**Penetration Testing Report**

**Test Information**

Test Date: 31.1.2024.

Penetration Testing Team: 19

Methodology: The testing team employed a combination of manual and automated penetration testing techniques, utilizing tools such as OWASP to identify vulnerabilities.

**Testing Results**

1. XSS Vulnerability

Description: While regex expressions are employed for input validation on all input fields, the application is still susceptible to XSS attacks due to the absence of complete protection against various attack tactics. Specifically, dynamic data rendering on certain pages lacks sufficient user input validation.

Recommendations:

* Enhance existing input validation mechanisms to cover all possible XSS attack vectors.
* Implement input filtering mechanisms to sanitize user inputs effectively.
* For dynamic data rendering, utilize Angular's DomSanitizer to mitigate XSS risks associated with dynamic content.

2. SQL Injection Vulnerability

Description: A potential SQL injection vulnerability has been observed in specific parts of the application.

Recommendations: Utilize parameterized queries and appropriate protection functions against SQL injection.

3. Library Vulnerabilities

High-Risk Vulnerabilities: [commons-jxpath-1.3](file:///D:\faks\MAS\SEP\SEP\PSPBackend\target\dependency-check-report.html#l17_c22d7d0f0f40eb7059a23cfa61773a416768b137), [guava-19.0](file:///D:\faks\MAS\SEP\SEP\PSPBackend\target\dependency-check-report.html#l24_6ce200f6b23222af3d8abb6b6459e6c44f4bb0e9), [jackson-core-2.15.3](file:///D:\faks\MAS\SEP\SEP\PSPBackend\target\dependency-check-report.html#l34_60d600567c1862840397bf9ff5a92398edc5797b), [jettison-1.4.0](file:///D:\faks\MAS\SEP\SEP\PSPBackend\target\dependency-check-report.html#l48_8f0e7bb69242e9ed5bfdf7384b37c1095b0974fc), [logback-core-1.4.11](file:///D:\faks\MAS\SEP\SEP\PSPBackend\target\dependency-check-report.html#l52_2f9f280219a9922a74200eaf7138c4c17fb87c0f), [reactor-netty-core](file:///D:\faks\MAS\SEP\SEP\PSPBackend\target\dependency-check-report.html#l67_378dc5a375e6440099e837b22cf4b01341cbe4ea), [snakeyaml-1.33](file:///D:\faks\MAS\SEP\SEP\PSPBackend\target\dependency-check-report.html#l70_2cd0a87ff7df953f810c344bdf2fe3340b954c69), [spring-boot-3.1.5](file:///D:\faks\MAS\SEP\SEP\PSPBackend\target\dependency-check-report.html#l71_c188015a5a79f5df65e876dcfdef16148c45fe2c), [spring-cloud-starter](file:///D:\faks\MAS\SEP\SEP\PSPBackend\target\dependency-check-report.html#l74_7a90112c371183668b51e657a1f9113ed38f126f), [spring-security-rsa-1.0.12.RELEASE](file:///D:\faks\MAS\SEP\SEP\PSPBackend\target\dependency-check-report.html#l79_2c6aca9e7eee11aee196a29eeea35d7aa8692398)

Medium-Risk Vulnerabilities: [jackson-databind-2.15.3](file:///D:\faks\MAS\SEP\SEP\PSPBackend\target\dependency-check-report.html#l35_a734bc2c47a9453c4efa772461a3aeb273c010d9), [jakarta.activation-api-2.1.2](file:///D:\faks\MAS\SEP\SEP\PSPBackend\target\dependency-check-report.html#l37_640c0d5aff45dbff1e1a1bc09673ff3a02b1ba12), [httpclient-4.5.3](file:///D:\faks\MAS\SEP\SEP\PSPBackend\target\dependency-check-report.html#l27_d1577ae15f01ef5438c5afc62162457c00a34713), [commons-lang-2.6](file:///D:\faks\MAS\SEP\SEP\PSPBackend\target\dependency-check-report.html#l18_0ce1edb914c94ebc388f086c6827e8bdeec71ac2)

Low-Risk Vulnerabilities: /

Remediation Steps

* Utilize OWASP guidelines for registering identified vulnerabilities.
* Use plugins to regularly update and patch libraries used in the application.
* Remove unnecessary and vulnerable parts of libraries that are not essential for the application.

**PCI DSS Compliance Report**

**Overview**

This report provides an overview of the PCI DSS compliance measures implemented in the system which is Payment Service Provider.

**Compliance Measures**

1. Non-Usage of Insecure Protocols

Status: Implemented

Description: The system does not use protocols known to be insecure, ensuring compliance with PCI DSS requirements.

2. Protection of Stored Account Data

Status: Implemented

Description: Sensitive data stored in the database is encrypted to comply with PCI DSS requirements.

3. Information Security Policy for Personnel

Status: Implemented

Description: A comprehensive information security policy has been established, covering all personnel. This policy encompasses various security aspects, including access controls, data protection, network security, and secure coding practices. Modern technologies, such as Spring, Angular, and PostgreSQL, are strategically employed to enforce and monitor adherence to this policy.

Utilization of modern technologies:

* **Spring (Backend):** Implements robust access controls, secure data handling, and enforces coding practices, ensuring overall data protection.
* **Angular (Frontend):** Enables secure user interfaces, supporting input validation and secure communication for a protected user experience.
* **PostgreSQL (Database):** Utilizes encryption mechanisms for data confidentiality and leverages reliability and security features, aligning with PCI DSS requirements.