Game Market

Security Report

Text

Description automatically generated with medium confidence

24/03/2022 Eindhoven

Version: 2.0

Members:

Lars Kluijtmans: 4220269

Table of Contents

[Version 3](#_Toc106116752)

[Overview 4](#_Toc106116753)

[Security report 5](#_Toc106116754)

[1 Broken access control 5](#_Toc106116755)

[2 Cryptographic failure 5](#_Toc106116756)

[3 Injection 5](#_Toc106116757)

[4 Insecure Design 5](#_Toc106116758)

[5 Security Misconfiguration 5](#_Toc106116759)

[6 Vulnerable and outdated components 6](#_Toc106116760)

[7 Identification and authorization failures 6](#_Toc106116761)

[8 Software and data integrity failure 6](#_Toc106116762)

[9 Security logging and monitoring failures 6](#_Toc106116763)

[10 Server side request forgery 6](#_Toc106116764)

# Version

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Autor | Changes | Status |
| 0.1 | 08/03/2022 | Lars Kluijtmans | First edition. | Not finished |
| 1.0 | 14/06/2022 | Lars Kluijtmans | Moved to separate document |  |

# Overview

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Likelyhoob | Impact | Risk | Actions possible | Planned |
| A1: Broken access control | Moderate | Severe | Severe | Fixed | Yes |
| A2: Cryptographic failure | Moderate | Moderate | Moderate | All of a users private data needs to be encoded. | No |
| A3: Injection | Moderate | Severe | Severe | Need to be fixed | No |
| A4: Insecure Design | Low | Severe | Severe | Fixed | Yes |
| A5: Security Misconfiguration | Moderate | Low | Low | Fixed | Yes |
| A6: Vulnerable and outdated components | Low | Low | Low | Fixed | No, risk accepted |
| A7: Identification and authorization failures | Moderate | Severe | Severe | Ensure users use strong passwords | No |
| A8: Software and data integrity failure | Moderate | Moderate | Moderate | Fixed | No |
| A9: Security logging and monitoring failures | Severe | Moderate | Moderate | Needs to be implemented | No |
| A10: Server side request forgery | Low | Low | Low | Fixed | No, risk accepted |

# Security report

How does this application deal with the OWASP top 10 security risks(2021):

## 1 Broken access control

To prevent this all recourses ,except some public once, are not accessible by default. Cross-origin resource sharing is used so only requests from some websites are permitted. Ownership of records is enforced so only users that own a record can update and delete them.

## 2 Cryptographic failure

All data except emails, phone number and password do not fall under the Privacy law. Sadly of there three that do only one (Password) is saved in the database as encrypted. No sensitive data is stored unnecessarily, the only sensitive data that is saved are emails and phone numbers for the purpose of contacting users when needed and passwords for security reasons.

All keys and algorithms are up to the standard. Security controls are in place for data as per it’s classification so for example: only admins and the users them selves can see their information.

There is a vulnerability in the data transfer as data in transit is not encrypted.

## 3 Injection

Great venerability for this application because all connections with the database go through JPA and JPA does not prevent injections there is also no limits tot eh amount to data requested in one call to the database because I don’t know how to use limit it in combination with JPA.

## 4 Insecure Design

The design is secure because there are unit tests to validate to threats(at least to a extent) and layers of the system are segregated based on the protection needed.

## 5 Security Misconfiguration

Cross-site request forgery(CSRF) is disabled.

Error messages are not overly informative. There are no unnecessary features, components, documentations and samples.

Application architecture provides effective and secure separation between components with segmentation/containerization. (at least I would have made it that way of I had more time. Did it for the group project, know how to do it.)

## 6 Vulnerable and outdated components

There are no unused dependencies. All components are new, from trusted sources and being updated regularly.

## 7 Identification and authorization failures

Passwords only have a min length there is no checking for their ’strength’. There is no credential recovery or forgot-password process. So malicious users can’t take advantage of it? Passwords are stored encrypted and the encryption can be considered ‘strong’.

## 8 Software and data integrity failure

To ensure that integration is always successful we use a CI/CD pipeline that has proper segregation, configuration.

## 9 Security logging and monitoring failures

Sadly there is no monitoring of: Logins, failed logins and high value transactions, Logs are not supervised for suspicious activities.

There is nothing to trigger alerts if there is a case of Penetration testing or dynamic application security testing (DAST).

## 10 Server side request forgery

Not relevant for this application as there are no other application that can connect to the rest API except the website.