OWASP Report

Fontys University of Applied Sciences

Individual Project(TickieSystem)

Author: Aleksandar Georgiev

Table of Contents

Description of the document

The purpose of this document is to show the security standards of OWASP that the project is currently fulfilling. OWASP is an organization that gives the standards for web security and in this document, it will be showed which of the security ricks the projects has fixed and why there are some ricks that are still not secured. The document is going to cover the 10 to cover the TOP 10 security ricks for web applications.

Security Risks

Injection

Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization. (OWASP.org)

The project is secured from the SQL injection because it uses the JPA framework which has implemented the security.

Broken Authentication

Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently. (OWASP.org)

The application uses JWT authentication which prevents session hijacking, and it is a secure authentication in general.

Sensitive data Exposure

Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser. (OWASP.org)

There is no sensitive data in the front end of for this project, Therefore, this is not applicable for the project.

XML External Entities (XEE)

Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks. (OWASP.org)

XEE is not applicable to this project.

Broken Access Control

Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc. (OWASP.org)

Authorization of the web application is not properly enforced, and the attacker can gain access to sensitive data. The authentication of the project Is design and implemented.

Security Misconfiguration

Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched/upgraded in a timely fashion. (OWASP.org)

There are no sensitive HTTP headers, no secret keys and API keys in the project, only need to configure good error messages.

Cross-Site Scripting(XSS)

XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

The application uses CORS Filter in every API end point to counter XSS.

Insure Deserialization

Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks. (OWASP.org)

The APIs DO NOT send any sensitive data to the front end. There is no Special Model for the API in place.

Using Components with Known Vulnerabilities

Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts. (OWASP.org)

The project uses only best practices and tested frameworks like React and Spring boot.

Insufficient Logging and Monitoring

Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring. (OWASP.org)

There is no logging and monitoring in place.

References

1. OWASP.org (2017). OWASP top ten security risks. https://owasp.org/www-project-top-ten/