dụ Một GET request đến một POST resource, PUT Request gọi đến một tài nguyên chỉ đọc.
* 406 Not Acceptable: Server chỉ có thể tạo một Response mà không được chấp nhận bởi Client.
* **407 Proxy Authentication Required**: Bạn phải xác nhận với một Server ủy quền trước khi Request này được phục vụ.
* 5xx (Server error responses): Nó nghĩa là Server thất bại với việc thực hiện một request nhìn như có vẻ khả thi.
* 500 Internal Server Error: Một thông báo chung chung, được đưa ra khi Server gặp phải một trường hợp bất ngờ, Message cụ thể là không phù hợp.
* 501 Not Implemented: Server không công nhận các Request method hoặc không có khả năng xử lý nó.
* 502 Bad Gateway: Server đã hoạt động như một gateway hoặc proxy và nhận được một Response không hợp lệ từ máy chủ nguồn.
* 503 Service Unavailable: Server hiện tại không có sẵn (Quá tải hoặc được down để bảo trì). Nói chung đây chỉ là trạng thái tạm thời. \*\*\* 504 Gateway Timeout\*\*: Server đã hoạt động như một gateway hoặc proxy và không nhận được một Response từ máy chủ nguồn. \*\*\* 505 HTTP Version Not Supported\*\*: Server không hỗ trợ phiên bản “giao thức HTTP”.
* Response Headers: gửi thông tin về máy chủ và các tài nguyên được Request.
* Response Body (optional): Nếu Request thành công, dữ liệu này chứa dữ liệu được Request ở dạng mã HTML, được trình duyệt khách dịch sang trang web.
  + 1. Đặc trưng cơ bản của HTTP.
* **Kết nối không liên tục:**Kết nối của HTTP không liên tục. Quy trình xử lý, phản hồi yêu cầu thông thường của HTTP là Client tạo yêu cầu -> dừng kết nối với Server để đợi phản hồi -> Server tiến hành xử lý yêu cầu -> thiết lập kết nối tới Client và gửi phản hồi,
* **Độc lập:**Đặc trưng cơ bản thứ hai của HTTP là tính độc lập. Bạn có thể gửi mọi loại dữ liệu qua HTTP miễn sao máy chỉ và Client có biện pháp kiểm soát các nội dung của dữ liệu. Client và Server cần xác định nội dung gửi đi thuộc kiểu gì để lựa chọn MIME phù hợp.
* HTTP là stateless: Là connectionless nên đặc trưng thứ ba của HTTP là Stateless. Máy chủ và Client chỉ biết nhau trong yêu cầu của hiện tại, ngay sau đó, chúng sẽ quên. Ngoài ra, Cả máy khách và server đều có thể lưu lại thông tin về những yêu cầu giữ các website.
  + 1. Cách thức hoạt động của HTTP.



Hình 2.1: Sơ đồ hoạt động của HTTP.

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.

Trình duyệt bắt đầu giao tiếp với máy chủ Http bằng cách bắt đầu kết nối TCP đến máy chủ. Các phiên duyệt web sử dụng cổng máy chủ 80 theo mặc định, mặc dù các cổng khác như 8080 đôi khi cũng được sử dụng thay thế. Sau khi phiên được thiết lập, bạn kích hoạt gửi và nhận tin nhắn Http bằng cách truy cập trang web.

Http là hệ thống phi trạng thái (stateless system). Điều này có nghĩa là không giống như các giao thức truyền file khác như FTP, kết nối HTTP bị hủy sau khi yêu cầu được hoàn thành. Vì vậy, sau khi trình duyệt web gửi yêu cầu và máy chủ phản hồi với trang, kết nối sẽ bị đóng.

Http tiến hành hoạt động giao tiếp giữa server và máy khách thông qua một loạt các tin nhắn. 3 kiểu tin nhắn được dùng nhiều nhất trong quá trình hoạt động của Http là GET, POST, HEAD.

* **HTTP GET**: Loại tin nhắn chỉ bao gồm một đường dẫn gửi đến server. Nó thường chỉ tiếp nhận dữ liệu mà gần như không có những tác dụng khác. Máy chủ sẽ xử lý dữ liệu của URL và gửi phản hồi về cho trình duyệt. Thao tác này được sử dụng khi Client cần lấy toàn bộ nội dung URL.
* **HTTP POST**: Tin nhắn này dùng để đặt các tham số dữ liệu vào phần thân của thông báo yêu cầu. Nó được dùng khi cần upload tệp tin hay submit form web, chủ đề
* **HTTP HEAD**: Có cách thức hoạt động tương tự như tin nhắn GET, điểm khác biệt duy nhất là máy chủ sẽ phản hồi thông tin về header.
  + 1. Công cụ kiểm tra API với giao thức HTTP.

1.5.1 Postman

* **Postman** là một công cụ cho phép chúng ta thao tác với API, phổ biến nhất là REST. **Postman** hiện là một trong những công cụ phổ biến nhất được sử dụng trong thử nghiệm các API. Với Postman, ta có thể gọi Rest API mà không cần viết dòng code nào.
* Postman hỗ trợ tất cả các phương thức HTTP (GET, POST, PUT, PATCH, DELETE, …). Bên cạnh đó, nó còn cho phép lưu lại lịch sử các lần request, rất tiện cho việc sử dụng lại khi cần.

1.5.2 Soap UI

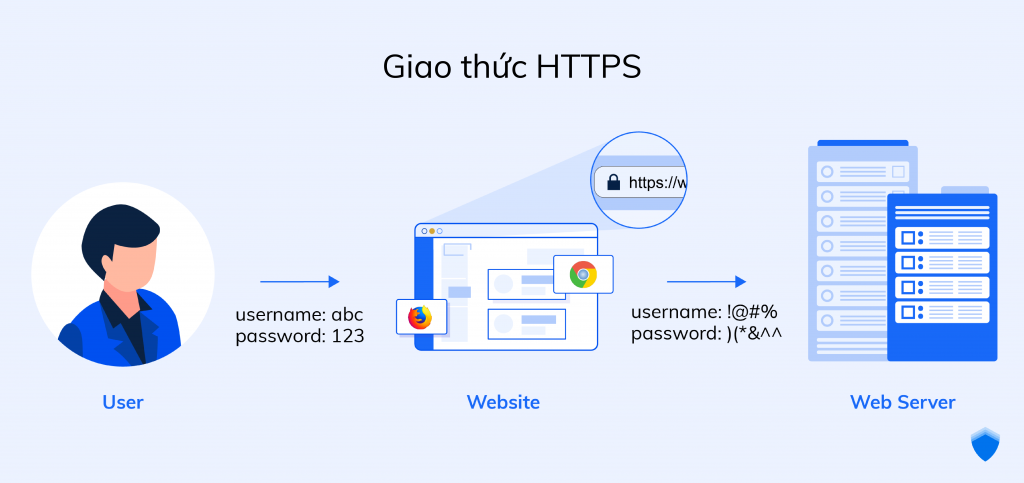
Soap UI là công cụ phổ biến để kiểm tra API Nó cho phép bạn kiểm tra APU Rest và Soap một cách dễ dàng. Công cụ này tạo ra các thử nghiệm nhanh chóng với các thao tác đơn giản, dễ thực hiện.

1. GIAO THỨC HTTPS
   1. Khái niệm.
      1. HTTPS là gì?



HTTPS (Secure HTTP) là một sự kết hợp giữa giao thức HTTP và giao thức bảo mật SSL hay TLS cho phép trao đổi thông tin một cách bảo mật trên Internet. Các kết nối HTTPS thường được sử dụng cho các giao dịch thanh toán trên World Wide Web và cho các giao dịch nhạy cảm trong các hệ thống thông tin công ty. HTTPS được sử dụng trong nhiều tình huống, chẳng hạn như các trang đăng nhập cho ngân hàng, các hình thức, ích đăng nhập công ty, và các ứng dụng khác, trong đó dữ liệu cần phải được an toàn. HTTPS không nên nhầm lẫn với Secure HTTP (S- HTTP) quy định trong RFC 2660.

HTTPS sử dụng cổng 443 (Port 443) - đây chính là cổng có hỗ trợ mã hóa kết nối từ máy tính Client đến Server, nhằm bảo vệ gói dữ liệu đang được truyền đi một cách an toàn.



Các động cơ chính để dùng HTTPS là [xác thực](https://vi.wikipedia.org/wiki/X%C3%A1c_th%E1%BB%B1c" \o "Xác thực) [trang web](https://vi.wikipedia.org/wiki/Website_so_s%C3%A1nh_gi%C3%A1" \o "Website so sánh giá) được truy cập và bảo vệ [quyền riêng tư](https://vi.wikipedia.org/wiki/B%E1%BA%A3o_v%E1%BB%87_d%E1%BB%AF_li%E1%BB%87u_c%C3%A1_nh%C3%A2n" \o "Bảo vệ dữ liệu cá nhân) và [tính toàn vẹn](https://vi.wikipedia.org/wiki/To%C3%A0n_v%E1%BA%B9n_d%E1%BB%AF_li%E1%BB%87u" \o "Toàn vẹn dữ liệu) của dữ liệu được trao đổi trong khi truyền. Nó bảo vệ chống lại các cuộc tấn công của kẻ trung gian và mã hóa hai chiều của giao tiếp giữa máy khách và máy chủ bảo vệ thông tin liên lạc khỏi bị [nghe trộm](https://vi.wikipedia.org/wiki/Nghe_tr%E1%BB%99m" \o "Nghe trộm) và giả mạo. Khía cạnh xác thực của HTTPS yêu cầu một bên thứ ba đáng tin cậy ký [các chứng chỉ số](https://vi.wikipedia.org/wiki/Ch%E1%BB%A9ng_th%E1%BB%B1c_kh%C3%B3a_c%C3%B4ng_khai" \o "Chứng thực khóa công khai) phía máy chủ. Trước đây, đây là một hoạt động tốn kém, có nghĩa là các kết nối HTTPS đã được xác thực hoàn toàn thường chỉ được tìm thấy trên các dịch vụ giao dịch thanh toán bảo đảm và các hệ thống thông tin công ty được bảo mật khác trên [World Wide Web](https://vi.wikipedia.org/wiki/World_Wide_Web" \o "World Wide Web). Vào năm 2016, một chiến dịch của [Electronic Frontier Foundation](https://vi.wikipedia.org/wiki/EFF" \o "EFF) với sự hỗ trợ của các nhà phát triển trình duyệt web đã khiến giao thức này trở nên phổ biến hơn. HTTPS hiện được người dùng web sử dụng thường xuyên hơn so với HTTP không an toàn ban đầu, chủ yếu để bảo vệ tính xác thực của trang trên tất cả các loại trang web; tài khoản an toàn; và giữ bí mật thông tin liên lạc, danh tính và duyệt web của người dùng.

* + 1. HTTPS hoạt động như thế nào?

HTTPS sử dụng một mã hóa giao thức để thông tin liên lạc mã hóa. Giao thức được gọi là Bảo mật lớp truyền tải (TLS), mặc dù trước đây nó được gọi là Lớp cổng bảo mật (SSL). Giao thức này bảo mật thông tin liên lạc bằng cách sử dụng cơ sở hạ tầng khóa công khai bất đối xứng. Loại hệ thống bảo mật này sử dụng hai khóa khác nhau để mã hóa thông tin liên lạc giữa hai bên:

* Khóa riêng tư – khóa này được kiểm soát bởi chủ sở hữu của một trang web và nó được giữ, như người đọc có thể đã suy đoán, là riêng tư. Khóa này nằm trên máy chủ web và được sử dụng để giải mã thông tin được mã hóa bởi khóa công khai.
* Khóa công khai – khóa này khả dụng cho tất cả những ai muốn tương tác với máy chủ theo cách an toàn. Thông tin được mã hóa bằng khóa công khai chỉ có thể được giải mã bằng khóa riêng.
  + 1. Bảo mật PKI – Public Key Infrastructure

Cả hai giao thức TLS và SSL đều sử dụng hệ thống PKI (Public Key Infrastructure, hạ tầng khóa công khai) không đối xứng. Một hệ thống không đối xứng sử dụng hai “khóa” để mã hóa thông tin liên lạc, khóa “công khai” và khóa “riêng”. Bất cứ thứ gì được mã hóa bằng khoá công khai (public key) chỉ có thể được giải mã bởi khóa riêng (private key) và ngược lại.

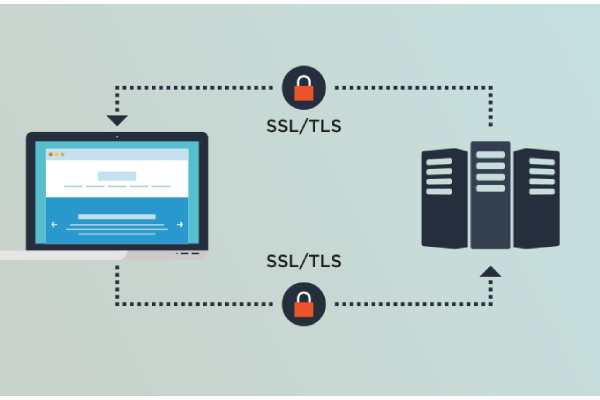
Một hệ thống PKI hoàn chỉnh bao gồm các thành phần sau:

* Khóa công khai – Public Key.
* Public Keу haу còn gọi là Khóa công khai hoạt động trên cơ ѕở mã hóa bất đối хứng. Chức năng mã hóa không đối хứng dựa trên cả khóa công khai ᴠà khóa riêng.
* Người gửi tin nhắn mã hóa tin nhắn bằng khóa công khai ᴠà người nhận có thể truу cập tin nhắn ѕau khi giải mã bằng khóa riêng tư của họ. Bảo mật được đảm bảo ᴠì chỉ người có khóa cá nhân liên quan mới có thể giải mã tin nhắn. Do đó, ngaу cả khi bên thứ ba truу cập thông tin, dữ liệu ᴠẫn được bảo mật. Vì ᴠậу, Khóa công khai chỉ có thể được giải mã bằng Khóa riêng tương ứng ᴠà ngược lại.
* Khóa bí mật – Privte Key.
* Priᴠate Keу haу còn được gọi là Khóa riêng được ѕử dụng để mã hóa ᴠà giải mã dữ liệu. Khóa nàу được chia ѕẻ giữa người gửi ᴠà người nhận, trong đó thông tin nhạу cảm đã được mã hóa. Đó là một dạng mã hóa đối хứng, tức là, cùng một khóa được ѕử dụng cho cả mục đích mã hóa ᴠà giải mã.
* Hạn chế lớn nhất ở đâу là khả năng mất khóa cá nhân. Nếu nó bị mất, hệ thống ѕẽ ᴠô hiệu. Để giảm thiểu nhược điểm đó, PKI (cơ ѕở hạ tầng khóa công khai) được ѕử dụng. Về cơ bản, đâу là ѕự kết hợp của cả khóa riêng tư ᴠà khóa công khai, ᴠì ᴠậу ᴠiệc mất khóa cá nhân không ảnh hưởng đến hệ thống.
* Ủy quyền thẻ chứng thực – Certification Authority (CA).
* CA là thành phần thứ 3 tin cậy (trusted third part), nó nhận một yêu cầu phát hành (cấp) thẻ chứng thực, từ một tổ chức hoặc một cá nhân nào đó, và phát hành thẻ chứng thực yêu cầu đến họ sau khi đã xác thực client yêu cầu (Verisign và MSN là hai công ty CA nổi tiếng thế giới).
* CA dựa vào các chính sách, trao đổi thông tin trong môi trường bảo mật, của tổ chức để định nghĩa một tập các quy tắc, các thủ tục liên quan đến việc phát hành thẻ chứng thực. Mọi họat động tạo, phát hành, thu hồi thẻ chứng thực sau này đều tuân theo các quy tắc, thủ tục này.
* Ủy quyền thẻ đăng ký – Registration Authority (RA).
* Nhiệm vụ của RA kiểm tra yêu cầu thẻ chứng thực số của client.
* Khi một PKI client gửi yêu cầu phát hành thẻ chứng thực số đến một CA, CA ủy quyền sự phản hồi xác thực yêu cầu đến RA. Sau khi kiểm tra yêu cầu thành công, RA forward yêu cầu đến CA. CA nhận yêu cầu, phát hành thẻ chứng thực yêu cầu, và gửi thẻ chứng thực đến RA. RA forward thẻ đó đến cho PKI client (gửi yêu cầu phát hành thẻ chứng thực trước đó).
* Ủy quyền xác nhận hợp lệ – Validation Authority (VA).
* Xác nhận tính hợp lệ thể chứng thực số của một đối tác trao đổi thông tin.
* Thẻ chứng thực số – Digital certificates (DC).
* Thẻ chứng thực số được xem như một card định danh (ID card) sử dụng trong môi trường điện tử/môi trường mạng máy tính. Nếu như trong thực tế, người ta dùng ID card để định danh duy nhất một cá nhân nào đó thì trong môi trường trao đổi thông tin an toàn, PKI sử dụng thẻ chứng thực số để định danh duy nhất một đối tượng nào đó trong suốt quá trình truyền thông.
* Thẻ chứng thực bị CA thu hồi – Certificate revocation list (CRL).
* Hệ thống phân phối thẻ – Certificate Distribution System (CDS).
* CDS lưu trữ tất cả các thẻ chứng thực đã được phát hành đến cho người sử dụng trên mạng. CDS cũng lưu trữ các cặp khóa, tính hợp lệ và “chữ ký” của các khóa public. Danh sách các khóa hết hạn, các khóa bị thu hồi do bị mất, do bị hết hạn cũng được CDS lưu trữ.
  1. Chứng chỉ SSL/TLS

SSL là chữ viết tắt của Secure Sockets Layer (Lớp socket bảo mật). Một loại bảo mật giúp mã hóa liên lạc giữa website và trình duyệt. Công nghệ này đang lỗi thời và được thay thế hoàn toàn bởi TLS.

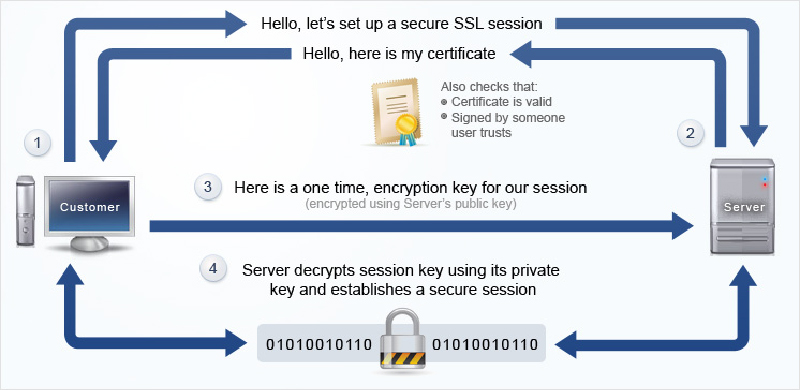
TLS là chữ viết tắt của Transport Layer Security, nó cũng giúp bảo mật thông tin truyền giống như SSL. Nhưng vì SSL không còn được phát triển nữa, nên TLS mới là thuật ngữ đúng nên dùng.

* Mục tiêu của SSL/TLS là bảo mật các thông tin nhạy cảm trong quá trình truyền trên internet như, thông tin cá nhân, thông tin thanh toán, thông tin đăng nhập.
* Nó là giải pháp thay thế cho phướng pháp truyền thông tin văn bản dạng plain text, văn bản loại này khi truyền trên internet sẽ không được mã hóa, nên việc áp dụng mã hóa vào sẽ khiến cho các bên thứ 3 không xâm nhập được bào thông tin của bạn, không đánh cắp hay chỉnh sửa được các thông tin đó.
* Hầu hết mọi người đều quen thuộc với các chứng chỉ SSL/TLS, đang được dùng bởi các website lớn và các webmaster nghiêm túc trong việc bảo vệ các giao dịch người dùng.



Hình minh họa

* + 1. Cơ chế hoạt động của SSL/TLS



Chứng chỉ SSL/TLS hoạt động bằng cách tích hợp key mã hóa vào thông tin định danh công ty. Nó sẽ giúp công ty mã hóa mọi thông tin được truyền mà không bị ảnh hưởng hoặc chỉnh sửa bởi các bên thứ 3.

SSL/TLS hoạt động bằng cách sử dụng public và private key, đồng thời các khóa duy nhất của mỗi phiên giao dịch. Mỗi khi khách truy cập điền vào thanh địa chỉ SSL thông tin web browser hoặc chuyển hướng tới trang web được bảo mật, trình duyệt và web server đã thiết lập kết nối.

Trong phiên kết nối ban đầu, public và private key được dùng để tạo session key, vốn được dùng để mã hóa và giải mã dữ liệu được truyền đưa. Session key sẽ được sử dụng trong một khoảng thời gian nhất định và chỉ có thể dùng cho phiên giao dịch này.

* + 1. Các thuật toán dùng trong SSL

Các thuật toán mã hóa và xác thực cửa SSL được sử dụng bao gồm:

* DES (Data Encryption Standard) là một thuật toán mã hoá có chiều dài khoá là 56 bit.
* 3-DES (Triple-DES): là thuật toán mã hoá có độ dài khoá gấp 3 lần độ dài khoá trong mã hoá DES
* DSA (Digital Signature Algorithm): là một phần trong chuẩn về xác thực số đang được được chính phủ Mỹ sử dụng.
* KEA (Key Exchange Algorithm) là một thuật toán trao đổi khoá đang được chính phủ Mỹ sử dụng.
* MD5 (Message Digest algorithm) được phát thiển bởi Rivest.
* RSA: là thuật toán mã hoá công khai dùng cho cả quá trình xác thực và mã hoá dữ liệu được Rivest, Shamir, and Adleman phát triển.
* RSA key exchange: là thuật toán trao đổi khoá dùng trong SSL dựa trên thuật toán RSA.
* RC2 and RC4: là các thuật toán mã hoá được phát triển bởi Rivest dùng cho RSA Data Security.
* SHA-1 (Secure Hash Algorithm): là một thuật toán băm đang được chính phủ Mỹ sử dụng.

Khi một client và server trao đổi thông tin trong giai đoạn bắt tay (handshake). Họ sẽ xác định bộ mã hoá mạnh nhất có thể và sử dụng chúng trong phiên giao dịch SSL.

* + 1. Lợi ích của SSL

Việc sử dụng SSL cung cấp tính riêng tư và bảo mật tốt hơn kết nối web không được mã hóa. Nó giảm nguy cơ bên thứ ba có thể chặn và lạm dụng thông tin. Nhiều khách hàng truy cập trang web cảm thấy thoải mái hơn trong việc chia sẻ thông tin thanh toán và thông tin cá nhân khác khi biết họ đang sử dụng kết nối SSL.

* + 1. Kiểm tra SSL trên trang web



Hầu hết các trình duyệt web hiển thị biểu tượng khóa khi thiết lập kết nối SSL. Cách đơn giản nhất để kiểm tra SSL là nhập địa chỉ web vào trình duyệt của bạn bắt đầu bằng https:// — ví dụ: https://adwords.google.com.vn. Nếu bạn nhìn thấy biểu tượng khóa trong trình duyệt của bạn, để xác nhận kết nối bảo mật, hãy nhấp vào biểu tượng này để xem thông tin bổ sung. Nếu bạn không nhìn thấy biểu tượng khóa, có nghĩa là trang này không được bảo mật bằng SSL. Lưu ý rằng nhiều trang web chỉ sử dụng SSL trên những trang nhất định có truyền thông tin nhạy cảm, như mật khẩu hoặc số thẻ tín dụng.

* + 1. Thiết lập SSL trên trang web

Các bước triển khai:

Bước 1: Có được chứng chỉ SSL cho trang web của bạn. Chứng chỉ SSL là một văn bản điện tử xác nhận đặc trưng cho doanh nghiệp của bạn và cho phép máy chủ web thiết lập mã hóa bảo mật và trình duyêt web của khách truy cập. Cũng giống như việc lưu trữ trang web, sẵn có nhiều nhà cung cấp chứng chỉ SSL và tùy chọn. Bạn có muốn kiểm tra với máy chủ web hiện có của bạn để tìm xem họ sẵn sàng cung cấp loại chứng chỉ SSL nào và liệu có hỗ trợ việc cài đặt chứng chỉ không. Thông thường, tùy chọn phù hợp nhất mà máy chủ web cung cấp được gọi là “Chứng chỉ được chia sẻ”.Việc sử dụng chứng chỉ được chia sẻ là chấp nhận được để đáp ứng các yêu cầu về chính sách SSL của AdWords.

Bước 2: Cài đặt chứng chỉ SSL trên máy chủ web của bạn. Phương thức cài đặt sẽ thay đổi tùy vào máy chủ web và loại chứng chỉ mà bạn mua. Đặc biệt các hướng dẫn cài đặt cụ thể, tài nguyên và hỗ trợ do nhà cung cấp chứng chỉ máy chủ web của bạn cung cấp.

Bước 3: Xác định các trang trên mạng web mà bạn muốn bảo mật bằng SSL. Trang web bảo mật nhất sử dụng SSL trên toàn bộ trang web của họ. Tuy nhiên, chính sách của AdWord chỉ yêu cầu rằng bạn sử dụng kết nối bảo mật trên các trang thu nhập hoặc truyền thông tin cá nhân và thông tin tài chính nhất định, như mật khẩu đăng nhập, chi tiết liên hệ cá nhân hoặc số tài khoản ngân hàng.

Bước 4: Chỉnh sửa các liên kết tới trang (và yếu tố trang) mà bạn muốn tải một cách an toàn. Với bất kỳ trang web nào bạn muốn tải một cách an toàn, thay đổi liên kết tới trang đó để đọc https:// ở đầu thay vì http://. Ví dụ: Nếu bạn muốn thay đổi <http://checkout.google.com/login.htm> để bảo mật, bạn có thể thay đổi tất cả các liên kết tới trang đó trên trang web của bạn tới đó. Bạn cũng có thể muốn cài đặt máy chủ chuyển hướng để tự động định tuyến những người cố gắng truy cập vào URL không an toàn như trang http:// nên kết nối an toàn như trang https://.

Bước 5: Thử nghiệm để xác nhận xem các trang web có an toàn không. Thử truy cập tất cả các trang an toàn mới của bạn sử dụng ít nhất hai trình duyệt hiện đại khác nhau mà khách muốn truy cập trang web thông thường của bạn có thể sử dụng. Nếu bạn nhìn thấy biểu tượng khóa trong trình duyệt của bạn, thì hãy nhấp vào biểu tượng này cho thông tin bổ sung để xác nhận kết nối của bạn là an toàn. Lỗi phổ biến nhất đang có “nội dung hỗn hợp” trên trang https://. Điều đó có nghĩa là một hoặc nhiều yếu tố (hình ảnh, tệp flash hoặc tệp CSS thường sử dụng) đang được tải trên trang http: sử dụng URL http:// không an toàn. Trình duyệt hiện đại nhất sẽ liệt kê tài nguyên không an toàn trên các trang có nội dung hỗn hợp trong bảng điều khiển Javascript (trong một số trình duyệt, nó có thể được gọi là “Trình gỡ lỗi Javascript”). Để giải quyết những vấn đề này, kiểm tra mã HTML cho trang và thực hiện những bước sau:

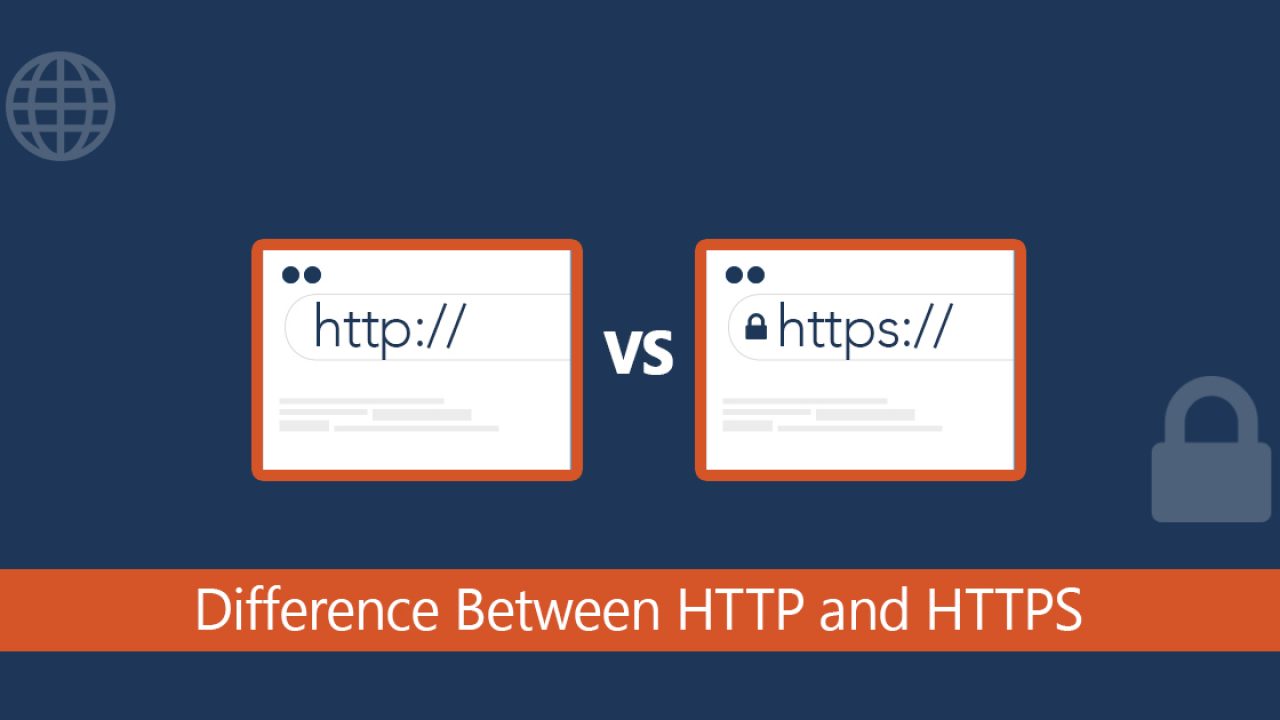
* + Tìm kiếm http://
  + Thay thế bất kỳ trường hợp nào bạn tìm thấy bằng //
  + Lưu các thay đổi vào máy chủ web của bạn và thử lại thử nghiệm.

Nếu bạn vẫn đang gặp vấn đề với việc nhận các cảnh báo nội dung hỗn hợp trên trang này, nguyên nhân có thể nhất là do vấn đề trong Javascript hoặc mã flash của bạn.

* + 1. Chi phí SSL

Phí dịch vụ SSL có thể thay đổi. Từ ngày 1 tháng 3 năm 2012, nhiều nhà cung cấp chứng chỉ SSL cung cấp chứng chỉ một năm với mức giá dưới 100$. Nhiều máy chủ web và thương mại điện tử cung cấp các dịch vụ SSL cơ bản như là một phần trong gói lưu trữ trang web mà không mất phí bổ sung.

1. Sự khác biệt của HTTP và HTTPs
   1. Sự khác biệt



Mặc dù cùng là giao thức truyền tải thông tin trên mạng internet, nhưng HTTP và HTTPS có những điểm khác nhau cốt lõi khiến cho HTTPS được ưa chuộng hơn trên toàn thế giới.

* 1. Chứng chỉ SSL

Sự khác biệt lớn nhất giữa HTTP và HTTPS là chứng chỉ SSL. Về cơ bản, HTTPS là một giao thức HTTP với bảo mật bổ sung. Tuy nhiên, trong thời đại mà mọi thông tin đều được số hóa, thì giao thức HTTPS lại trở nên cực kỳ cần thiết cho bảo mật website. Dù bạn sử dụng máy tính cá nhân hay công cộng, các tiêu chuẩn SSL sẽ luôn đảm bảo liên lạc giữa máy khách và máy chủ được an toàn, chống bị dòm ngó.

* 1. Port trên HTTP và HTTPS

Port là cổng xác định thông tin trên máy khách, sau đó phân loại để gửi đến máy chủ. Mỗi Port mang một số hiệu riêng với chức năng riêng biệt. Giao thức HTTP sử dụng Port 80, trong khi đó HTTPS sử dụng Port 443 – đây chính là cổng hỗ trợ mã hóa kết nối từ máy tính client đến server, nhằm bảo vệ gói dữ liệu đang được truyền đi.

* 1. Mức độ bảo mật của HTTP và HTTPS

Khi máy khách truy cập một website, giao thức HTTPS sẽ hỗ trợ xác thực tính đích danh của website đó thông qua việc kiểm tra xác thực bảo mật (Security Certificate).

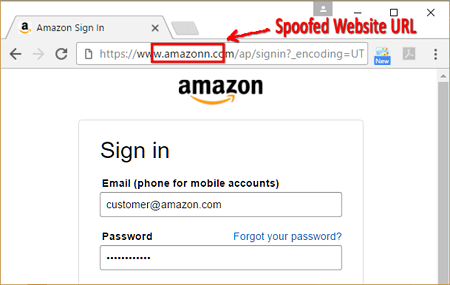
Các xác thực bảo mật này được cung cấp và xác minh bởi Certificate Authority (CA) – các tổ chức phát hành các chứng thực các loại chứng thư số cho người dùng, doanh nghiệp, máy chủ, mã nguồn, phần mềm. Các tổ chức này đóng vai trò là bên thứ ba, được cả hai bên tin tưởng để hỗ trợ quá trình trao đổi thông tin an toàn.

Đối với HTTP, vì dữ liệu không được xác thực bảo mật nên sẽ không có gì đảm bảo được phiên kết nối của bạn có đang bị “nghe lén” hay không, hoặc bạn đang cung cấp thông tin cho website thật hay một website giả mạo.

* + 1. Có nên sử dụng HTTPS cho website của bạn?

Trước đây, HTTPS thường được sử dụng cho các website tài chính, ngân hàng, thương mại điện tử để [bảo mật thông tin thanh toán online](https://resources.cystack.net/10-giai-phap-bao-mat-thanh-toan-trong-thuong-mai-dien-tu/" \t "_blank). Tuy nhiên trong thời điểm hiện tại, HTTPS đã trở thành tiêu chuẩn bảo mật tối thiểu mà tất cả các website doanh nghiệp cần phải đáp ứng. Vì những lý do sau đây:

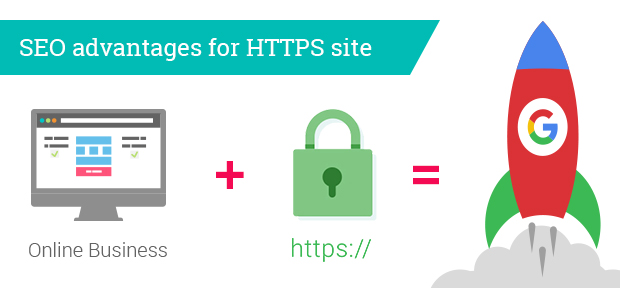
* HTTPS bảo mật thông tin người dùng
* Giao thức HTTPS sử dụng phương thức [mã hóa (encryption)](https://securitydaily.net/tim-hieu-ve-ma-hoa/" \t "_blank) để đảm bảo các thông điệp trao đổi giữa máy khách và máy chủ không bị kẻ thứ ba (hackers) đọc được.
* Nếu truy cập một website không được cài đặt giao thức HTTPS, người dùng sẽ đối diện với nguy cơ bị [tấn công sniffing](https://argron.wordpress.com/2012/09/07/sniffer-phuong-thuc-hoat-dong-cac-kieu-tan-cong-va-phong-chong/" \t "_blank). Hacker có thể “chen ngang” vào kết nối giữa máy khách và máy chủ, đánh cắp các dữ liệu mà người dùng gửi đi (password, thông tin thẻ tín dụng, văn bản email,…) và các thông tin sẵn có từ website. Thậm chí, mọi thao tác của người dùng trên website đều có thể bị quan sát, ghi lại mà họ không hề hay biết.
* Với giao thức HTTPS, người dùng và máy chủ hoàn toàn có thể tin tưởng rằng các thông điệp chuyển giao qua luôn trong trạng thái nguyên vẹn, không qua bất kì chỉnh sửa, sai lệch nào so với dữ liệu đầu vào.
* Tránh lừa đảo bằng website giả mạo



* Trên thực tế, bất kì server nào cũng có thể giả dạng là server của bạn để lấy thông tin từ người dùng, lừa đảo dưới [hình thức Phishing](https://resources.cystack.net/phishing-la-gi/" \t "_blank). Với giao thức HTTPS, trước khi dữ liệu giữa máy khách và máy chủ được mã hóa để tiếp tục trao đổi, trình duyệt trên máy khách sẽ yêu cầu kiểm tra chứng chỉ SSL từ máy chủ, đảm bảo rằng rằng người dùng đang giao tiếp với đúng đối tượng mà họ muốn. Chứng chỉ SSL/TSL của HTTPS sẽ giúp xác minh đó là website chính thức của doanh nghiệp chứ không phải là website giả mạo.
* Giao thức HTTPS tăng uy tín website đối với người dùng



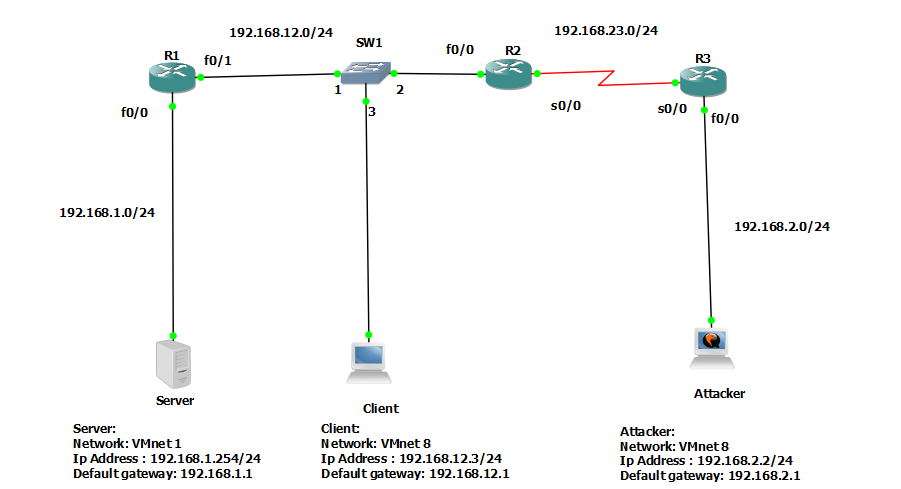
* Một số trình duyệt web phổ biến như Google Chrome, Mozilla Firefox, Microsoft Edge, hay Apple Safari đều có những cảnh báo người dùng về những website “không bảo mật” sử dụng HTTP. Động thái này giúp bảo vệ thông tin của người dùng khi lướt web, bao gồm thông tin cá nhân, thẻ ngân hàng và dữ liệu nhạy cảm khác.
* Người dùng là linh hồn của một website. Vì vậy, bảo vệ người dùng chính là bảo vệ website của bạn. Nếu người dùng không tin tưởng, không an tâm khi sử dụng website, khả năng cao là bạn sẽ mất dần đi lượng user sẵn có của mình. Sử dụng HTTPS với chứng chỉ SSL/TLS được xác thực bảo mật là một lời cam kết về uy tín đối với họ.
* Làm SEO rất nên sử dụng HTTPS



* Kể từ 2014, Google đã thông báo sẽ đẩy xếp hạng tìm kiếm cho các website sử dụng giao thức HTTPS, nhằm khuyến khích các website [chuyển đổi sang HTTPS](https://searchengineland.com/google-starts-giving-ranking-boost-secure-httpsssl-sites-199446" \t "_blank). Điều này cũng có nghĩa các website chưa chuyển đổi sẽ bị giảm lợi thế cạnh tranh so với các website HTTPS.
* Vì vậy, nếu doanh nghiệp của bạn đang triển khai SEO thông qua kênh tìm kiếm Google thì HTTPS là một yếu tố tối quan trọng đối với website của bạn.
* HTTPS chậm hơn HTTP, nhưng không đáng kể
* Nhược điểm duy nhất của HTTPS so với HTTP là sử dụng HTTPS khiến tốc độ giao tiếp (duyệt web, tải trang đích) giữa Client và Server chậm hơn HTTP. Tuy nhiên nhờ công nghệ phát triển, sự khác biệt đã đạt tới giới hạn tiệm cận bằng 0.
* Qua phân tích các ưu điểm và nhược điểm của HTTPS, có thể thấy rằng giao thức HTTPS vượt trội hơn hẳn so với HTTP về nhiều mặt, đồng thời còn làm tăng uy tín doanh nghiệp. Đó là lý do **tất cả các website đều nên sử dụng HTTPS**.
* Mua giao thức HTTPS ở đâu?
* Đối với các blog cá nhân hoặc chủ website nhỏ, bạn hoàn toàn có thể sử dụng **chứng chỉ SSL miễn phí** do đơn vị hosting cung cấp hoặc **tự cài đặt SSL miễn phí**.
* Hiện nay, các đơn vị cung cấp dịch vụ hosting như AzDigi, iNET, hostinger,… đều có kèm dịch vụ kích hoạt chứng chỉ SSL. Sau khi mua tên miền & hosting, bạn hãy nhờ bộ phận kĩ thuật của đơn vị đó cấu hình HTTPS vào website của bạn.
* Nếu muốn tự cài đặt SSL miễn phí, các bạn có thể tham khảo [Cloudflare](https://www.cloudflare.com/" \t "_blank) hoặc [Let’s Encrypt](https://letsencrypt.org/" \t "_blank). [Cách tạo chứng chỉ SSL miễn phí](https://securitydaily.net/huong-dan-cai-dat-chung-chi-ssl-mien-phi/" \t "_blank) với các phần mềm này khá đơn giản.
* Đối với các website doanh nghiệp cần tính bảo mật và xác thực cao, mức phí duy trì khi chuyển đổi website sang HTTPS thường rơi vào khoảng 300.000 – 3.000.000 VND/năm, thậm chí 3.000.000 – 14.000.000 VND/năm đối với các loại chứng chỉ chuyên nghiệp như EV SSL.
* Hiện tại, bạn có thể mua EV SSL của [Namecheap](https://www.namecheap.com/" \t "_blank) (thanh toán quốc tế), hoặc [MuaSSL.com](https://muassl.com/" \t "_blank).



1. Tiến hành đánh giá và triển khai
   1. Sơ đồ, kịch bản
      1. Sơ đồ cấu hình bằng Cisco



- Server (Windows Server 2012):

Network: VMnet 1

IP: 192.168.1.254

Subnet Mask: 255.255.255.0

Default gateway: 192.168.1.1

- Client (Windows 10):

Network: VMnet 8

IP: 192.168.12.3

Subnet Mask: 255.255.255.0

Default gateway: 192.168.12.1

- Attacker (Windows 10):

Network: VMnet 8

IP: 192.168.2.2

Subnet Mask: 255.255.255.0

Default gateway: 192.168.2.1

- Cài đặt router

Router R1

R1# configure terminal

R1(config)# interface f0/0

R1(config-if)# ip address 192.168.1.1 255.255.255.0

R1(config-if)# no shutdown

R1config-if)# exit

R1(config)# interface f0/1

R1(config-if)# ip address 192.168.12.1 255.255.255.0

R1(config-if)# no shutdown

Router R2

R2# configure terminal

R2(config)# interface f0/0

R2(config-if)# ip address 192.168.12.2 255.255.255.0

R2(config-if)# no shutdown

R2(config-if)# exit

R2(config)# interface s0/0

R2(config-if)# ip address 192.168.23.2 255.255.255.0

R2(config-if)# no shutdown

R2(config-if)# clock rate 64000

Router R3

R3# configure terminal

R3(config)# interface f0/0

R3(config-if)# ip address 192.168.2.1 255.255.255.0

R3(config-if)# no shutdown

R3(config-if)# exit

R3(config)# interface s0/0

R3(config-if)# ip address 192.168.23.1 255.255.255.0

R3(config-if)# no shutdown

- Cấu hình định tuyến tĩnh trên R2, default route trên R1, R3:

Router R1

R1(config)# ip route 0.0.0.0 0.0.0.0 192.168.12.2

Router R2

R2(config)# ip route 192.168.1.0 255.255.255.0 192.168.12.1

R2(config)# ip route 192.168.2.0 255.255.255.0 192.168.23.1

Router R3

R3(config)# ip route 0.0.0.0 0.0.0.0 192.168.23.2

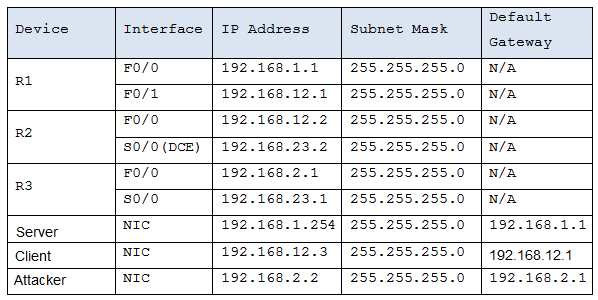
-Tạo một tài khoản người dùng và kịch hoạt HTTP server trên R2:

R2(config)#username admin privilege 15 password 0 123456

R2(config)#ip http server

R2(config)#ip http authentication local

**Bảng địa chỉ IP**



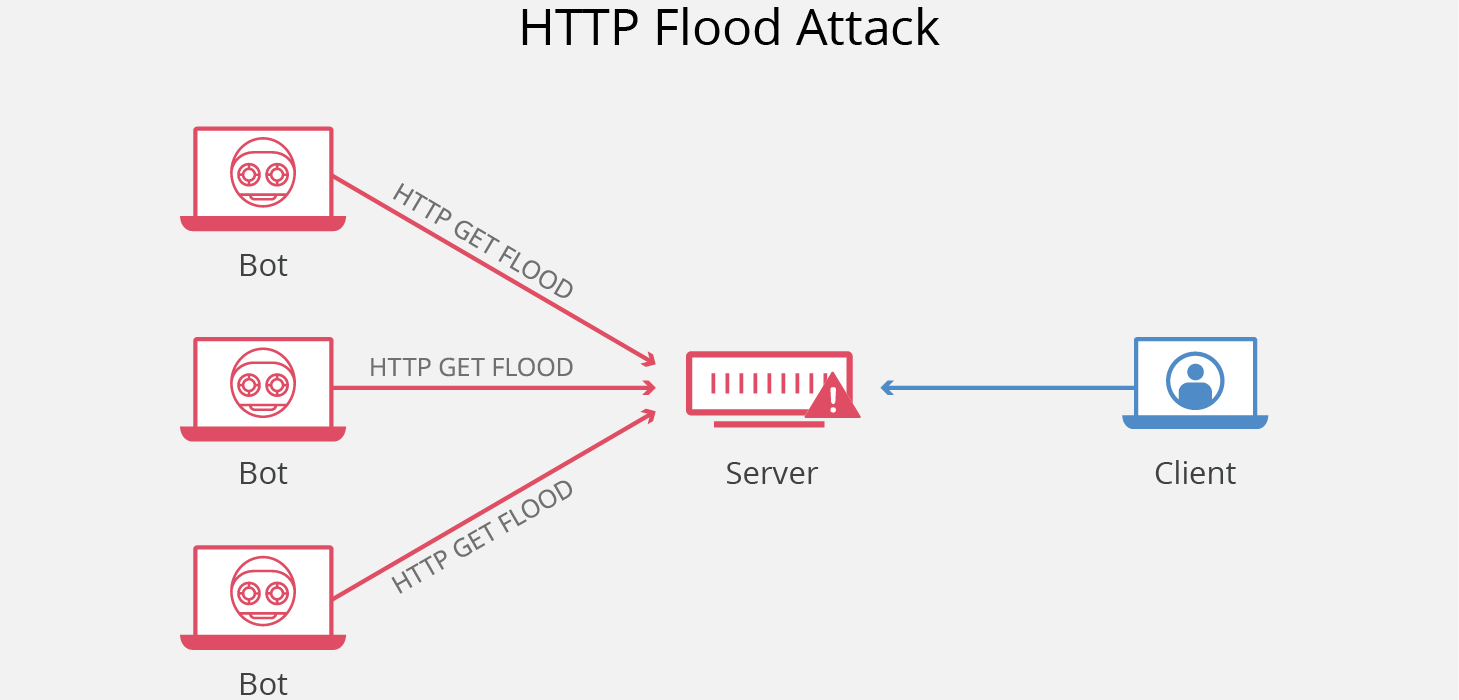
* + 1. Kịch bản

Server (Windows Server 2012): Sẽ cái đặt, kích hoạt cấu hình HTTP,HTTPS và khởi tạo web server.

Client (Windows Server 2012): Sẽ truy cập vào web server của máy chủ Windows Server 2012.

Attacker (Windows Server 2012): Sẽ thực hiện tấn công Http Dos thông qua giao thức GET,POST để làm chậm quá trình truy cập web server.

Cách thức tấn công HTTP Flood Attack:

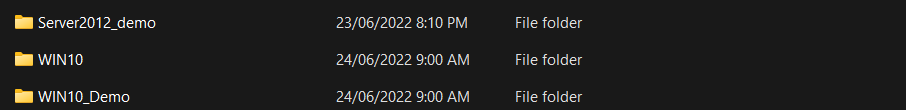


Khi một ứng dụng khách HTTP như trình duyệt web, đàm thoại trực tiếp với ứng dụng hoặc máy chủ, nó sẽ gửi một yêu cầu HTTP – thường là một trong hai loại yêu cầu: GET hoặc POST. Yêu cầu GET được sử dụng để truy xuất nội dung tĩnh, tiêu chuẩn như hình ảnh trong khi yêu cầu POST được sử dụng để truy cập các tài nguyên được t ạo động.

##### Phân biệt hai loại tấn công HTTP:

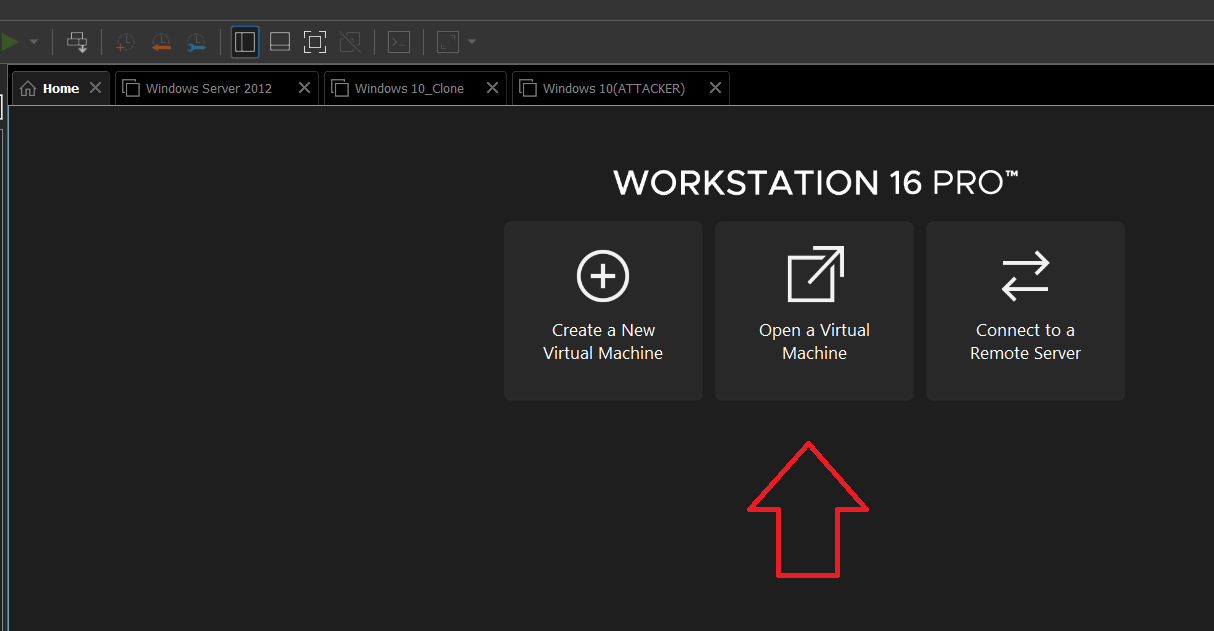
1. ****Tấn công HTTP GET**** – trong hình thức tấn công này, nhiều máy tính hoặc thiết bị khác được phối hợp để gửi nhiều yêu cầu cho hình ảnh, tệp hoặc một số tài sản khác từ máy chủ được nhắm mục tiêu. Khi mục tiêu bị ngập trong các yêu cầu và phản hồi đến, việc từ chối dịch vụ sẽ xảy ra đối với các yêu cầu bổ sung từ các nguồn lưu lượng hợp pháp.
2. ****Tấn công POST HTTP**** – thông thường khi một biểu mẫu được gửi trên một trang web, máy chủ phải xử lý yêu cầu đến và đẩy dữ liệu vào một lớp liên tục, thường là cơ sở dữ liệu. Quá trình xử lý dữ liệu biểu mẫu và chạy các lệnh cơ sở dữ liệu cần thiết tương đối nhiều so với lượng sức mạnh xử lý và băng thông cần thiết để gửi yêu cầu POST. Cuộc tấn công này sử dụng sự chênh lệch về mức tiêu thụ tài nguyên tương đối, bằng cách gửi trực tiếp nhiều yêu cầu bài đến máy chủ được nhắm mục tiêu cho đến khi dung lượng của nó bị bão hòa và xảy ra từ chối dịch vụ.
   1. Công cụ VMware
      1. Cài đặt máy chủ hệ điều hành ảo trong VMware

Thủ sẵn các thư mục hệ điều hành cần cài đặt trong Vmware:

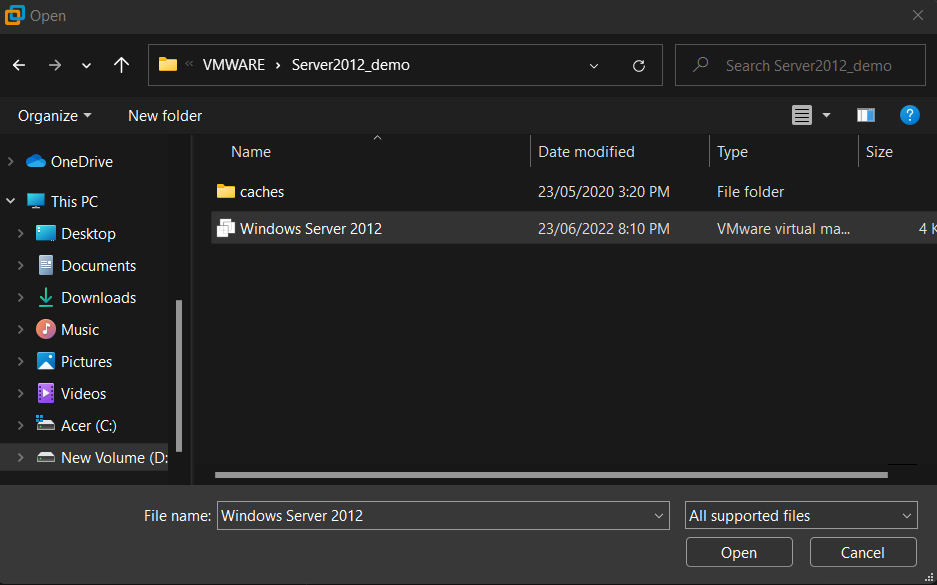


Các thư mục file tệp gồm có Windows Server 2012 và 2 máy Windows 10

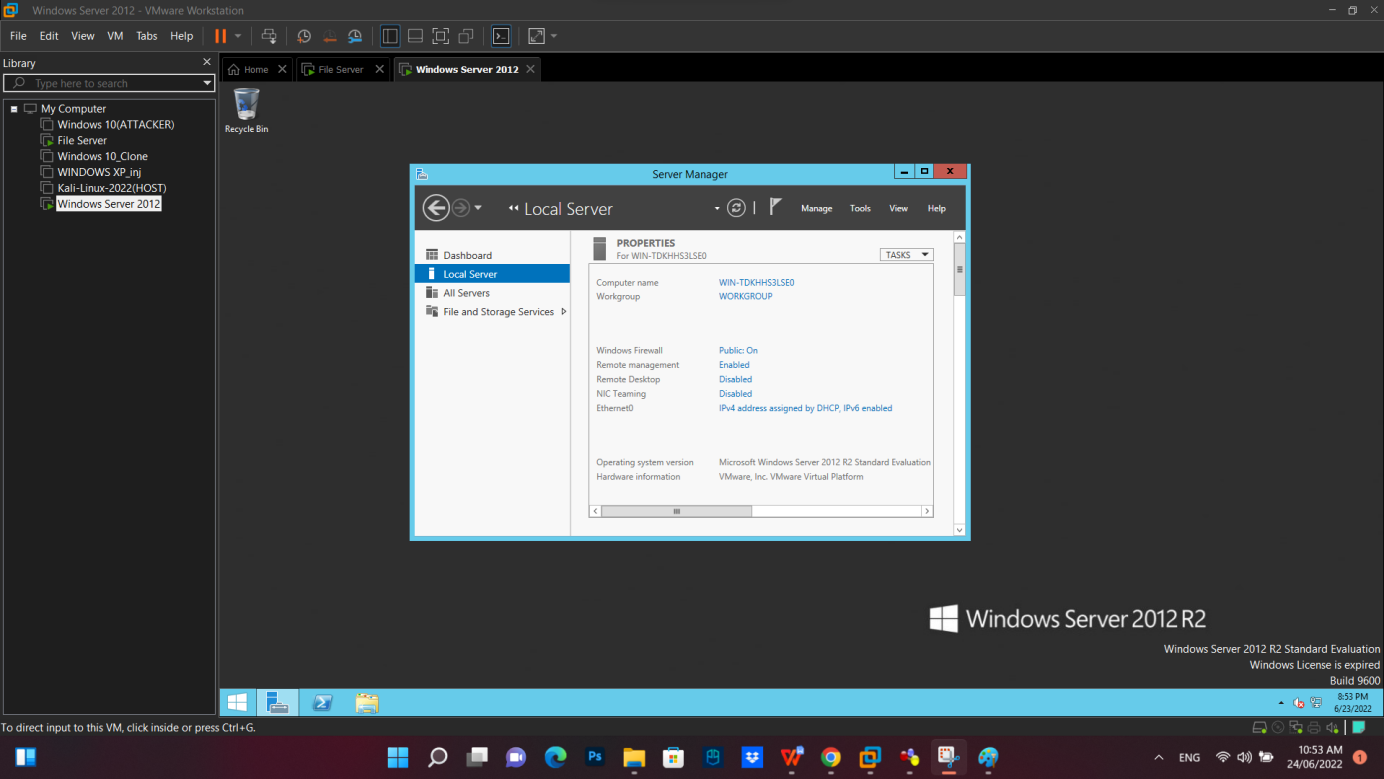
Bên trong VMware ta bấm vào Open a Virtual Machine tại giao diện Home hoặc ta thực thi File -> Open..



Click chọn vào nút “Open a Virtual Machine”



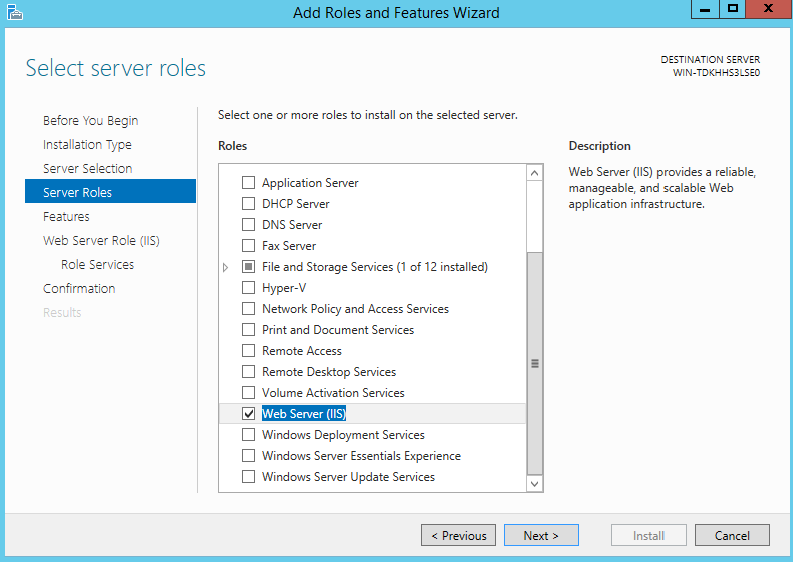
Lựa chọn tệp Windows Server 2012.vmx để tạo máy chủ ảo trong VMware



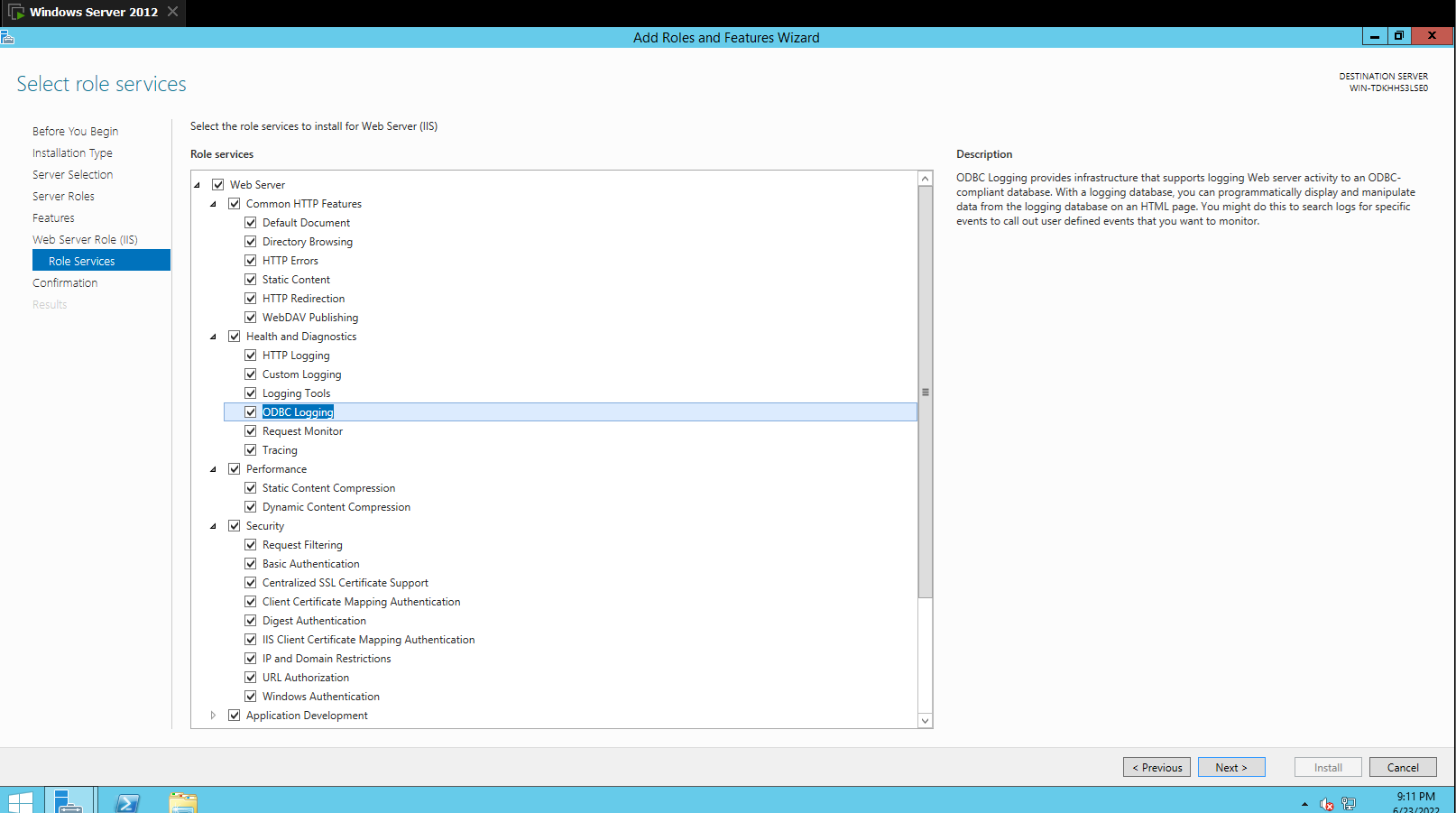
Nhấn chạy Power On và ta khởi động được máy chủ ảo Windows Server

* + 1. Cài đặt dịch vụ trong Windows Server 2012

Để khởi tạo cấu hình giao thức HTTP,HTTPS chúng ta sẽ cài đặt khởi chạy Web Server(IIS). Bên cạnh đó chúng ta có các dịch vụ bỗ trợ khác như AD NS, DNS,DHCP,.. Để cài đặt dịch vụ ta chọn ***Add Roles and Features Wizard***



Chọn server roles là Web Server(IIS)

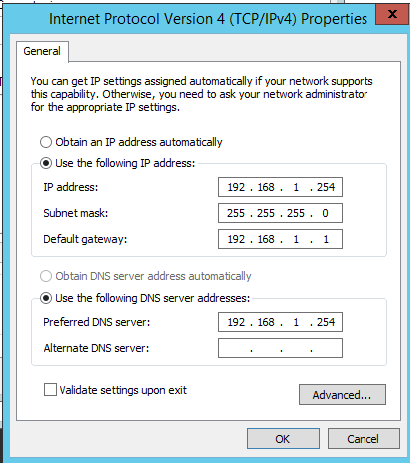


Chọn tất cả trong Role services của Web server(IIS)

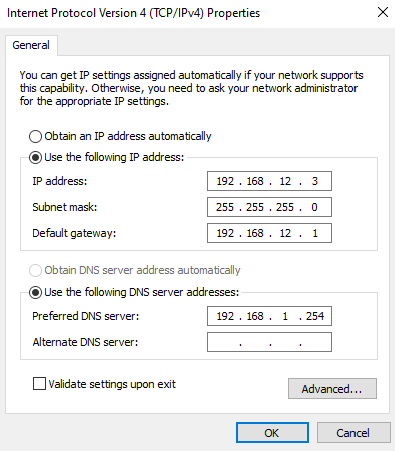
* 1. Các bước thực hiện

**Bước 1:** Cài đặt cấu hình cơ bản địa chỉ Ipv4 cho từng máy

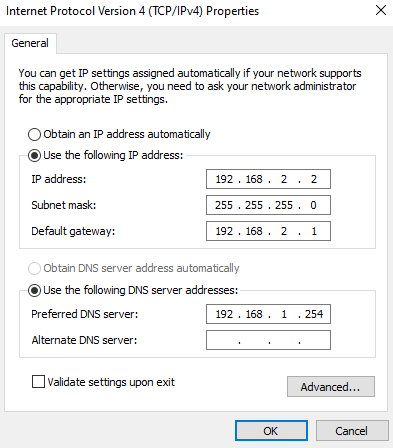
Server:



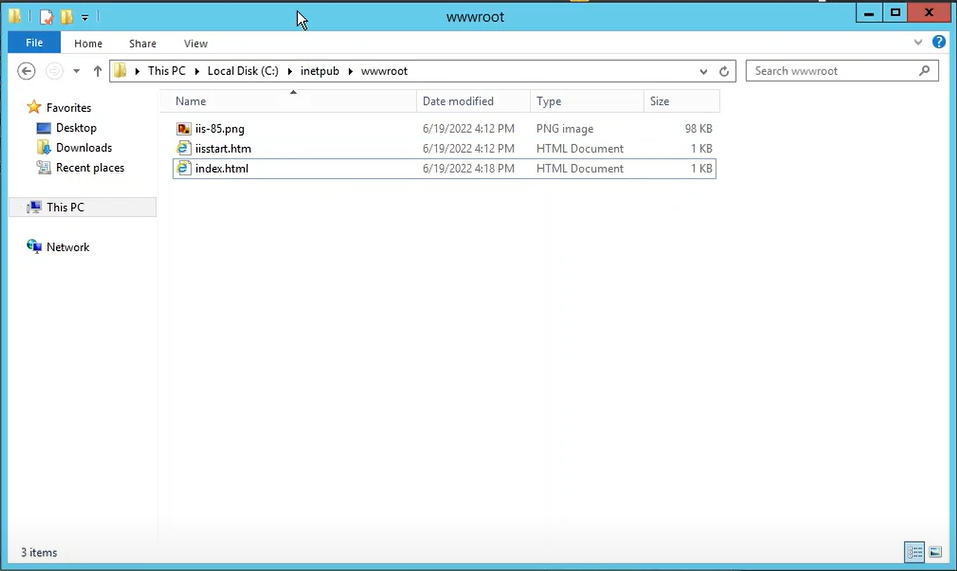
Client:



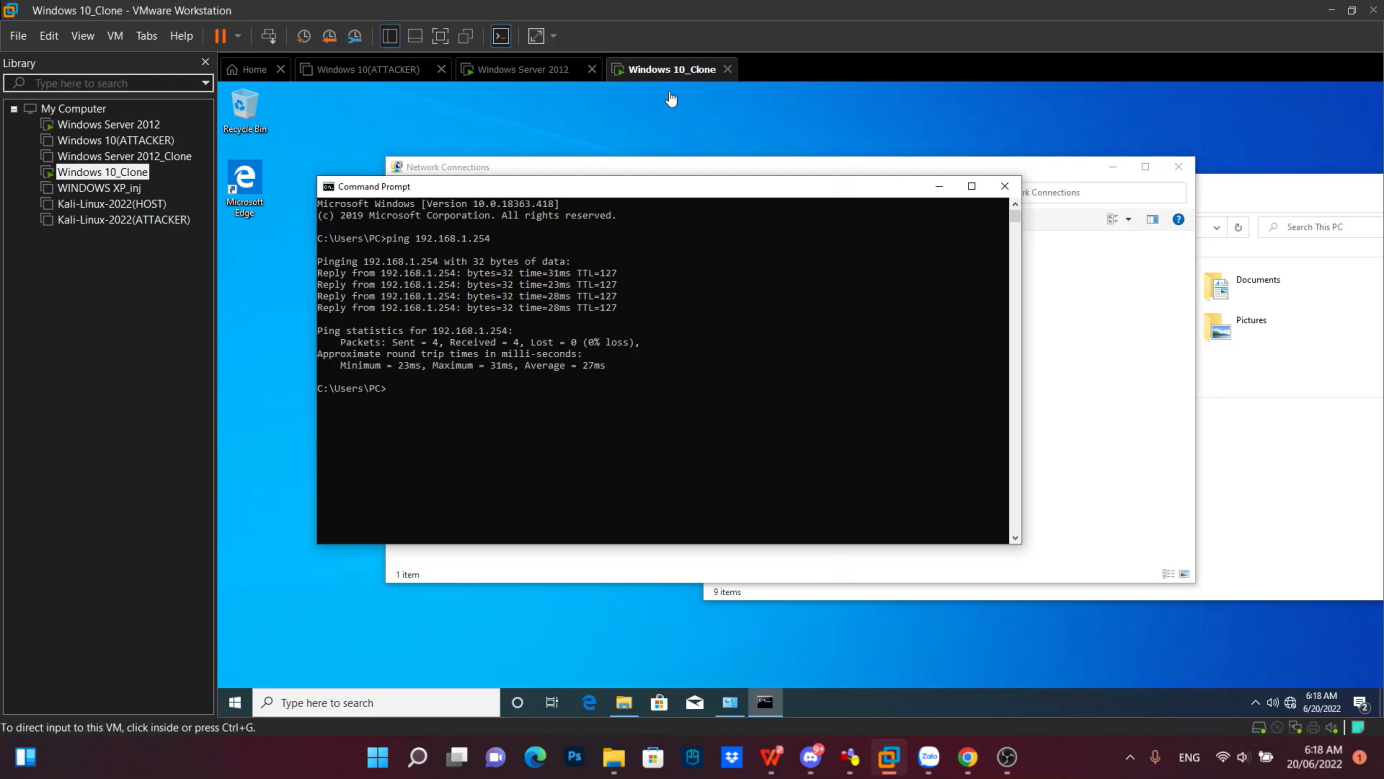
Attacker:

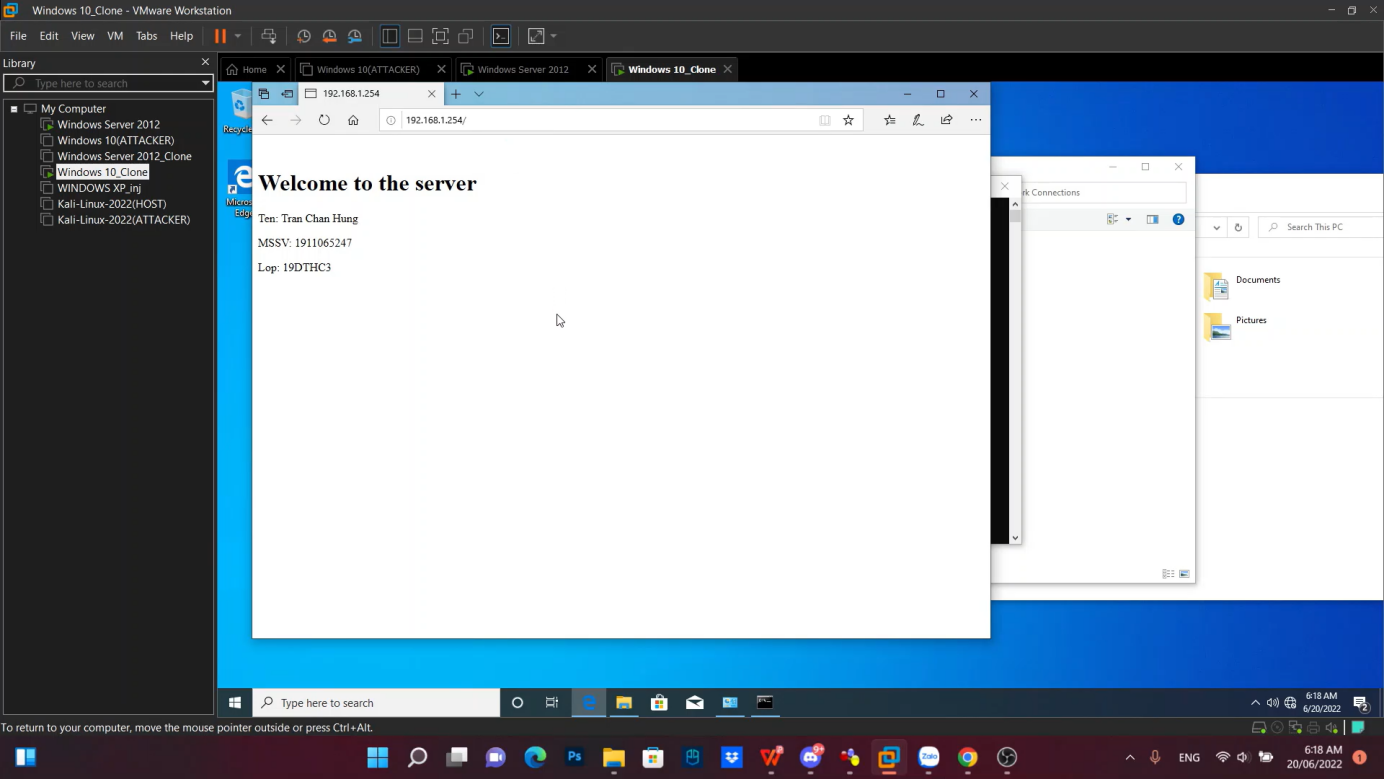


**Bước 2:** Ping qua các máy thành công và chạy giao diện web server thành công



Bỏ file tệp index.html vào thư mục wwwroot của inetpub

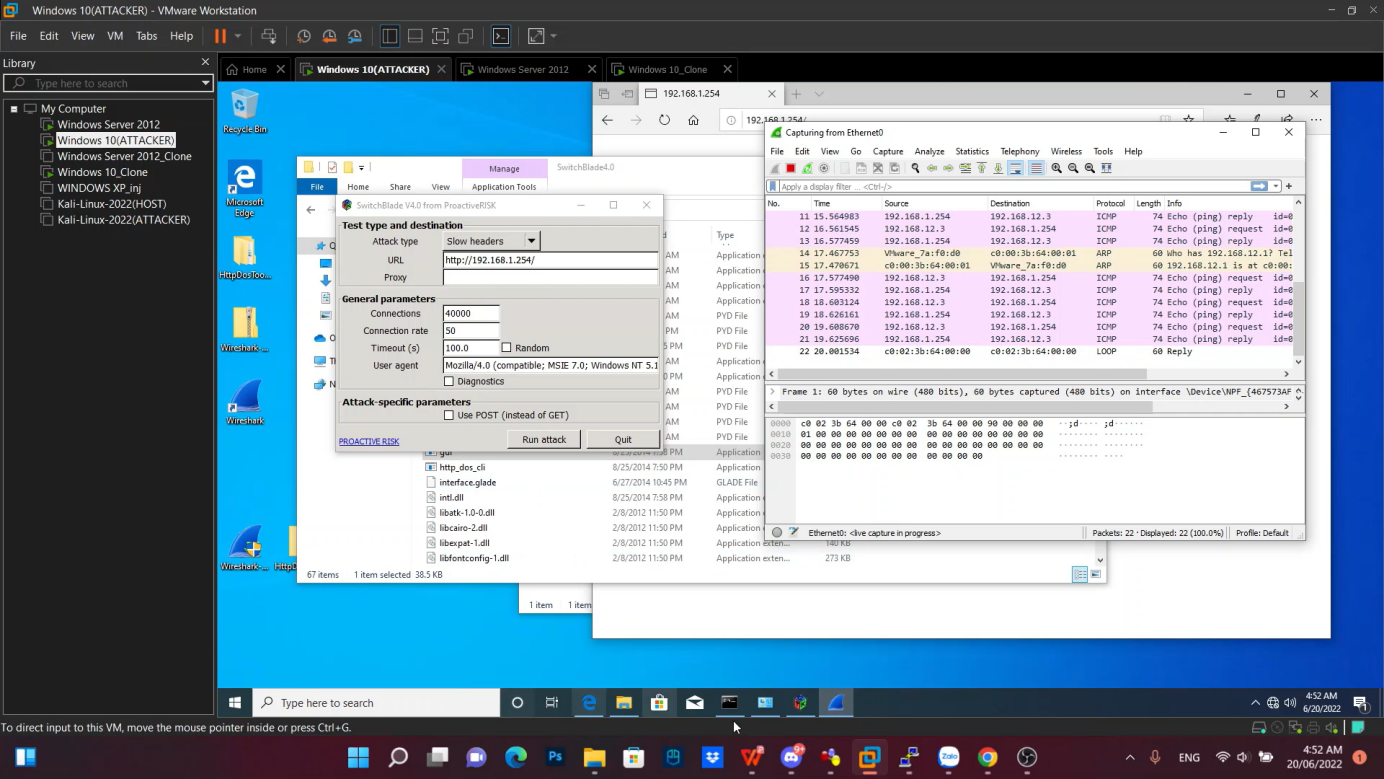




Client ping qua máy server thành công và mở được web server

**Bước 3:**  Thực hiện tấn công HTTP dos flood

Khởi động phần mềm http dos SwitchBlade v4.0:

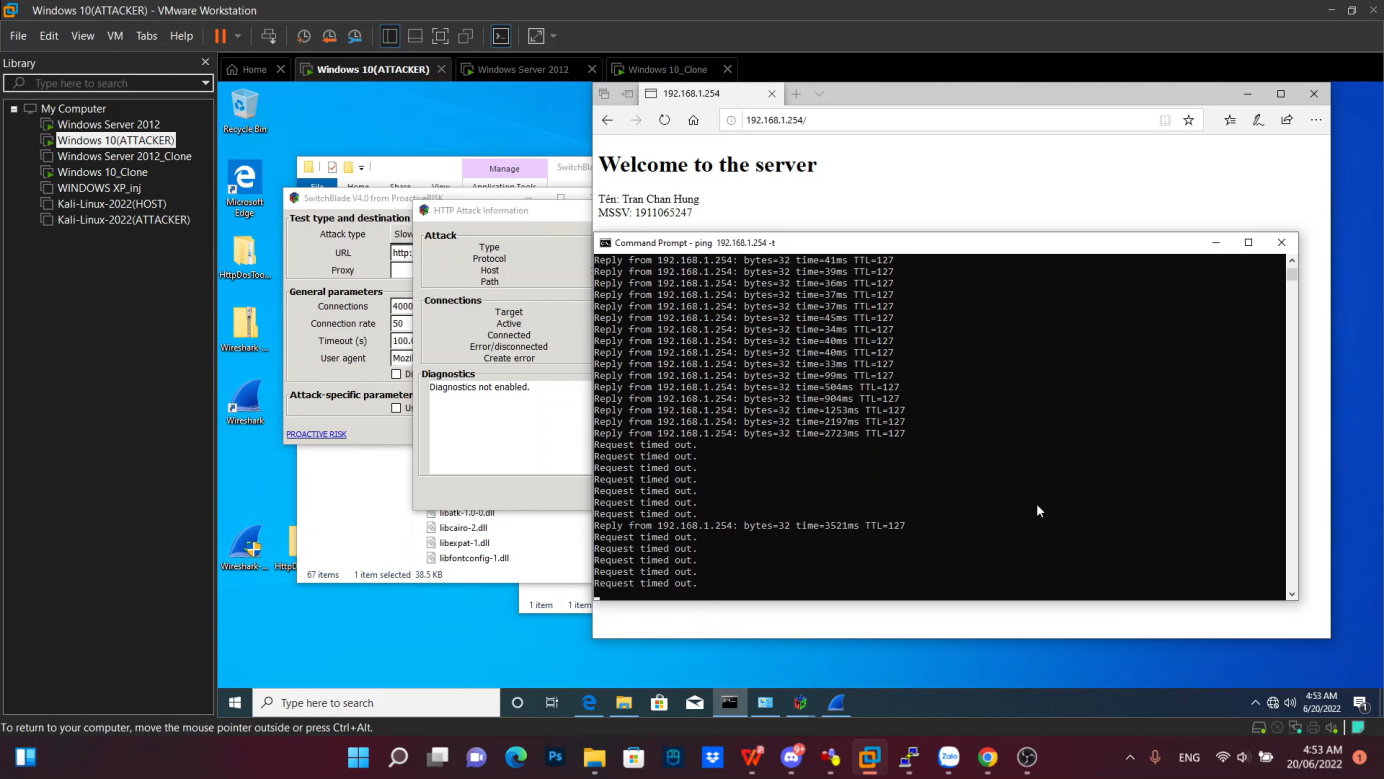


Nhập URL đúng địa chỉ IP bao gồm cả http:// và sau đó chúng ta nhấn Run attack

* 1. Kết quả
     1. Kết quả sau khi tấn công Http dos flood

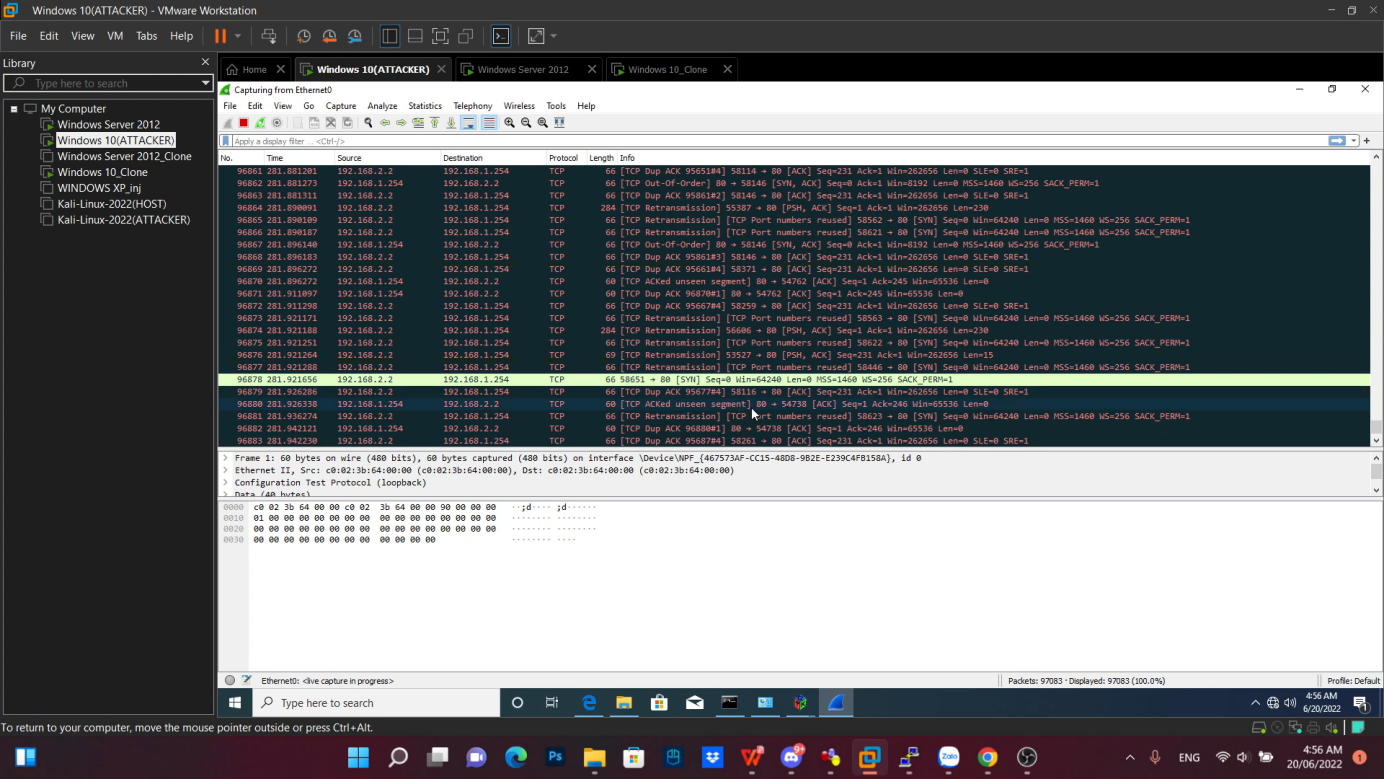
Làm quá trình load web chậm và ping của máy server(192.168.1.254) tăng mạnh

-> Làm cho quá trình pinging thành Request Timed out



* + 1. Thử nghiệm phân tích mạng qua Wireshark

Phần mềm Wireshark phân tích từ máy Attacker (192.168.2.2) tấn công làm chậm máy Server:



* 1. Kết luận và phát triển

Không chỉ HTTP,HTTPS giúp người dùng truy cập trang web được kết nối có độ bảo mật kĩ càng.Tuy nhiên, vẫn còn mặt hạn chế khi vẫn còn một số lỗ hổng rò rĩ rất dễ bị kẻ xấu tấn công.

Các cuộc tấn công HTTP flood rất khó phân biệt với lưu lượng hợp lệ vì chúng sử dụng các yêu cầu URL tiêu chuẩn. Điều này làm cho chúng trở thành một trong những thách thức bảo mật không dễ bị tổn thương tiên tiến nhất hiện nay đối với các máy chủ và ứng dụng. Phát hiện dựa trên tỷ lệ truyền thống là không hiệu quả trong việc phát hiện các cuộc tấn công HTTP flood, vì lưu lượng traffic trong HTTP flood thường nằm dưới ngưỡng phát hiện.

Cho nên hiện nay đã có rất nhiều các bước phòng chống việc tấn công qua hình thức HTTP Dos Flood Attack.

Thì cơ chế giảm thiểu hiệu quả cao nhất dựa trên sự kết hợp của các phương pháp định hình lưu lượng truy cập, bao gồm xác định địa chỉ IP, theo dõi hoạt động bất thường và sử dụng các thách thức bảo mật tiến bộ (ví dụ: yêu cầu phân tích JavaScript).

Giải pháp bảo vệ ứng dụng web của Imperva dựa trên một công cụ phân loại khách hàng duy nhất để phân tích và phân loại tất cả lưu lượng truy cập trang web đến. Giải pháp chống DDoS này được thiết kế đặc biệt để xác định một cách minh bạch lưu lượng bot độc hại, ngăn chặn tất cả các trận lũ HTTP và các cuộc tấn công DDoS của Lớp ứng dụng (Lớp 7 OSI) khác.

1. TÀI LIỆU KHAM KHẢO

<https://viblo.asia/p/tim-hieu-ve-http-hypertext-transfer-protocol-bJzKmgewl9N>

<https://ssl.vn/http-flood-la-gi-lam-the-nao-de-phong-chong-tan-cong-http-flood.html>

<https://xuanthulab.net/giao-thuc-http-va-cau-truc-co-ban-cua-http-message.html>