❖ OWASP (Open Web Application Security Project) Top 10 Vulnerabilities

1. SQL Injection

- Attackers inject malicious SQL queries into input fields, allowing unauthorized access to a database.
- **Impact:** Data breaches, data manipulation, or database destruction.
- **Prevention:**
    - Use parameterized queries and prepared statements.
    - Validate and sanitize user input.

2. Cross-Site Scripting (XSS)

- Malicious scripts are injected into web pages, which are then executed in a user's browser.
- **Impact:** Theft of user data, session hijacking, and defacement of websites.
- **Prevention:**
    - Use content security policies (CSP).
    - Escape and validate user input.

3. Cross-Site Request Forgery (CSRF)

- Forces users to execute unintended actions on a web application they're authenticated with.
- **Impact:** Unauthorized transactions or changes.
- **Prevention:**
    - Implement anti-CSRF tokens.
    - Validate HTTP request origins.

4. Security Misconfiguration

- Using default configurations, leaving unnecessary features enabled, or exposing sensitive data.
- **Impact:** Exploitation of system weaknesses.
- **Prevention:**

- Regularly audit and update configurations.
- Remove unused features.

5. **Broken Authentication & Session Management**

- Flaws in the authentication system, such as weak passwords or unsecure session tokens.
- **Impact:** Unauthorized access to user accounts.
- **Prevention:**
  - Use strong password policies and multi-factor authentication.
  - Protect session tokens.

6. **Sensitive Data Exposure**

- Exposing sensitive information like credit card numbers, passwords, or personal details.
- **Impact:** Identity theft and financial loss.
- **Prevention:**
  - Encrypt data in transit (HTTPS) and at rest.
  - Avoid storing sensitive data unnecessarily.

7. **Insecure Deserialization**

- Exploiting vulnerabilities in deserialization processes to execute malicious code or tamper with data.
- **Impact:** Data corruption or unauthorized access.
- **Prevention:**
  - Validate and sanitize serialized objects.
  - Avoid accepting untrusted serialized data.

8. **Using Components with Known Vulnerabilities**

- Using outdated libraries or software components with known exploits.
- **Impact:** Allows attackers to exploit these vulnerabilities.
- **Prevention:**

- Regularly update libraries and software.
- Monitor vulnerability databases for known issues.

9. **Insufficient Logging & Monitoring**

- Lack of proper logging mechanisms to detect malicious activities.
- **Impact:** Delayed detection of security breaches.
- **Prevention:**
  - Implement centralized logging.
  - Regularly review logs.

10.XML External Entity (XXE)

- Exploiting XML parsers to access sensitive files or execute malicious commands.
- **Impact:** Data leaks and system compromise.
- **Prevention:**
  - Disable external entity processing in XML parsers.
  - Use modern libraries that prevent XXE attacks.

❖ Website Security Measures

1. CAPTCHA

- **What It Is:** A test to differentiate between humans and bots.
- **Purpose:** Prevent automated attacks like spam and brute-force login attempts.
- **How It Works:** Users must solve puzzles (like selecting images or typing distorted text) to prove they're human.

2. HTTPS (HyperText Transfer Protocol Secure)

- **What It Is:** Encrypts communication between the user's browser and the server.
- **Purpose:** Protects sensitive information like login credentials and credit card details from being intercepted.
- **How It Works:** Uses SSL/TLS certificates to encrypt data.

3. Content Security Policy (CSP)

- **What It Is:** A browser feature that restricts resources a webpage can load.
- **Purpose:** Mitigates cross-site scripting (XSS) and data injection attacks.
- **How It Works:** Developers define policies specifying trusted sources for scripts, styles, and images.

4. Input Validation

- **What It Is:** Ensuring user-provided data is safe and within expected formats.
- **Purpose:** Prevents injection attacks like SQL injection and XSS.
- **How It Works:** Sanitizes input by removing harmful characters and validating formats.

5. Firewall

- **What It Is:** A security system that monitors and filters network traffic.
- **Purpose:** Blocks malicious traffic and protects against attacks like DDoS.
- **How It Works:** Filters traffic based on predefined security rules.