withdrive

withdrive

OWASP report

OWASP top 10 report

	Likelihood	Impact	Risk	Actions	Planned
A01:2021-Broken Access Control	Likely	Severe	High	Use of access control instead of CORS, due to the limitations in the technology.	N/A
A02:2021-Cryptographic Failures	Low	Low	Low	Usage of proper key management. (UUIDs). Can be exploited with the 'Sandwich attack'.	N/A
A03:2021-Injection	Low	Moderate	Very Low	JPA does not allow for SQL injection as it sanitises queries.	N/A
A04:2021-Insecure Design	Low	Moderate	Low	Write tests that will check for errors	Unit tests, Integration tests
A05:2021-Security Misconfiguration	Very likely	Severe	Low	Using Security headers to ensure crucial requests are made by users with correct permissions. Could use OAuth.	N/A
A06:2021-Vulnerable and Outdated Components	Very unlikely	Moderate	Low	Deletion of unused/redundant dependencies and imports.	N/A
A07:2021-Identification and Authentication Failures	Likely	Moderate	Moderate	2+-factor authentication could be used to counteract this vulnerability Through email login links for auth.	N/A
A08:2021-Software and Data Integrity Failures	Unlikely	Moderate	Moderate	Ensure that CI/CD has been set up, and has all the important stages tested, and ensure that pushed code passes all tests.	N/A

A09:2021-Security Logging and Monitoring Failures	Very likely	Severe	High	Ensure all login, access control, and server-side input validation failures can be logged	N/A
A10:2021-Server-Side Request Forgery	Very likely	Severe	Moderate	In the application layer, all data given or provided by the users, should be sanitized, and validated (using regex ect.). Whitelisting of supported sites for user uploaded content.	N/A

Explanation:

It is quite difficult for me to gage the overall security of the application, as the extent to which it may be exploitable depends on the attacker’s time and skill. Given an infinite amount of time, I think any application has exploitable vulnerabilities. A risk assessment should be done in a way where you need to be of the approach that your application is always vulnerable, and if some new exploit arises as for example Log4j incident, you should be quick to implement changes/fixes and never write off the application as “fully” secure.