**A Mobile Hotel Management System for Cyber Security**

**Overview :-**

Designing a mobile hotel management system with a strong focus on cybersecurity is essential to protect guest data, ensure secure communications, and prevent unauthorized access. Here are key features and security considerations for such a system:

* Authentication and Authorization: Implement robust authentication mechanisms, such as username/password, biometric authentication, or two-factor authentication (2FA).
* Secure Communication: Use encryption (e.g., SSL/TLS) to secure communication between the mobile app and backend servers.
* Data Encryption: Encrypt all sensitive data stored on mobile devices and backend servers to prevent unauthorized access.
* Access Controls: Implement role-based access controls (RBAC) to restrict user access based on their roles within the hotel.
* Secure APIs: Protect APIs with authentication, authorization, and proper validation mechanisms.
* Secure Payment Processing: Comply with industry standards (e.g., PCI DSS) and use tokenization or encryption for payment data.
* Device Security: Use mobile device management (MDM) to enforce security policies and remotely wipe data on staff devices.
* Secure Data Storage: Use secure databases and apply encryption to protect data at rest.
* Regular Security Audits and Penetration Testing: Identify and address vulnerabilities proactively.
* Intrusion Detection and Prevention: Implement systems to monitor and respond to suspicious activities.
* Security Training: Provide cybersecurity training to hotel staff.
* Incident Response Plan: Develop a comprehensive plan to respond to security incidents.
* Privacy Compliance: Ensure compliance with data privacy regulations and obtain proper guest consent.
* Secure Updates: Verify software updates from official sources to prevent tampered versions.
* Third-Party Service Review: Regularly review security practices of third-party services.
* Secure Guest Communication: Enable secure channels to prevent phishing and social engineering attacks.

By incorporating these security measures, a mobile hotel management system can protect guest privacy, prevent data breaches, and maintain the hotel's integrity and trust. Continuous monitoring and improvement are essential to adapt to emerging threats and maintain a secure environment for both guests and staff.

**List of teammates–**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no** | **name** | **collage** | **contact** |
| **1** | **Dr. P. Umaeswari** | **R. M. K. Engineering College** | **9884613147**  **pui.csbs@rmkec.ac.in** |

**List of Vulnerability Table ➖**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Vulnerability Name** | **CWE - No** |
| **1** | Injection | **CWE-564: SQL Injection: Hibernate** |
| **2** | [**Broken Access Control**](https://owasp.org/Top10/A01_2021-Broken_Access_Control/) | **CWE-284 : Improper Access Control** |
| **3** | [**Cryptographic Failures**](https://owasp.org/Top10/A02_2021-Cryptographic_Failures/) | **CWE -310:Cryptographic Issues** |
| **4** | [**Insecure Design**](https://owasp.org/Top10/A04_2021-Insecure_Design/) | **CWE-657: Violation of Secure Design Principles** |
| **5** | [**Security Misconfiguration**](https://owasp.org/Top10/A05_2021-Security_Misconfiguration/) | **CWE -1349 - Security Misconfiguration** |
| **6** | [**Vulnerable and Outdated Components**](https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/) | **CWE-657: Violation of Secure Design Principles** |
| **7** | [**Identification and Authentication Failures**](https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/) | **CWE -1353: Identification and Authentication Failures** |
| **8** | [**Software and Data Integrity Failures**](https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures/) | CWE - 1354 : Software and Data Integrity Failures |
| **9** | [**Security Logging and Monitoring Failures**](https://owasp.org/Top10/A09_2021-Security_Logging_and_Monitoring_Failures/) | **CWE-778: Insufficient Logging** |
| **10** | [**Server-Side Request Forgery**](https://owasp.org/Top10/A10_2021-Server-Side_Request_Forgery_%28SSRF%29/) |  |

**REPORT:-**

**Vulnerability Name:- Injection**

**CWE : - CWE-564: SQL Injection: Hibernate**

**OWASP Category:-** [**A03:2021-Injection**](https://owasp.org/Top10/A03_2021-Injection/)

**Description:- Hibernate is an SQL injection vulnerability found in applications utilizing the Hibernate Object-Relational Mapping (ORM) framework. It falls under the OWASP Injection category.**

**Business Impact**::- This weakness arises when developers improperly construct SQL queries by directly concatenating user input, without adequate validation or parameterization. Malicious actors can exploit this flaw to inject harmful SQL code into input fields, potentially compromising data integrity, exposing sensitive information, causing application disruptions, and leading to financial losses. To mitigate this risk, developers should employ parameterized queries, input validation, and least privilege principles, along with regular security testing to identify and fix vulnerabilities, ensuring the security and trustworthiness of their applications.

**Vulnerability Name:- Broken Access Control**

# **CWE : - CWE-284 : Improper Access Control**

**OWASP Category:-** **[A01 : 2021 Broken Access Control](https://owasp.org/Top10/A01_2021-Broken_Access_Control/)**

**Description:-** Broken Access Control is a security vulnerability that occurs when an application fails to enforce proper access controls, allowing unauthorized users to access sensitive functionalities, data, or resources. This vulnerability can arise due to insufficient authorization checks or improper enforcement of user privileges within the app.

**Business Impact**::- When Broken Access Control exists, attackers may exploit it to gain unauthorized access to restricted areas, perform actions beyond their intended permissions, modify or delete critical data, and compromise the confidentiality, integrity, and availability of the application's resources. This vulnerability can lead to data breaches, unauthorized account access, privilege escalation, and other serious security incidents.

**Vulnerability Name:-** [**Cryptographic Failures**](https://owasp.org/Top10/A02_2021-Cryptographic_Failures/)

**CWE : - CWE-310: Cryptographic Issues**

**OWASP Category:-** [**A02:2021-**[**Cryptographic Failures**](https://owasp.org/Top10/A02_2021-Cryptographic_Failures/)](https://owasp.org/Top10/A03_2021-Injection/)

**Description:-** [**Cryptographic Failures**](https://owasp.org/Top10/A02_2021-Cryptographic_Failures/) shifts up one position to #2, previously known as Sensitive Data Exposure, which was broad symptom rather than a root cause. The renewed focus here is on failures related to cryptography which often leads to sensitive data exposure or system compromise.

**Business Impact**::- To mitigate CWE-310 cryptographic issues, developers must use strong and up-to-date encryption algorithms, implement proper key management practices, and conduct regular security assessments to identify and address cryptographic vulnerabilities. Security best practices and standards should be followed to ensure that cryptographic functions are properly implemented, and applications are adequately protected against cryptographic attacks.

**Vulnerability Name:- Insecure Design**

**CWE : - CWE-657: Violation of Secure Design Principles**

**OWASP Category:-** [**A03:2021- Insecure Design**](https://owasp.org/Top10/A03_2021-Injection/)

**Description:-** [**Insecure Design**](https://owasp.org/Top10/A04_2021-Insecure_Design/) is a new category for 2021, with a focus on risks related to design flaws. If we genuinely want to “move left” as an industry, it calls for more use of threat modeling, secure design patterns and principles, and reference architectures.

**Business Impact**::- The business impact of A04:2021-Insecure Design can be severe and far-reaching. Insecure design decisions in web applications can lead to data breaches, financial losses, and reputational damage. Vulnerabilities like inadequate authentication, improper access controls, and insecure data storage can result in unauthorized access to sensitive information, compromising user privacy and leading to legal consequences. Security incidents caused by insecure design may disrupt services, erode customer trust, and lead to customer churn. Additionally, exposure of intellectual property through weak design decisions can result in theft and loss of competitive advantage. To mitigate these risks, organizations must prioritize secure design principles, conduct regular security assessments, and provide training to developers to build more robust and secure applications.

**5.Vulnerability Name:-** [**Security Misconfiguration**](https://owasp.org/Top10/A05_2021-Security_Misconfiguration/)

**CWE : - CWE -1349 - Security Misconfiguration**

**OWASP Category:-** [**A05:2021-**[**Security Misconfiguration**](https://owasp.org/Top10/A05_2021-Security_Misconfiguration/)](https://owasp.org/Top10/A03_2021-Injection/)

**Description:-** [**Security Misconfiguration**](https://owasp.org/Top10/A05_2021-Security_Misconfiguration/) moves up from #6 in the previous edition; 90% of applications were tested for some form of misconfiguration. With more shifts into highly configurable software, it’s not surprising to see this category move up. The former category for XML External Entities (XXE) is now part of this category.

**Business Impact**::- Security Misconfiguration is a vulnerability that arises due to improper or insecure configuration of software and systems. When security settings, permissions, or defaults are not appropriately configured, attackers can exploit these weaknesses to gain unauthorized access, execute malicious code, or compromise sensitive data. The business impact of security misconfigurations can be severe, leading to data breaches, financial losses, reputational damage, and legal consequences. Attackers may exploit misconfigurations to gain unauthorized access to critical resources, leading to data theft, service disruptions, and potential compliance violations. Insecure configurations can also expose proprietary information, intellectual property, and trade secrets, resulting in a loss of competitive advantage. To mitigate this risk, organizations must implement secure configuration practices, conduct regular security audits, and follow industry best practices to ensure proper and secure configuration of their software and systems.

**6. Vulnerability Name:-** [**Vulnerable and Outdated Components**](https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/)

**CWE : - CWE-657: Violation of Secure Design Principles**

**OWASP Category:-** [**A06:2021-**[**Vulnerable and Outdated Components**](https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/)](https://owasp.org/Top10/A03_2021-Injection/)

**Description:-** [**Vulnerable and Outdated Components**](https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/) was previously titled Using Components with Known Vulnerabilities and is #2 in the Top 10 community survey, but also had enough data to make the Top 10 via data analysis. This category moves up from #9 in 2017 and is a known issue that we struggle to test and assess risk.

**Business Impact**::- CWE-657: Violation of Secure Design Principles is a vulnerability that occurs when software is developed without adhering to secure design principles, leading to the creation of inherently insecure applications. By disregarding crucial security considerations during the design phase, applications become susceptible to various exploits and attacks, causing severe business impacts. The business impact of violating secure design principles includes data breaches, financial losses due to theft or fraud, reputational damage, and legal consequences. Attackers can exploit vulnerabilities arising from insecure design to gain unauthorized access, manipulate data, disrupt services, and compromise user privacy. Additionally, the exposure of sensitive information and intellectual property due to insecure design can erode customer trust, damage the organization's reputation, and lead to a loss of competitive advantage. To address this risk, organizations must prioritize secure design principles, conduct thorough security reviews, and integrate security from the early stages of software development to build robust and resilient applications.

**7. Vulnerability Name:- Identification and Authentication Failures**

**CWE : CWE -1353: Identification and Authentication Failures**

**OWASP Category:-** **[A07:2021- Identification and Authentication Failures](https://owasp.org/Top10/A03_2021-Injection/)**

**Description:-** [**Identification and Authentication Failures**](https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/) was previously Broken Authentication and is sliding down from the second position, and now includes CWEs that are more related to identification failures. This category is still an integral part of the Top 10, but the increased availability of standardized frameworks seems to be helping.

**Business Impact**::- Identification and Authentication Failures is a category in the OWASP Top 10 list that highlights vulnerabilities related to improper identification and authentication mechanisms in web applications. The business impact of these failures can be severe and wide-ranging. Insufficient or weak identification and authentication can lead to unauthorized access to sensitive information, user accounts, and critical functionalities. Attackers may exploit these vulnerabilities to perform unauthorized actions, compromise user privacy, commit fraud, and steal sensitive data. The consequences include data breaches, financial losses due to fraudulent activities, reputational damage, and potential legal and regulatory implications. Additionally, identification and authentication failures can erode customer trust and loyalty, leading to a loss of business opportunities and competitive disadvantage. To mitigate these risks, organizations must implement strong identification and authentication practices, including multi-factor authentication, to ensure the security and integrity of their web applications and protect user data effectively.

## **8. Vulnerability Name:-** Software and Data Integrity Failures

**CWE : -** CWE - 1354 : Software and Data Integrity Failures

## **OWASP Category:-** **[A08:2021-](https://owasp.org/Top10/A03_2021-Injection/)** [Software and Data Integrity Failures](https://owasp.org/Top10/A03_2021-Injection/)

**Description:-** [**Software and Data Integrity Failures**](https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures/) is a new category for 2021, focusing on making assumptions related to software updates, critical data, and CI/CD pipelines without verifying integrity. One of the highest weighted impacts from Common Vulnerability and Exposures/Common Vulnerability Scoring System (CVE/CVSS) data mapped to the 10 CWEs in this category. Insecure Deserialization from 2017 is now a part of this larger category.

**Business Impact**::- Software and Data Integrity Failures represent vulnerabilities that arise when software and data integrity are compromised, leading to potential security breaches and significant business impacts. Attacks on software and data integrity can result in unauthorized modification, deletion, or corruption of critical data, applications, or system files. The business impact of these failures includes data loss, service disruptions, financial losses due to system downtime, and reputational damage. In cases where the integrity of software code is compromised, attackers may inject malicious code or execute unauthorized commands, leading to unauthorized access, data breaches, and potential compliance violations. The erosion of customer trust, loss of business opportunities, and legal repercussions are additional consequences of software and data integrity failures. Organizations must implement robust security measures, such as integrity checks, code signing, and access controls, to ensure the integrity and authenticity of their software and data, mitigating the risk of these vulnerabilities and protecting their assets and reputation.

**9. Vulnerability Name:-** [**Security Logging and Monitoring Failures**](https://owasp.org/Top10/A09_2021-Security_Logging_and_Monitoring_Failures/) 

**CWE : - CWE-778: Insufficient Logging**

**OWASP Category:-** [**A09:2021-**[**Security Logging and Monitoring Failures**](https://owasp.org/Top10/A09_2021-Security_Logging_and_Monitoring_Failures/)](https://owasp.org/Top10/A03_2021-Injection/)

**Description:-** [**Security Logging and Monitoring Failures**](https://owasp.org/Top10/A09_2021-Security_Logging_and_Monitoring_Failures/) was previously Insufficient Logging & Monitoring and is added from the industry survey (#3), moving up from #10 previously. This category is expanded to include more types of failures, is challenging to test for, and isn’t well represented in the CVE/CVSS data. However, failures in this category can directly impact visibility, incident alerting, and forensics.

**Business Impact**::- Insufficient Logging refers to the vulnerability where an application lacks adequate and comprehensive logging of security-related events. Insufficient logging and monitoring can have a significant business impact, making it difficult to detect and respond to security incidents effectively. Without proper logging, organizations may fail to identify and investigate suspicious activities, leaving them unaware of ongoing attacks or breaches. This can lead to delayed incident response, prolonged unauthorized access, data exfiltration, and further exploitation of vulnerabilities. The consequences include data breaches, financial losses due to theft or fraud, damage to customer trust, reputational harm, and potential legal and regulatory repercussions. By implementing robust logging and monitoring practices, organizations can improve their ability to detect and respond to security incidents promptly, enhancing the overall security posture and mitigating the potential business impact of insufficient logging vulnerabilities..

**10. Vulnerability Name:- Server-Side Request Forgery**

**CWE : - CWE-918: Server-Side Request Forgery**

**OWASP Category:-** [**A10:2021-Server-Side Request Forgery**](https://owasp.org/Top10/A03_2021-Injection/)

**Description:-** [**Server-Side Request Forgery**](https://owasp.org/Top10/A10_2021-Server-Side_Request_Forgery_%28SSRF%29/) is added from the Top 10 community survey (#1). The data shows a relatively low incidence rate with above average testing coverage, along with above-average ratings for Exploit and Impact potential. This category represents the scenario where the security community members are telling us this is important, even though it’s not illustrated in the data at this time.

**Business Impact**::- Server-Side Request Forgery (SSRF) is a vulnerability that allows attackers to manipulate an application into making unauthorized requests to internal or external resources on behalf of the server. The business impact of SSRF can be severe, as it enables attackers to access sensitive information, compromise internal systems, and perform malicious actions. Attackers can use SSRF to bypass firewalls and security measures, leading to data breaches, unauthorized access to backend services, and potential exposure of sensitive data. This vulnerability can also facilitate attacks on other vulnerable internal systems, resulting in service disruptions, financial losses, and reputational damage. Additionally, SSRF can be used to launch attacks on third-party services, causing compliance violations, contractual breaches, and legal consequences. Organizations must implement input validation, enforce strict access controls, and conduct thorough security assessments to detect and address SSRF vulnerabilities, mitigating the risk of exploitation and the potential business impact.