# Assignment 2: Research on Web Security Threats

Name: Abdulgafur Adan

Reg No: C024/401419/2023

Course: BCS 2.2

Date: April 25, 2025

## Introduction

Web applications are essential in modern digital services, but they are also primary targets for various security threats. This report examines five critical web programming threats-SQL Injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), Session Hijacking, and Man-in-the-Middle (MITM) attacks. For each threat, we provide an explanation, real-world example, and mitigation strategies to ensure web developers can design more secure systems.

## 1. SQL Injection

SQL Injection (SQLi) is a code injection technique where attackers exploit vulnerabilities in input fields by injecting malicious SQL code. This can lead to unauthorized access, data leakage, or even complete database deletion.

Case Study: In 2012, hackers breached Yahoo Voices and leaked over 450,000 email addresses and passwords using a simple SQL injection.

Mitigation:

- Use prepared statements and parameterized queries

- Validate and sanitize all user inputs

- Limit database permissions for web applications

## 2. Cross-Site Scripting (XSS)

Cross-Site Scripting (XSS) allows attackers to inject malicious scripts into webpages, which then run in a user's browser. This can lead to session theft, defacement, or redirection to malicious sites.

Case Study: MySpace in 2005 suffered an XSS worm that automatically propagated by injecting malicious JavaScript into profiles.

Mitigation:

- Encode user-generated content before rendering
- Use Content Security Policy (CSP)
- Validate input and escape output

## 3. Cross-Site Request Forgery (CSRF)

Cross-Site Request Forgery (CSRF) tricks a user's browser into executing unwanted actions on a web app in which they're authenticated. This often results in unauthorized fund transfers or data changes.

Case Study: In 2008, Netflix faced a CSRF vulnerability allowing attackers to change user account settings without consent.

Mitigation:

- Use anti-CSRF tokens
- Validate HTTP Referer headers
- Implement SameSite cookie attributes

## 4. Session Hijacking

Session hijacking occurs when an attacker steals a user's session ID to impersonate them and gain unauthorized access. This is often achieved through packet sniffing, XSS, or man-in-the-middle attacks.

Case Study: Firesheep, a Firefox extension released in 2010, demonstrated how easily session hijacking could occur over public Wi-Fi networks.

Mitigation:

- Use HTTPS for all communication
- Regenerate session IDs after login
- Implement secure and HttpOnly cookie flags

## 5. Man-in-the-Middle (MITM) Attacks

Man-in-the-Middle (MITM) attacks involve an attacker secretly intercepting and possibly altering the communication between two parties. These attacks commonly occur on insecure Wi-Fi networks.

Case Study: In 2011, the DigiNotar breach allowed attackers to issue fraudulent SSL certificates, enabling MITM attacks against Gmail users in Iran.

Mitigation:

- Use strong encryption with HTTPS and TLS
- Employ VPNs in insecure environments
- Validate digital certificates rigorously

## Conclusion

The growing complexity and interactivity of web applications come with increased security risks. Understanding these common threats and implementing best practices can significantly reduce vulnerabilities. Developers must stay informed about the latest attacks and continuously improve security protocols to protect user data and maintain trust.