OWASP top10

A1.Broken Access Control

It is like an unauthorized user are allowed to perform actions beyond their permission.

- Least Privilege
- Bypassing Access Control
- IDOR (Insecure Direct Object Reference)
- Missing Access Control in APIs
- Privilege Escalation / Token Manipulation
- Force Browsing
- CORS Misconfiguration

A2.CRYPTOGRAPHIC FAILURES

It means “insecure handling of sensitive data” When the sensitive information are leaked because of

1)lack of encryption

2)or the cryptographic functions or tools are not used properly

- Lack of Encryption
- Use of Weak or Old Algorithms
- Certificate Validation Issues
- Weak Password Management
- Improper Token Generation
- No Encryption at Rest

A3.Injection

It happens when an application takes untrusted input (data from users or other sources) and directly uses it inside commands or queries without proper checks or protections.

- User Input Not Validated or Sanitized
- Dynamic Queries with Concatenation
- No Use of Parameterized Queries
- Hostile Data in ORM Tools
- No Escaping of Special Characters

A4.Insecure Design

It refers to missing or weak security controls that should have been part of the initial architecture or planning phase.

- Abuse of Business Logic
- Insecure Workflow
- Lack of Anti-Bot Measures
- Privilege Escalation
- Resource Abuse

A5.Security Misconfiguration

Security Misconfiguration means not properly setting up security controls in applications, servers, or databases. This can expose sensitive data or allow attackers to exploit the system easily.

- Unnecessary Features Enabled
- Default Accounts and Passwords
- Detailed Error Messages Shown
- Missing or Weak Security Headers
- Outdated or Vulnerable Software
- Upgrades Without New Security Features
- Insecure Framework or Database Settings

A6.Vulnerable and Outdated Components

Vulnerable and Outdated Components means using software that are old, unpatched, or have known security flaws making it easy for attackers to exploit them.

- Use of Old or Unpatched Software
- Not Checking for Known Issues
- Delayed Patching
- No Testing After Updates

A7.Identification and Authentication Failures

This vulnerability happens when the system doesn't properly verify the user’s identity or manage their login session securely.

- Brute Force Possible
- Default or Weak Passwords Allowed
- Insecure Password Storage
- No Multi-Factor Authentication
- Session ID in URL
- Session Fixation
- Sessions Not Logged Out Properly

A8.Software and Data Integrity Failures

Software and data integrity failures happen when an app trusts files, code, or data without checking if they are safe. Attackers can change or replace them and make the app run harmful code.

- Using Code from Untrusted Sources

Developers rely on libraries or modules (e.g., from GitHub, npm, Maven, or CDNs) without checking if they're safe.

- Auto-Updates Without Integrity Checks

Some apps download and install updates automatically without verifying their source or contents.

A9.Security Logging and Monitoring Failures

Security Logging and Monitoring Failures happen when an application doesn't properly keep track of important events — like login attempts, suspicious actions, system errors

Important Events Not Logged

Such as- Timestamps ,user identifiers, IP addresses, Request source, actions ("login failed", "password changed),results(success/failures)

- Logs Stored Only Locally
- Without capturing these events, it is impossible to track user activity, and to check the malicious activity.

A10.SERVER-SIDE REQUEST FORGERY(SSRF)

SSRF happens when an attacker tricks a server into making requests to internal or external systems, often bypassing firewalls or network restrictions.

Attackers can abuse this to access sensitive data, internal services, or perform unauthorized actions.

- Bypassing Firewalls and Network Restrictions
- Accessing Internal Services
- Abuse of URL Parameters
- Fetching Sensitive Data from Internal Systems
- Triggering Unauthorized Actions via Server Requests