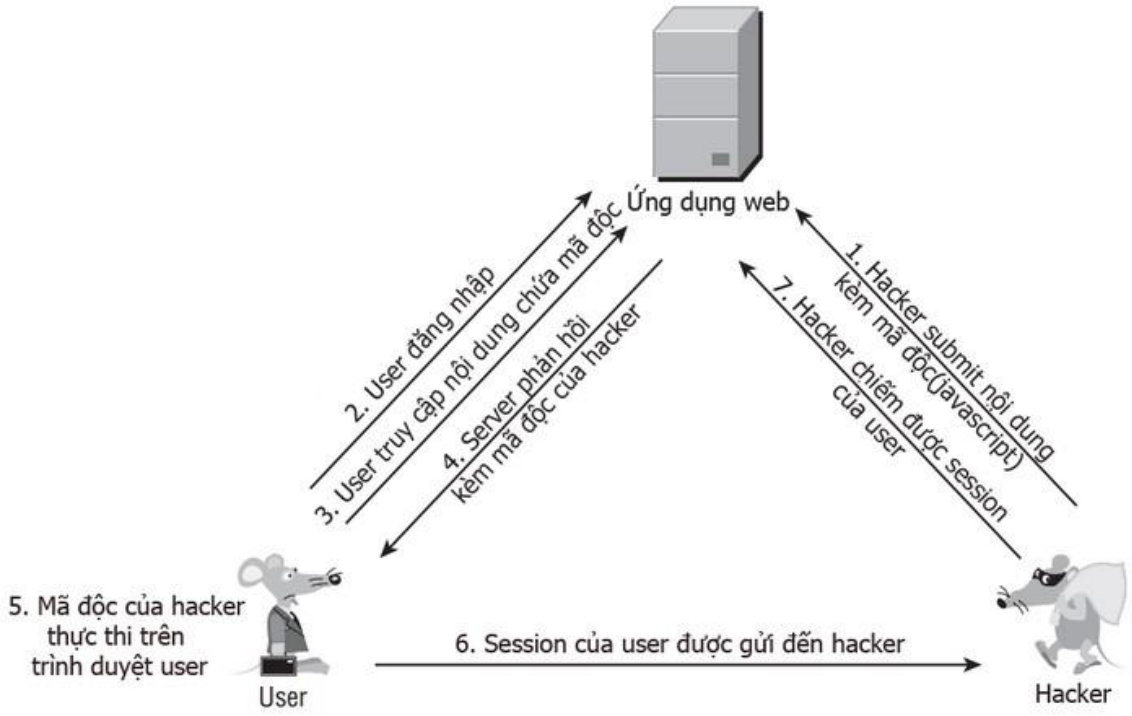
Reflected XSS

1. **Khái niệm XSS:**
   - XSS hay Cross-Site Scripting là một trong những tấn công nguy hiểm nhất đối với các ứng dụng web và có thể mang lại những hậu quả nghiêm trọng.
   - XSS có hình thức tấn công là việc chèn mã độc thông qua các đoạn script để thực thi chúng ở phía Client và từ đó vượt qua truy cập và mạo danh người dùng, ăn cắp dữ liệu nhận dạng người dùng (cookies, session ID)
2. **Hình thức tấn công:**
   - Tấn công Cross Site Scripting nghĩa là gửi và chèn lệnh và script độc hại, những mã độc này thường được viết với ngôn ngữ lập trình phía client và thông thường là Javascript và HTML.
   - Những mã độc có thể được phản chiếu trên trình duyệt của nạn nhân hoặc được lưu trữ trong cơ sở dữ liệu và được chạy mỗi khi người dùng gọi chức năng thích hợp. -> Khi không được lọc hay bảo mật tốt thì mã độc sẽ được thực thi khi người dùng đăng nhập.
   - Để một lỗi XSS xảy ra, phải bao gồm 2 quá trình:
     Thứ 1: Kẻ tấn công chèn các đoạn mã độc vào hệ thống web
     Thứ 2: Người dùng truy cập vào trang web
   - Mô hình các bước XSS tấn công:



3. **Reflected XSS:**
   - Đây là 1 dạng tấn công trong XSS, cùng với các bước như trên nhưng cụ thể ở đây, hacker sẽ lấy thông tin của người dùng thông qua các tham số ở url.
   - Hacker sẽ tạo 1 đường link với tham số url để lấy thông tin người dùng và bằng cách nào đó gưi cho người dùng và buộc họ bấm vào hoặc vô tình bấm vào.
   - Giả sử trang web của ta cho phép người dùng tìm kiếm sản phẩm theo tên, cách triển khai của hệ thống là truyền tham số tìm kiếm thông qua phương thức GET trên URL.

     VD: https://abc.com/search.php?keyword=yua_mikami.

- Nhưng ở phân đoạn thông tin cần get thì hacker truyền 1 đoạn script với phương thức lấy dữ liệu của người dùng -> khiên cho khi kích vào đường link đoạn url này, dữ liệu người dùng sẽ bị lấy mất.

  VD: https://abc.com/search.php?keyword=<script>alert("XSS attack!");</script>

4. Phương pháp ngăn chặn:

+) Validation (Xác thực):

- Đưa ra những quy chuẩn để xác thực thông tin đầu vào.
- Nếu đầu vào khả nghi thì ngăn chặn.
- Tuy nhiên, cách này chỉ mang ý nghĩa giảm thiểu rủi ro, chứ không ngăn chặn được hẳn.

+) Flitering (Lọc dữ liệu):

- Lọc các dữ liệu mang những từ khóa nguy hiểm và xóa chúng như:

  +) Thẻ <script> </ script>

  +) Lệnh Javascript

  +) Đánh dấu HTML

  …

  Những từ khóa mà có thể gây ảnh hưởng đến chương trình và web.

+) Escape (mã hóa các ký tự lập trình bằng các mã đặc biệt)

Character	Escape Code	Character	Escape Code
SPACE	%20	[	%5B
<	%3C	]	%5D
>	%3E	`	%60
#	%23	;	%3B
%	%25	/	%2F
{	%7B	?	%3F
}	%7D	:	%3A
|	%7C	@	%40
\	%5C	=	%3D
^	%5E	&	%26
~	%7E	$	%24

Các ký tự đặc biệt dùng để chạy các đoạn mã lập trình gây ảnh hưởng xấu đến chương trình sẽ được mã hóa để tránh việc thực thi các mã lệnh đó.

5. **Kết luận:**
   - *(Cái này mày tự nghĩ để kết bài cho mượt nhớ Thái =)))))).*