OWASP Report

Zhaklin Yanakieva

Student number: 3811468

Date: 18/12/2020

**A1 Injection**

Is the Application Vulnerable?:

In my application the user-data is validated, so it makes it not resistant to attacks. Hibernate Query Language (HQL) is not used in the code and the data is retrieved by using mapping of a whole object or path variable. Hostile data is not directly used or concatenated by using SQL or command that contains both structure and hostile data in dynamic queries, commands, or stored procedures. CRUD repositories are the ORM in my application. In my configuration of CI/CD I haven’t added the static source (SAST) and dynamic application test (DAST) tools, which are used to identify newly introduced injection flaws prior to production deployment.

In conclusion, my app is not that vulnerable and cannot be broken into by attacking the queries or by modifying the ‘id’ in the browser url, for example, because it is not displayed there and is safe.

**A2 Broken Authentication**

Is the Application Vulnerable?:

My software application doesn’t implement the of the user’s identity, authentication because the authentication is done automatically after logging in. In my application there is no forget password function, so the app is still safe at some level because the attackers cannot change get into the account and change the password of the user. The password of the users in are hashed both when registering and when updating a user. The only vulnerability here is that the token of the authorization is saved into the local storage rather than in a cookie (which is stated to be the safer way). The expiration of the token is until either the person logs out, or the server is stopped, which makes the app vulnerable to attacks in this case.

**A3 Sensitive Data Exposure**

Is the Application Vulnerable?:

None of the data in the website is transmitted in clear text (by HTTP or FTP). The only for which they are used are sending a request (OK, CREATED, etc.) in the backend to outline the result. I still do not have the functionalities in this case that can be attacked. The only thing is that password of the user uses simple hashes to store everyone’s passwords. The password should be encoded once, for which I use Encoder implemented in Spring boot, and a random token is generated each time the password is created or changed.

**A4 XML External Entities**

Is the Application Vulnerable?:

This is one of the ten OWASP that is not included in my project, so I cannot say if the application is vulnerable, according to this.

**A5 Broken Access Control**

Is the Application Vulnerable?:

Access control enforces policy such that users cannot act outside of their intended permissions. In my application, when implementing the authorization, I made the user to not be able access the actions that require an ADMIN role. There is a “@PreAuthorized(“hasrole(‘ADMIN’)”)”, which states that a certain functionality is only permitted to the admin. Also, the information of the user can be changed either by the admin or the user, and the password can be changed only by the user themselves, and not in the url. In order the website to be safer, there is used “@CrossOrigin(origins = “localhost:3000")”, where I state that only this url can receive information or send requests to the backend. Accessing API with missing access controls for POST, PUT and DELETE is not applied, which means there is no vulnerability.

**A6 Security Misconfiguration**

Is the Application Vulnerable?:

In order to protect my application I use WebSecurityConfig, where I use different roles (user and admin) so as to give different access to users, depending on what they will have to see. For instance, the guest user will see only the homepage, the aggregated user will see also the online shop, posts, their cart, they will be able to add orders and finish them, whereas the admin will have only service functionalities (CRUD). Moreover, I prevent the security misconfiguration as I always authenticate all endpoints by default.

**A7 Cross-Site Scripting**

Is the Application Vulnerable?:

The frontend framework I am using in my project is React.js and according to the researches I made, it prevents from any embedded value in JSX by escaping anything that is not explicitly written in the application, whereas there is the much higher risk when it is no more about JSX, but case with more complexity. For instance, if there was a change in the business requirements, therefore, the application now needs to accept user input with embedded data, such as different styling or case sensitivity, etc. A good practice is to sanitize the scripts before rendering. However, the best option is to use a widely-used and maintained library with zero dependencies to prevent the application from further attacks.

**A8 Insecure Deserialization**

Is the Application Vulnerable?:

The only safe architectural pattern is not to accept serialized objects from untrusted sources or to use serialization mediums that only permit primitive data types. If that is not possible, I consider one of more of the following: Implementing integrity checks such as digital signatures on any serialized objects to prevent hostile object creation or data tampering; Logging deserialization exceptions and failures, such as where the incoming type is not the expected type, or the deserialization throws exceptions; Restricting or monitoring incoming and outgoing network connectivity from containers or servers that deserialize; Monitoring deserialization, alerting if a user deserializes constantly.

**A9 Using Components with Known Vulnerabilities**

Is the Application Vulnerable?:

In my application I am using libraries, components, modules, APIs, however, these sets have their own vulnerabilities. What I haven’t thought of yet is the unsafe working of these facilities when implementing them into my project. As far as I am concerned, to prevent my application I have to follow several steps, such that ensuring that the components, together with the dependencies are secure enough, then updating the platforms and the libraries to their newest versions. In my frontend app, I am using npm commands and when something is wrong I follow the steps to scan the program.

**A10 Insufficient Logging and Monitoring**

Is the Application Vulnerable?:

This is one of the ten OWASP that is not included in my project, so I cannot say if the application is vulnerable, according to this.