**SAD Lab**

**Exp-5**

**Name: Shashwat Tripathi**

**Division: D20A**

**Roll no: 64**

**Aim:** Study of OWAPS Vulnerabilities

**Theory:**

**What is OWASP?**

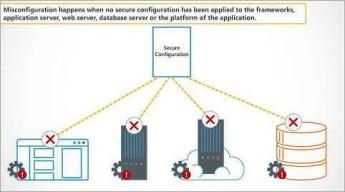
OWASP (Open Web Application Security Project) is a nonprofit foundation that works to improve the security of software. The OWASP Top Ten is a list of the most critical web application security risks.

**Types of Vulnerabilities and Their Mitigation:**

## **Security Misconfiguration:**

**Description**: Security misconfiguration occurs when security settings are incorrectly configured or left at insecure defaults.

## **Diagram**:



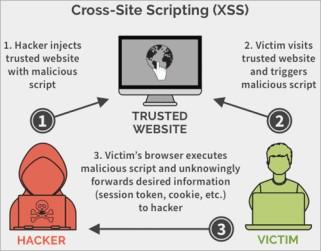
**Mitigation Steps**:

* + Regularly update and patch software.
  + Remove unnecessary features (e.g., unused services, default accounts).
  + Implement secure configurations for all application layers (servers, databases, etc.).

## **Cross-Site Scripting (XSS):**

**Description**: XSS flaws occur when untrusted data is injected into a web page and executed by a victim’s browser, allowing attackers to hijack sessions or redirect users.

## **Diagram**:



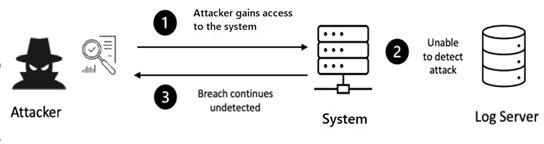
**Mitigation Steps**:

* + Sanitize and validate all user inputs.
  + Encode output properly in HTML, CSS, JavaScript, etc.
  + Implement Content Security Policy (CSP).

## **Insufficient Logging and Monitoring:**

**Description**: Insufficient logging and monitoring can delay detection of security breaches, increasing the impact of attacks.

## **Diagram**:



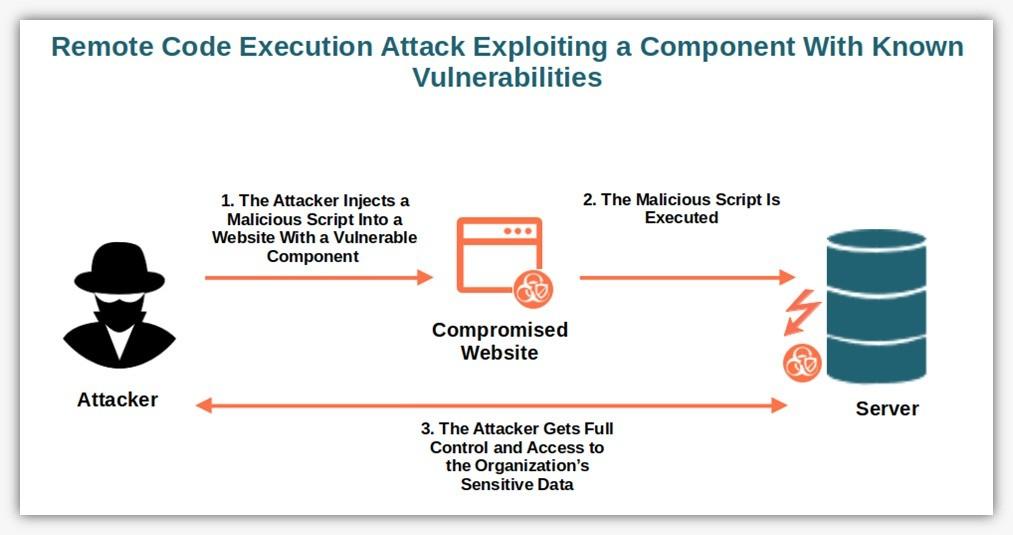
**Mitigation Steps**:

* + Implement logging of all security-relevant events.
  + Ensure logs are stored securely and regularly monitored.
  + Set up automated alerting for suspicious activity

## **Using Components with Known Vulnerabilities:**

**Description**: This vulnerability occurs when a web application uses libraries, frameworks, or components with known security flaws.

## **Diagram**:



**Mitigation Steps**:

* + Regularly update and patch dependencies.
  + Use automated tools to check for vulnerabilities in third-party components.
  + Avoid using unsupported or unmaintained libraries.

## **Insecure Deserialization:**

**Description**: Insecure deserialization occurs when data is improperly deserialized, leading to remote code execution or other attacks.

## **Diagram**:

## 

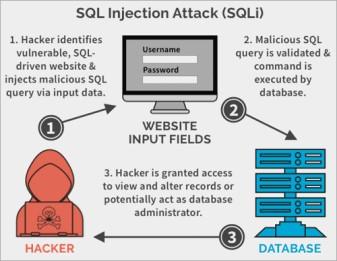
**Mitigation Steps**:

* + Use serialization frameworks that are safe and secure.
  + Validate and sanitize data during deserialization.
  + Implement integrity checks on serialized data (e.g., using cryptographic signatures).

1. **Injection (e.g., SQL Injection):**

**Description**: Injection flaws occur when untrusted data is sent to an interpreter as part of a command or query. This can lead to unauthorized access to sensitive data.

# **Diagram**:



**Mitigation Steps**:

* + Use prepared statements (parameterized queries).
  + Use stored procedures.
  + Validate and sanitize all user inputs.
  + Apply least privilege access to the database.

## **Broken Authentication:**

**Description**: Broken authentication occurs when an attacker can compromise passwords, session tokens, or exploit flaws in authentication mechanisms.

## **Diagram**:



**Mitigation Steps**:

* + Implement multi-factor authentication (MFA).
  + Enforce strong password policies.
  + Use secure storage for credentials (e.g., hashing with salts).
  + Ensure session management is secure.

## **Sensitive Data Exposure:**

**Description**: Sensitive data, such as credit card numbers and personal information, is improperly protected, leading to data leaks or unauthorized access.

## **Diagram**:



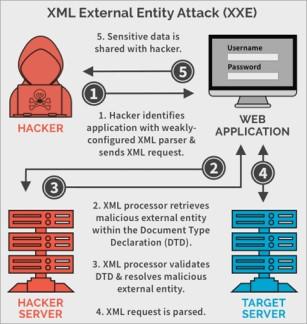
**Mitigation Steps**:

* + Encrypt sensitive data at rest and in transit.
  + Use strong encryption algorithms (e.g., AES, RSA).
  + Enforce secure protocols like HTTPS/TLS.
  + Implement proper key management procedures.

## **XML External Entities (XXE):**

**Description**: XXE vulnerabilities occur when untrusted XML input containing a reference to an external entity is processed by a weakly configured XML parser.

## **Diagram**:



**Mitigation Steps**:

* + Disable external entity processing in XML parsers.
  + Validate and sanitize XML inputs.
  + Use less complex data formats (e.g., JSON).

## **Broken Access Control:**

**Description**: Broken access control occurs when users can access data or functions they shouldn’t due to improper restrictions.

## **Diagram**:



**Mitigation Steps**:

* + Enforce proper access control mechanisms at both client and server levels.
  + Implement role-based access control (RBAC).
  + Perform regular audits of access control policies.

**Conclusion:** Thus, we have studied and understood the OWASP and its vulnerabilities.