# Links

1. <https://www.hacksplaining.com/>
2. <https://owasp.org/>
3. <https://www.exploit-db.com/>

# Foundation of Security

## Secure Product Lifecycle (SDLC)

* Also referred to as Secure software Development Lifecycle (SSDLC.
* Product development Lifecycle diagram

## Information Security

* Refers to
* CIA – Confidentiality, Integrity and Availability

## Security Culture – shift left

## Risks, Threats and Vulnerabilities

## Data governance and privacy

# Principles of protection

## Least Privilege

## Defense in depth

* Layer of defense applied multiple layers of protection.

## STRIDE

S – Spoofing

T – Tampering

R – Repudiation

I – Information disclosure

D – Denial of service

E – Elevation of privilege

## Fail safe/Fail secure

* Fail should happen gracefully.

## Complete mediation

## Session management

## Open design

## Psychological acceptability

## OWASP top 10

### SQL Injection

* They target databases.
* Untrusted user input is interpreted by the server and executed.
* Data might be stolen, modified or deleted.
* **How to prevent it?**
  + Validate data input/Sanitize the input.
  + Testing using pen tester.
  + Escaping inputs – ‘- - '
  + Ex- SQL Server injection, LDAP, OS commands, XML Parsers, Expression languages, XPath, NoSQL, ORM queries.

### Broken Authentication and Session Management

* **How to prevent it?**
  + Multi-factor authentication
  + Maximum password attempts
  + Authentication lockout

### Cross site scripting (XSS)

* Cross site scripting is a client-side attack and targets users’ browsers, mainly targets using JavaScript malicious code.
* Less dangerous than SQL injection.
* They can hijack cookies, webcams, microphones and get geolocation.
* Hackers can abuse the comment function by injecting JavaScript.
* <https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html>
* **How to prevent it?**
  + User inputs should be filtered.
  + JavaScript code should be clean and secure.

### Broken Access control

* Access control enforces policy such that users cannot not act outside of their intended permissions.
* Elevation of privilege – guest has admin access.
* Vertical elevation of privilege -
* Horizontal elevation of privilege – users get access to different teams.
* This will lead to unauthorized information disclosure, modification or destruction of all data or a business function outside the user’s limit.
* **How to prevent it?**
  + Deny by default – users start with minimum privilege.
  + Enable Role based access control (RBAC) rather than allowing access to each user.
  + Disable web server directory listing and ensure file metadata and backup files are not present within web roots.
  + Constant testing and auditing of access control by pentesters.

### Security Misconfiguration

* Default credentials will be very dangerous.
* **How to prevent it?**
  + Client-side error reporting should be turned off.
  + Use of HTTPS.
  + Access to production should be restricted.

### Sensitive data exposure

* Encryption at rest
* Encryption at transit
* **How to prevent it?**

### Insufficient attack protection

* Application which does not recognize any attack repeatedly.

### Cross site request forgery (CSRF)

* **How to prevent it?**

### Using components from known vulnerabilities

### Unprotected APIs

### XML External Entities (XXE)

* **How to prevent it?**

### Insecure deserialization

* **How to prevent it?**

### Insufficient logging and monitoring

* This happens when warning and error logs generate no or unclear logs.
* Example – Indian airlines data breach, Target attacked for consumer data.
* **How to prevent it?**
  + All logs should be kept back to a certain period.

### Cryptographic Failure

* This can happen if you are using weak encryption or when cryptographic encryptions fail to protect us.
* Network analyzer - Wireshark
* Session cookies - A session cookie is a cookie that tracks username and password of the user during a particular session.
* **How to prevent it?**
  + Use secure protocols like HTTPS, SSH, SFTP, FTPS and avoid insecure protocols like Telnet, FTP.
  + Use strong encryption – AES (Advanced Encryption Standard).

### Insecure design

* It focuses on risks related to design and architectural flaws, threat modelling and reference architecture.
* Threat modelling –
* Data classification

1. Public
2. Private
3. Restricted
4. Confidential

* Software development cycle security
* **How to prevent it?**
  + Principle of least privilege.
  + Validation of input.
  + Segregation of tenants.
  + Encryption.
  + Fail securely.
  + Observe

### Software and Data Integrity failures

* It focuses on making assumptions related to software updates, critical data and CI/CD pipelines without verifying integrity.
* Trusting updates from any software.
* **How to prevent it?**

### Server-side Request Forgery (SSRF)

* It happens to a web application when it fetches a remote resource without validating the user supplied URL.
* It can be used to launch DDOS attacks.
* **How to prevent it?**
  + Deny by default.
  + Segment RDP to separate networks.

### Vulnerable and outdated components

* Attacks – Log4j, SolarWinds
* **How to prevent it?**

### Identification and Authentication Failure

* **How to prevent it?**

## Tools and Techniques to implement security

### Product risk profiles

### Security requirements

### Threat modelling

### Architecture assessment

### Encryption at rest and in transit

* Data at rest
* Data in transit
* Data in use

### Security logging and monitoring

### Penetration testing

* Also known as ethical hacking.
* Type of testing

1. White box testing
2. Black box testing
3. Gray box testing

### Source code management (SCM)

* Enables code reviews.

### CI/CD tools

### Secret scanning tools

### Software composition analysis (SCA)

### Static application security testing (SAST)

* Also known as White box testing
* Examples - SonarQube

### Dynamic application security testing (DAST)

* Also known as Black box testing
* Examples – Burp Suite

### Interactive application security testing (IAST)

* Also known as Glass box testing