OWASP TOP 10

1. Injection:
Injection vulnerabilities occur when an attacker can inject untrusted data into a web application. This can happen when user input is not properly validated or sanitized, allowing the attacker to inject malicious code. The most common types of injection vulnerabilities are SQL injection and command injection.

Remediation: The best way to prevent injection vulnerabilities is to use parameterized queries or prepared statements in the code. This ensures that any user input is properly sanitized and cannot be used to inject malicious code. Method/Technique: Parameterized Queries or Prepared Statements

2. Broken Authentication:
This vulnerability occurs when there are flaws in the authentication and session management mechanisms of a web application. This can lead to attackers gaining unauthorized access to user accounts and sensitive data.

Remediation: The best way to prevent this vulnerability is to use strong passwords, implement multi-factor authentication, and use secure session management techniques such as session expiration and token-based authentication. Method/Technique: Multi-Factor Authentication

3. Sensitive Data Exposure:
This vulnerability occurs when sensitive data is not properly protected, such as passwords or credit card numbers. This can lead to the theft of user data.

Remediation: The best way to prevent this vulnerability is to use strong encryption algorithms, properly salt passwords, and store sensitive data in secure locations such as a dedicated hardware security module (HSM). Method/Technique: Encryption

4. XML External Entities (XXE):
This vulnerability occurs when a web application processes XML input from untrusted sources, allowing an attacker to access or manipulate data on the server or gain sensitive information.

Remediation: The best way to prevent this vulnerability is to disable XML external entities, use a different parser or library that is not vulnerable to XXE attacks, and validate and sanitize all user input. Method/Technique: Disabling XML External Entities

5. Broken Access Control:
This vulnerability occurs when an attacker can gain access to data or functionality that should be restricted. This can happen when access controls are not properly implemented or enforced.

Remediation: The best way to prevent this vulnerability is to use proper access controls and implement role-based access control (RBAC) where appropriate. All access controls should be enforced on both the client and server side. Method/Technique: Role-Based Access Control

6. Security Misconfiguration:
This vulnerability occurs when a web application is not configured securely. This can happen when default settings are not changed, or when unused services or features are not disabled.

Remediation: The best way to prevent this vulnerability is to follow secure coding practices, keep software and servers up to date with the latest patches, and use tools such as vulnerability scanners to identify potential vulnerabilities. Method/Technique: Vulnerability Scanners

7. Cross-Site Scripting (XSS):

This vulnerability occurs when an attacker can inject malicious scripts into a web page, which are then executed by the user's browser. This can lead to the theft of user data, such as session cookies.

Remediation: The best way to prevent XSS vulnerabilities is to validate and sanitize all user input, encode output properly, and implement Content Security Policy (CSP) to restrict the sources of content that can be loaded by a page. Method/Technique: Content Security Policy

8. Insecure Deserialization:
This vulnerability occurs when untrusted data is deserialized by a web application, allowing an attacker to execute arbitrary code or access sensitive data.

Remediation: The best way to prevent this vulnerability is to use safe deserialization methods or libraries, validate and sanitize all user input, and implement input validation on the server side

Here are some of the commonly used methods and tools for testing each vulnerability in the OWASP Top 10:

1. Injection:
- SQLMap
- Havij
- Burp Suite
- OWASP ZAP

2. Broken Authentication:
- OWASP Testing Guide
- Burp Suite
- Zed Attack Proxy (ZAP)
- Open Web Application Security Project (OWASP) Broken Authentication and Session Management Project

3. Sensitive Data Exposure:
- OWASP Testing Guide
- Nmap
- Metasploit

4. XML External Entities (XXE):
- OWASP Testing Guide
- Burp Suite
- Zed Attack Proxy (ZAP)
- XML Quadratic Blowup Attack Tool (BOMB)

5. Broken Access Control:
- OWASP Testing Guide
- Burp Suite
- Zed Attack Proxy (ZAP)
- Open Web Application Security Project (OWASP) Broken Access Control Project

6. Security Misconfiguration:
- OWASP Testing Guide
- Nmap
- Nessus
- OpenVAS

7. Cross-Site Scripting (XSS):

- OWASP Testing Guide
- Burp Suite
- Zed Attack Proxy (ZAP)
- DOMinator

8. Insecure Deserialization:
- OWASP Testing Guide
- Burp Suite
- Zed Attack Proxy (ZAP)
- ysoserial