OWASP Top 10 List

Team 23

1. **Broken Access Control**

We utilize Keycloak as our primary system for implementing role-based access control (RBAC) within our application. By leveraging Keycloak, we ensure that access to various resources and functionalities within our application is strictly governed by predefined user roles and associated permissions. This means that each user is assigned a specific role, and each role is granted access only to those resources and actions necessary for their function.

2. **Cryptographic failures**

To protect passwords from potential attacks, we employ a process of hashing passwords multiple times using strong cryptographic hashing algorithms. In addition to robust password hashing, we ensure that no data is transmitted in clear text over the network. All data transmission between clients and servers is encrypted using HTTPS. For secure key distribution, we use public key infrastructure (PKI). Public keys are embedded within digital certificates, which are issued and digitally signed by trusted Certificate Authorities (CAs). This process ensures the authenticity and integrity of the public keys, providing assurance that they have not been tampered with and are being distributed safely.

3. **Injection**

In order to prevent injection attacks and ensure the integrity of our application, we have implemented prepared statements for all database queries. Furthermore, every piece of input data that our application receives undergoes thorough validation. This validation process involves checking for expected data formats and constraints before the data is processed or used in any database operations.

4. **Insecure design**

Not supported in any capacity.

5. **Security Misconfiguration**

To enhance the security and stability of our application, we ensure that all necessary configuration settings are securely passed through configuration files, ensuring that sensitive information such as database credentials, API keys, and environment settings are protected.

We have also taken steps to minimize our application's attack surface by eliminating any unnecessary ports, services, pages, accounts, or privileges and end-users receive generic error messages that do not reveal the inner workings of the application.

6. Vulnerable and Outdated Components

Not supported in any capacity.

7. Identification and Authentication Failures

Passwords are checked against breached password list and require complexity rule defined in NIST 800-63b guideline. No accounts are generated with default credentials. Keycloak is used as an external authentication service which ensures that there is no errors in security configuration.

8. Software and Data Integrity Failures

Not supported in any capacity.

9. Security Logging and Monitoring Failures

Not supported in any capacity.

10. Server-Side Request Forgery

Not supported in any capacity.