OWASP Top 10

By Sebastian Bokkestijn

ecurity Risks:

1. Injection

Because I am using JPA, and only using JPA generated queries, I am safe from SQL Injection attacks.

1. Broken Authentication

My application is has password validation that requires a certain minimum length for passwords and uses JWT token for validation and destroys session after a period of time.

1. Sensitive Data Exposure

No sensitive data is sent in plain text, passwords are encrypted BCrypt hashing.

1. XML External Entities (XXE)

Due to the fact that there is no place to upload XML files, and SSO and SAML aren’t used either, this risk doesn’t exist in this case.

1. Broken Access Control

My application has a role configuration for the user that is signing in therefore is restricted to only what the user is able to see.

1. Security Misconfiguration

I have configured applications in almost every regard. An auditing configuration file has been created to keep track of changes made from the front end to the backend, The a WebMvc configuration class deals with the mvc patterns by giving it a select amount of requests that are allowed.

1. Cross-site Scripting (XSS)

React can automatically escape XSS therefore the application is safe

1. Insecure Deserialization

My application doesn’t implement integrity checks such as digital signatures on any serialized objects to prevent hostile object creation or data tampering.

1. Components with Known Vulnerabilities

The frameworks and dependencies are all up to date. The release of these components haven’t specified any vulnerabilities.

1. Insufficient Logging & Monitoring

The API has minimal logging, only logging sql queries executed, errors and warnings. There is no logging of potentially suspicious activity the application is vulnerable. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.