TASK 2

Overview

Explain about any 10 web application attacks

Web Application Attacks

1. SQL Injection (SQLi):

Attack: Malicious SQL queries are injected into input fields, enabling unauthorized access to databases.

Impact: Attackers can extract, modify, or delete sensitive data.

Prevention: Use parameterized queries, input validation, and output encoding.

2. Cross-Site Scripting (XSS):

Attack: Malicious scripts are injected into web pages, which are then executed by unsuspecting users' browsers.

Impact: Attackers can steal data, hijack user sessions, or deliver malware to users.

Prevention: Implement output encoding and use security mechanisms like Content Security Policy (CSP).

3. Cross-Site Request Forgery (CSRF):

Attack: Malicious websites trick users into executing unwanted actions on other sites where the victim is authenticated.

Impact: Attackers can perform actions on behalf of the victim, such as changing settings or making unauthorized transactions.

Prevention: Utilize anti-CSRF tokens, validate referrer headers, and use the SameSite attribute for cookies.

4. XML External Entity (XXE):

Attack: Attackers exploit insecure XML parsing by including external entities to disclose internal files or execute arbitrary code.

Impact: Sensitive data exposure, server-side request forgery (SSRF), and remote code execution.

Prevention: Disable external entity parsing, use strict XML parsers, and validate user input.

5. Server-Side Request Forgery (SSRF):

Attack: Attackers trick the server into making unauthorized requests to internal resources or external services.

Impact: Attackers can access internal systems, exfiltrate data, or perform denial of service.

Prevention: Implement input validation and firewall rules and restrict resource access.

6. Insecure Deserialization:

Attack: Attackers manipulate serialized data to execute arbitrary code or cause application crashes.

Impact: Potential remote code execution and DoS attacks.

Prevention: Employ strong access controls, input validation, and use safer serialization formats.

7. Broken Authentication:

Attack: Attackers exploit vulnerabilities in user authentication mechanisms to gain unauthorized access.

Impact: Unauthorized access to accounts, data breaches, and identity theft.

Prevention: Implement secure authentication practices, like strong password policies and multi-factor authentication.

8. Security Misconfigurations:

Attack: Attackers exploit misconfigured settings, such as default credentials, directory listings, or overly permissive access controls.

Impact: Unauthorized access, data exposure, and application instability.

Prevention: Regularly audit configurations, use least privilege principles, and follow security best practices.

9. File Upload Attacks:

Attack: Attackers abuse file upload functionality to execute malicious code or upload malware.

Impact: Remote code execution, data breaches, and system compromise.

Prevention: Implement strict file type validation, store uploads in non-executable locations, and scan uploaded files for malware.

10. Path Traversal (Directory Traversal):

Attack: Attackers manipulate input to access files and directories outside of the intended path.

Impact: Unauthorized access to sensitive files, data disclosure, and potential code execution.

Prevention: Implement input validation, use secure coding practices, and employ strong access controls.