# Security Report

Картина, която съдържа светлина

Описанието е генерирано автоматично

Content

[Security Report 1](#_Toc135917973)

[OWASP Top 10 Risks 3](#_Toc135917974)

[Reasoning 4](#_Toc135917975)

## OWASP Top 10 Risks

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Likelihood | Impact | Risk | Actions possible | Planned |
| A01 Broken Access Control | Medium | Severe | Moderate | Encrypting the parameters in the URL. | Yes |
| A02 Cryptographic Failures | Unlikely | Severe | Low |  |  |
| A03 Injection | Unlikely | Severe | Low |  |  |
| A04 Insecure Design |  |  |  |  |  |
| A05 Security Misconfiguration |  |  |  |  |  |
| A06 Vulnerable and Outdated Components |  |  |  |  |  |
| A07 Identification and Authentication Failures |  |  |  |  |  |
| A08 Software and Data Integrity Failures |  |  |  |  |  |
| A09 Security Logging and Monitoring Failures |  |  |  |  |  |
| A10 Server-Side Request Forgery |  |  |  |  |  |

## Reasoning

***A01 Broken Access Control***: As a whole, my application is protected from users gaining privileges that they are not authorized to have. The only thing that needs to be fixed is to encrypt the URL parameters, so they wouldn’t be able to access the profile page, which is a big security risk, but I plan to fix it. Other than this flaw, however, there are other weak spots through which users can gain permissions they are not supposed to have.

***A02 Cryptographic Failures***: Encryption is used for the passwords which are the only truly sensitive part of the data the users need to fill in. They can’t be decrypted. I am using axios which encrypts the requests and makes them more secure.

***A03 Injection***: The SQL queries are using parameterized statements which means the risk of injection is fairly low.