Lý thuyết

-Tìm hiểu giao thức HTTP

+Các method phổ biến gồm:

* GET:

GET được sử dụng để lấy lại thông tin từ Server đã cung cấp bởi sử dụng một URI đã cung cấp. Các yêu cầu sử dụng GET nên chỉ nhận dữ liệu và nên không có ảnh hưởng gì tới dữ liệu.

* POST:

Một yêu cầu POST được sử dụng để gửi dữ liệu tới Server, ví dụ, thông tin khách hàng, file tải lên, …, bởi sử dụng các mẫu HTML.

* PUT:

Thay đổi tất cả các đại diện hiện tại của nguồn mục tiêu với nội dung được tải lên.

* HEAD:

Tương tự như GET, nhưng nó truyền tải dòng trạng thái và khu vực Header.

* DELETE:

Gỡ bỏ tất cả các đại diện hiện tại của nguồn mục tiêu bởi URI.

* CONNECT:

Thiết lập một tunnel tới Server được xác định bởi URI đã cung cấp.

* OPTIONS:

Miêu tả các chức năng giao tiếp cho nguồn mục tiêu.

* TRACE:

Trình bày một vòng lặp kiểm tra thông báo song song với path tới nguồn mục tiêu.

+Cấu trúc của 1 URL:

URL(viết tắt là Uniform Resource Locator) là 1 định danh duy nhất cho tài nguyên web mà qua đó tài nguyên đó có thể được truy xuất. Cấu trúc URL hầu hết được biểu diễn như sau:

Protocol://hostname/[:port]/[path/]file[?param=value]

+Các status code phổ biến:

1xx: Thông tin

* 100 continue: được gửi trong một số trường hợp khi client gửi yêu cầu có chứa nội dung. Respone chỉ ra rằng các request header đã được nhận và client sẽ tiếp tục gửi nội dung. Máy chủ trả về response thứ hai khi request đã được hoàn thành.

2xx: Thành công

* 200 Ok: cho biết rằng request đã thành công và response sẽ chứa kết quả của request.

3xx: Máy client được chuyển hướng đến tài nguyên khác

* 301 Moved Permanently: chuyển hướng trình duyệt vĩnh viễn đến một URL khác, được chỉ định trong Location header. Client nên sử dụng URL mới trong tương lai thay vì URL ban đầu.
* 302 Found: chuyển hướng trình duyệt tạm thời đến một URL khác, được chỉ định trong Location header. Client sẽ trở về URL ban đầu trong các request tiếp theo.

4xx: Request gặp lỗi

* 400 Bad Request: cho biết rằng client đã gửi một HTTP request không hợp lệ. Client có thể sẽ gặp phải điều này khi họ đã sử đổi một request theo những cách không hợp lệ chẳng hạn như bằng cách đặt một ký tự không hợp lý vào URL.
* 401 Unauthorized: chỉ ra rằng máy chủ yêu cầu xác thực HTTP trước khi request được thực hiện. Header WWW-Authenticate chứa thông tin chi tiết về các loại xác thực được hỗ trợ.
* 403 Forbidden: chỉ ra rằng không ai được phép truy cập tài nguyên được yêu cầu, kể cả đã được xác thực.
* 404 Not Found: chỉ ra rằng tài nguyên được yêu cầu không tồn tại.

5xx: Máy chủ gặp lỗi khi thực hiện request

* 500 Internal Server Error: cho biết rằng máy chủ đã gặp lỗi khi thực hiện yêu cầu. Điều này thường xảy ra khi người dùng gửi thông tin đầu vào không như mong muốn và gây ra lỗi chưa được khắc phục ở đâu đó trong quá trình xử lý ứng dụng. Người dùng nên xem xét kỹ toàn bộ nội dung response của máy chủ để biết chi tiết nào chỉ ra tình trạng của lỗi.
* 503 Service Unavailable: thường chỉ ra rằng, mặc dù bản thân máy chủ web đang hoạt động và có thể phản hồi các request nhưng ứng dụng được truy cập qua máy chủ lại không phải hồi. Người dùng nên xem lại xem đây có phải là kết quả của bất kỳ hành động nào họ đã thực hiện hay không.

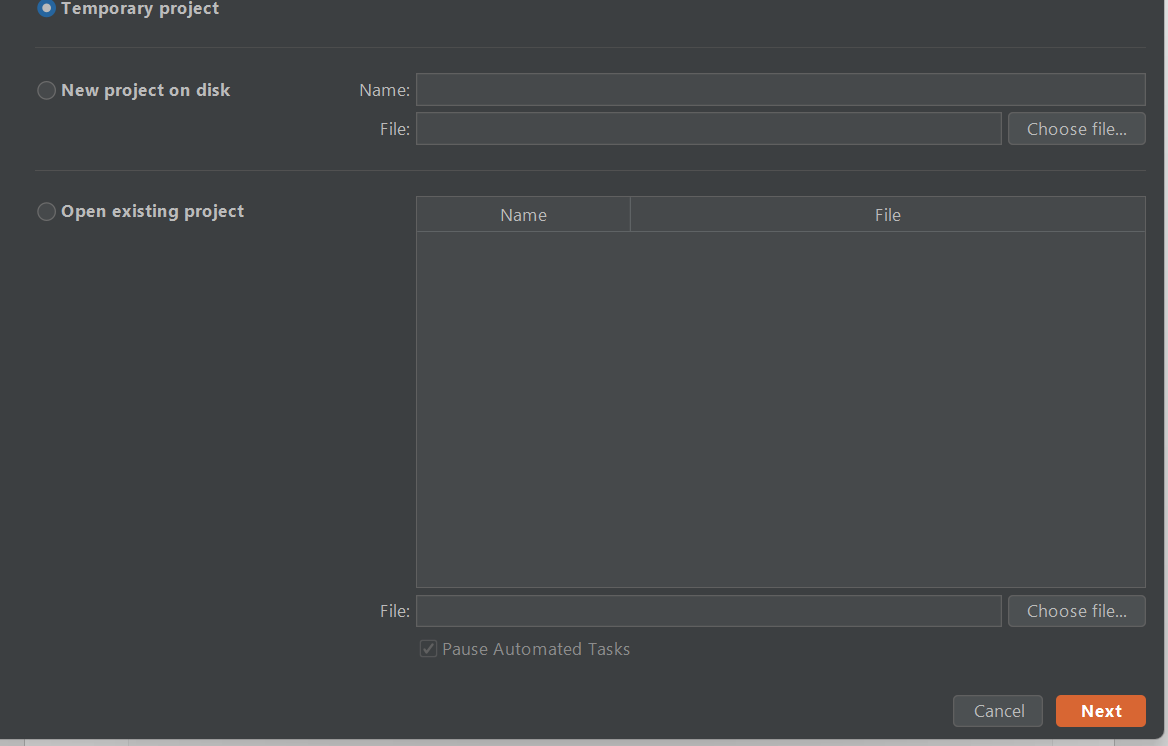
+ Các Header:

* Host: chỉ định tên máy chủ cùng với số port của máy chủ mà request đang gửi đến.
* User-agent: giúp server xác định loại ứng dụng truy cập đến server, hệ điều hành, nhà phân phối phần mềm và phiên bản phần mềm của client.
* Content-Type: chỉ loại media của tài nguyên - MIME types
* Content-Length: chỉ định độ dài của nội dung message, tính bằng byte (ngoại trừ trường hợp responses đối với HEAD requests, khi nó chỉ ra độ dài của nội dung trong responses đối với GET requests tương ứng).
* Cookie: chứa các HTTP cookie liên quan đến server (có thể được gửi tới client từ trước bởi server với header là Set-Cookie hoặc được tạo bằng Javascript thông qua lệnh Document.cookie).
* Location: được sử dụng trong các response chuyển hướng (những response có status code là 3xx (redirection) hoặc 201 (created)) để xác định URL mà page muốn chuyển hướng đến.

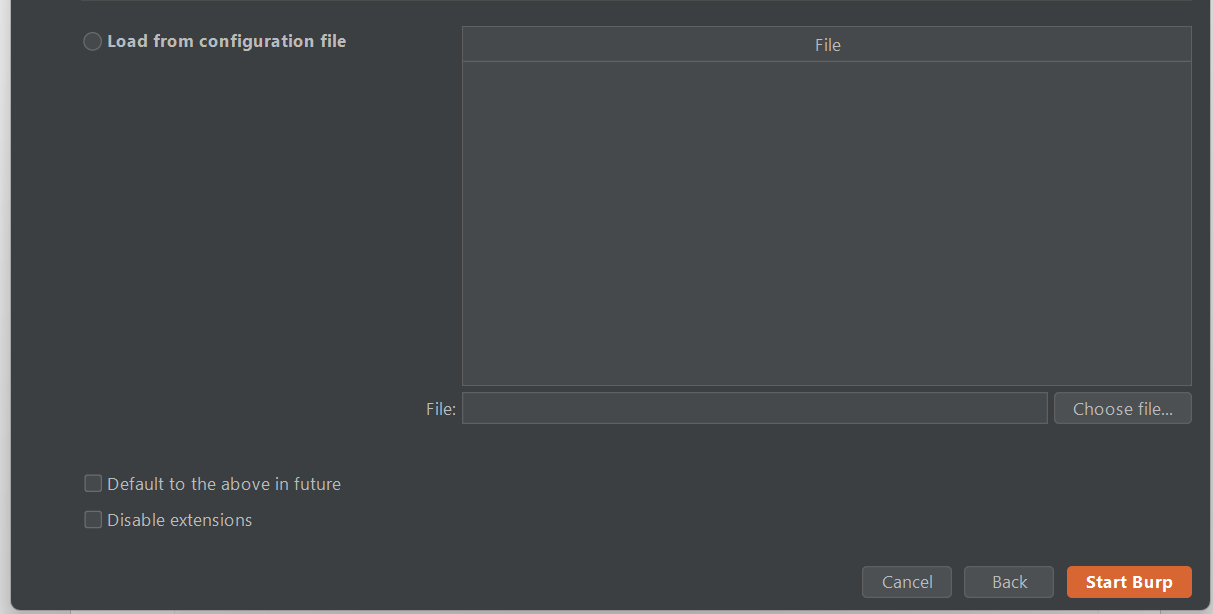
+ Các kiểu encode data trong POST request

* application / x-www-form-urlencoding – Đại diện cho một hình thức được mã hóa URL. Đây là giá trị mặc định nếu thuộc tính enctype không được đặt thành bất cứ thứ gì.
* multipart / form-data – Đại diện cho một hình thức Multipart. Loại biểu mẫu này được sử dụng khi người dùng muốn tải lên tệp
* text / plain – Một loại biểu mẫu mới được giới thiệu trong HTML5, như tên cho thấy, chỉ cần gửi dữ liệu mà không cần mã hóa
* Tìm hiểu công cụ BurpSuite để intercept HTTP Request

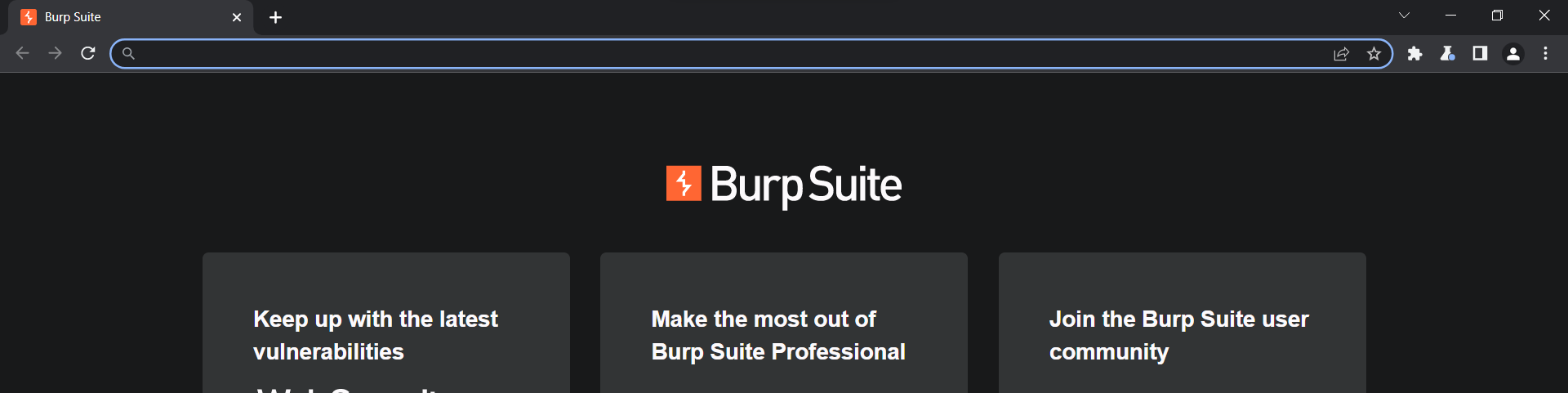
Mở BurpSuite -> Chọn next



Chọn start Burp

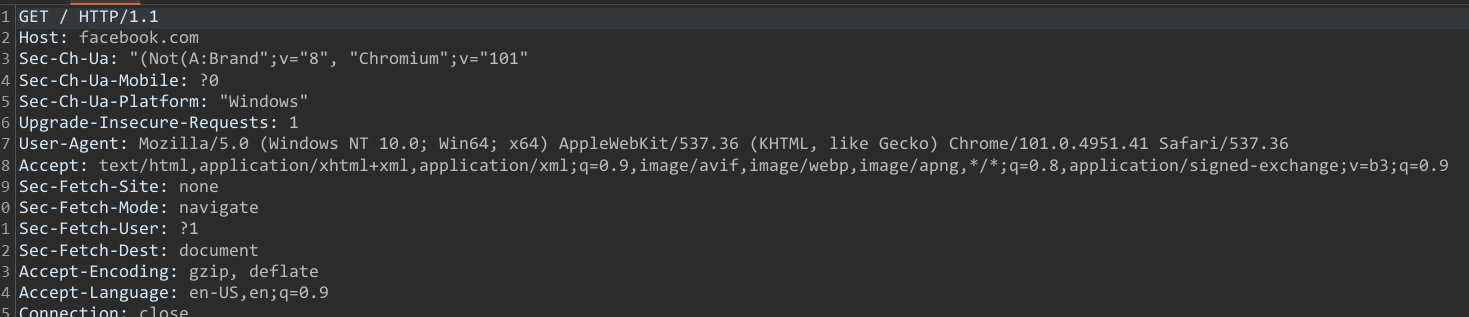


Vào proxy chọn OpenBrowser



Trên Burpsuite chọn Intercept is on

Gọi đến facebook.com burpsuite sẽ bắt được request



Ctrl+R để chuyển sang chế độ Repeater. Có thể sửa đổi Header, gửi request nhiều lần

