**1.Cross-Site Scripting (XSS)**

**Threats and Vulnerabilities:**

* **Threat:** Attackers inject malicious scripts into web pages.
* **Vulnerabilities:** Poor input sanitization, lack of Content Security Policies (CSP).
* **Affected Security Pillars:**
  + **Confidentiality:** Attackers can steal sensitive data.
  + **Integrity:** Malicious scripts can alter content.
  + **Availability:** XSS may cause DoS by exploiting browser vulnerabilities.

**Risks and Impact:**

* **Legal:** Exposure of data may lead to regulatory penalties.
* **Financial:** Loss from fraud and fines.
* **Reputational:** Users may abandon services due to safety concerns.

**Remediation Measures:**

* Use **CSP** to restrict script sources.
* **Sanitize** and **escape** user inputs.
* Use frameworks like **React** or **Angular** to prevent XSS.
* Set **HTTP-only** and **secure** flags on cookies.

**Risk Mitigation Strategies:**

* Regular **security scans** for XSS vulnerabilities.
* Educate users on **phishing** risks.
* **Limit script permissions** (avoid inline JS).

**2.DDoS (Distributed Denial of Service) Attack**

**Threats and Vulnerabilities:**

* **Threat:** Attackers flood servers, networks, or applications with traffic, causing unavailability.
* **Vulnerabilities:** Insufficient network capacity, lack of DDoS protection, weak infrastructure, and poor traffic filtering.
* **Affected Security Pillars:**
  + **Confidentiality:** No direct impact, but delays in access to data.
  + **Integrity:** No direct impact, but service disruption may cause operational inconsistencies.
  + **Availability:** Primary impact, causing unavailability of services.

**Risks and Impact:**

* **Legal:** Potential regulatory penalties for service downtime in critical sectors.
* **Financial:** Loss of revenue, increased costs, and potential compensation claims.
* **Reputational:** Loss of customer trust and long-term brand damage.

**Remediation Measures:**

* Use DDoS protection services (Cloudflare, AWS Shield).
* Increase bandwidth and implement rate limiting and traffic filtering.
* Set up redundant infrastructure and deploy WAF.

**Risk Mitigation Strategies:**

* Use CDNs to distribute traffic.
* Perform regular stress tests.
* Collaborate with ISPs and prepare an incident response plan.
* Monitor traffic patterns for early detection.

**3.Man-in-the-Middle (MITM) Attacks**

**Threats, Vulnerabilities, and Affected Security Pillars**

* **Threats**: Eavesdropping, data tampering, session hijacking, impersonation.
* **Vulnerabilities**: Lack of encryption (HTTP instead of HTTPS), weak authentication, SSL/TLS weaknesses.
* **Affected Pillars**:
  + **Confidentiality**: Exposure of sensitive data.
  + **Integrity**: Data manipulation.
  + **Availability**: Potential service disruption.

**Risks and Impact**

* **Exposure of Data**: Intercepted sensitive information (e.g., passwords, financial data).
* **Loss of Trust**: Damage to brand reputation.
* **Legal Risks**: Non-compliance with data protection laws (e.g., GDPR).
* **Financial Loss**: Potential fraud or unauthorized transactions.
* **Reputation**: Negative media coverage and customer churn.

**Remediation Measures**

* **Use HTTPS/TLS**: Encrypt all communications.
* **Implement MFA**: Add extra authentication layers.
* **Enable HSTS**: Prevent HTTP downgrade attacks.
* **Use DNSSEC**: Secure DNS to avoid spoofing.
* **Secure Session Management**: Protect session cookies with secure flags.
* **Regular Security Audits**: Identify vulnerabilities proactively.

**Risk Mitigation Strategies**

* **User Education**: Teach users to identify insecure connections.
* **VPNs**: Encourage usage on public networks.
* **End-to-End Encryption**: Further protect sensitive data.
* **Incident Response**: Prepare for quick detection and response to attacks.

**4.SQL Injection (SQLi) Attack**

**Threats and Vulnerabilities:**

* **Threat:** Attackers inject malicious SQL to access or modify databases.
* **Vulnerabilities:** Insufficient input validation, lack of parameterized queries, improper sanitization.
* **Affected Security Pillars:**
  + **Confidentiality:** Sensitive data exposure.
  + **Integrity:** Data corruption or deletion.
  + **Availability:** Database performance degradation.

**Risks and Impact:**

* **Legal:** Potential fines for data breaches.
* **Financial:** Losses from fraud, recovery, and legal actions.
* **Reputational:** Loss of customer trust.

**Remediation Measures:**

* Use **parameterized queries** and **input sanitization**.
* Limit **database permissions**.
* Deploy **WAF** and **stored procedures**.

**Risk Mitigation Strategies:**

* Regular **security audits** and **pen testing**.
* Educate developers on **secure coding**.
* Implement **intrusion detection** and **least privilege access**.

**5.Clickjacking Attack**

**Threats and Vulnerabilities:**

* **Threat:** Users are tricked into clicking on hidden, malicious elements via transparent iframes.
* **Vulnerabilities:** Lack of **X-Frame-Options** or **CSP** headers, improper iframe handling.
* **Affected Security Pillars:**
  + **Confidentiality:** Exposure of sensitive data via unintended actions.
  + **Integrity:** Manipulation of user actions or data.
  + **Availability:** Loss of trust, potentially reducing service usage.

**Risks and Impact:**

* **Legal:** Liability for data breaches or misuse under privacy laws.
* **Financial:** Losses from fraud or legal actions.
* **Reputational:** Damaged user trust.

**Remediation Measures:**

* Use **X-Frame-Options** or **CSP** to block iframe embedding.
* Validate user actions and prevent unintended clicks.
* Implement **UI redress protection** for sensitive actions.

**Risk Mitigation Strategies:**

* Conduct regular **security audits** for clickjacking risks.
* **Educate users** on suspicious behaviour.
* **Monitor user actions** for unusual behaviour.