|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Likelihood | Impact | Risk | Actions possible | Planned |
| A01:2021 – Broken Access Control | Low | Severe | Low | Extra verification for accessing someone else’s page with the same role as you have. | Yes |
| A02:2021 – Cryptographic Failures | Very low | Severe | Very low | Stick with the modern, secure encrypting classes and follow security features. | No, risks accepted |
| A03:2021 – Injection | Moderate | Severe | Moderate | Enforce validation on both frontend and backend, rely on testing for vulnerabilities. | Yes |
| A04:2021 – Insecure Design | Low | Moderate | Low | Discuss with professionals to help evaluate and design security and privacy-related controls. | Yes |
| A05:2021 – Security Misconfiguration | Unlikely | Low | Unlikely | Stick with the modern, secure, libraries, app frameworks and databases. | No |
| A06:2021 – Vulnerable and Outdated Components | Very low | Low | Low | Be aware of new component updates and their new security features. | No |
| A07:2021 – Identification and Authentication Failures | Moderate | Severe | High | Implement more security features and validate user input. | No, risks accepted |
| A08:2021 – Software and Data Integrity Failures | Low | Low | Moderate | Make sure the external sources are downloaded from a safe location and ensure secure communication channels among the layers of the application (e.g., deserialization between React and Spring Boot). | No |
| A09:2021 – Security Logging and Monitoring Failures | Low | Moderate | Moderate | Ensure all login, and server-side input validation failures can be logged with sufficient user context to identify suspicious or malicious accounts and held for enough time to allow delayed forensic analysis. | No |
| A10:2021 – Server-Side Request Forgery (SSRF) | Low | Moderate | Low | Validate URLs and do not send raw responses to clients. Perform validation for user input. | No |