OWASP REPORT

“Cinema_Now”

Andreea Șindrilaru
Fontys University of Applied Sciences
S3-CB-01

Table of Contents

	Likelihood	*Impact*	*Risk*	*Actions*	*Planned*
A01:2021-Broken Access Control	*Likely*	*Severe*	*High*	*Implement access control to minimize CORS usage and store JWT Token in httpOnly Cookies.*	*N/A*
A02:2021-Cryptographic Failures	*Low*	*Low*	*Low*	*Using UUIDs.*	*fixed*
A03:2021-Injection	*Likely*	*Moderate*	*Low*	*JPA provides input sanitization*	*N/A*
A04:2021-Insecure Design	*Low*	*Moderate*	*Low*	*Write unit and integration tests to validate all critical flows.*	*Unit tests, Integration tests*
A05:2021-Security Misconfiguration	*Very likely*	*Severe*	*High*	*Error handling reveals stack traces or other overly informative error messages to users.*	*N/A*
A06:2021-Vulnerable and Outdated Components	*Very unlikely*	*Moderate*	*Low*	*Remove unused dependencies, continuously inventory versions of application*	*fixed*
A07:2021-Identification and Authentication Failures	*Likely*	*Moderate*	*High*	*Multi-factor authentication*	*Possibly switching to OAuth2*
A08:2021-Software and Data Integrity Failures	*Likely*	*Moderate*	*Moderate*	*Ensure that CI/CD pipeline has proper segregation, configuration, and access control to ensure integrity of the code.*	*N/A*

A09:2021-Security Logging and Monitoring Failures	*Very likely*	*Severe*	*High*	*Penetration testing and scans by dynamic application security testing tools.*	*N/A*
A10:2021-Server-Side Request Forgery	*Very likely*	*Severe*	*High*	*In the application layer, all client-supplied data should be sanitized and validated. The URL schema, port and destination should be enforced.*	*N/A*

A01:2021-Broken Access Control

This security risk underlines the fact that each user should have his own privileges within the application. Any kind of vulnerability can lead to modification or even destruction of all the data. Therefore, it is crucial to make sure that, for example, the “secret” key used for encrypting the JWT Token is stored in a safe place, such a database or updated every 24 hours, or store the token in a httpOnly cookie, because storing it into the local storage or session storage can have a high chance of getting stolen.

A02:2021-Cryptographic Failures

Sensitive data (e.g. passwords, reports, credit cards) should have a protection layer. As nowadays this is a big concern, sensitive data should have encrypted IDs, as this is going to make it quite difficult to guess for hackers.

A03:2021-Injection

This risk showcases the importance of taking care of the user input. Generally speaking, user input is crucial, because if it is not sanitized, it can lead to the user controlling the response generated by the application, or even control the server by injection SQL queries, HTML or JavaScript code in forms, for example.

A04:2021-Insecure Design

When developing software, the design part also plays an important role, as any minor design error can lead to potential data loss. Therefore, I think that this risk presents the importance of regularly testing the application so that there is no “backdoor” for attackers. For example, we need to test if the data the user should see is the one wanted. (e.g. Address, Bank Account, Age)

A05:2021-Security Misconfiguration

One important aspect of this risk would be error handling. Sending custom errors to the user, instead of the full error, might lower the chance for this risk to occur, especially in case the user is an attacker who can later use the error to find other vulnerabilities within the application.

A06:2021-Vulnerable and Outdated Components

As technology develops, so should the software product. Therefore, it is crucial to make sure that each component which is used in the software is not outdated or vulnerable.

A07:2021-Identification and Authentication Failures

Weak passwords that are easy to decrypt or tokens that are not refreshed might lead to data leaks. Confirmation of the user's identity is important to protect the application against authentication-related attacks. A strong encryption algorithm for the passwords has to be used to strengthen each user's account. However, the token, used for authenticating and authorizing the user, can be bypassed if it is stored in the local storage of the browser.

A08:2021-Software and Data Integrity Failures

CI/CD plays a crucial role in software development. The pipeline needs to be secure as, if it is not, attackers can comprise the application by introducing malicious code. Integrity checks should be added for each update of the application.

A09:2021-Security Logging and Monitoring Failures

To avoid data breaches, logging and monitoring needs to be done to see the behavior of the application with different operations. (login, transactions, suspicious activity etc.).

A10:2021-Server-Side Request Forgery

Making unauthorized HTTP request (e.g. posting data that may break the security architecture) is a very common attack that applications have to deal with. Using JWT can be used to avoid such attacks, however, due to the fact that it can be stored in the local storage of the user, this can be easily altered.