Securing OWASP Top 10 Vulnerabilities

1. Injection

- Attack wherein an attacker can execute malicious code/statements. Occures when data input validation is not performed.
- Our system uses SQL database, so it’s potentially vulnerable of SQL injection.
- To protect from SQL injection we use hibernate “prepared” statements.

2. Cross-Site Scripting (xss)

- Enables attackers to inject client-side scripts to web pages viewed by other users.
- Our system is secured by using angularjs as frontend and input data validation.

3. Broken Authentication And Session Management

- Involves all kinds of flaws that are caused by error in implementations of authentication and/or session management
- Our system is secured by requiring password strength, using https, spring security sessions and we use JWT for authentication.

4. Insecure Direct Object Reference

- Occurs when reference to an internal implementation object, such as a file or database key, is exposed to users without any other access control
- Our system is secured with authorization for nearly every api call.

5. Security Misconfiguration

- Occur due to insecure default configuration, poorly documented default configuration, or poorly documented side-effects of optional configuration
- Our system is secured by not exposing any configuration file, and we are using spring framework with important configuration options like https being set.

6. Sensitive Data Exposure

- Occurs when an application does not adequately protect sensitive information.
- Our system is secured with authorization, passwords are salt hashed in database.

7. Insufficient Attack Protection

- Protection of automated queries, brute force attacks..
- In our system, Client have 3 tries to login, otherwise he is lock out for 5 mins.

8. Cross-Site Request Forgery (CSRF)

- Attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated.
- Our system is secured with csrf token which we send with every request.

9. Using Components with Known Vulnerabilities

- Our system uses mostly latests versions of libraries, which should be secure, we trust spring security.

10. Underprotected APIs

- Our system is secured by setting authorization for nearly each api call.