# OWASP Report

1. A01:2021-Broken Access Control

-Broken access control refers to the ability of users to performs acts outside of their permissions.   
  
To prevent this, we are using token authentication with JWTs and roles to indicate every user’s permission. //Clarify more where its used

1. A02:2021 – Cryptographic Failures

-Cryptographic failures could happen with sensible data being transferred as clear text. An example of such data are passwords  
  
To avoid them, we are hashing the passwords with BCrypt, which is currently the industry standard

1. A03:2021 – Injection

-This occurs when User-supplied data is not validated, filtered, or sanitized by the application. Hostile data is directly used or concatenated. The SQL or command contains the structure and malicious data in dynamic queries, commands, or stored procedures

To prevent this, we must implement input validation and can also use safe APIs which avoids using the interpreter entirely, provides a parameterized interface, or migrates to ORM.

1. A04:2021 – Insecure Design

-One form of this is the lack of business risk profiling inherent in the software or system being developed, and thus the failure to determine what level of security design is required  
  
To prevent it we can establish and use a secure development lifecycle and use of secure design libraries or components

1. A05:2021 – Security Misconfiguration

-Common causes of this is for example improperly configured permissions on cloud services. Default accounts and their passwords are still enabled and unchanged. Error handling reveals stack traces or other overly informative error messages to users

To prevent it, we should remove or do not install unused features and frameworks. An automated process to verify the effectiveness of the configurations and settings in all environments

1. A06:2021 – Vulnerable and Outdated Components

-This refers to the usage of old dependencies or deprecated libraries or if you do not scan for vulnerabilities regularly and subscribe to security bulletins related to the components you use

To prevent it, we should remove unused dependencies from the application, only use components from official and trusted sources and constantly monitor for components that are unmaintained

1. A07:2021 – Identification and Authentication Failures

* Identification and Authentication Failures occur when the software is vulnerable to for example brute force attacks, where the attacker uses weak password to attempt to break into the account

To prevent this, usage of 2FA(2 factor authentication) is advised, as well as security checks against weak passwords (top 10000 worst passwords)

1. A08:2021 – Software and Data Integrity Failures

-This is related to code and infrastructure that is not protected against integrity violations, such as malicious alteration to third party plugins or libraries. Another cause could be the usage of an insecure CI/CD

To prevent this, we can use digital signatures to verify the external components have not been altered and CI/CD has proper segregation configuration and access control.

1. A09:2021 – Security Logging and Monitoring Failures

-Detection of auditable breaches, such as failed logins, high value transactions and more, or usage of unclear log messages. Only local storage of logs or logs of application and APIs are not monitored for suspicious activity

1. A10:2021 – Server-Side Request Forgery (SSRF)

* SSRF flaws occur whenever a web application is fetching a remote resource without validating the user-supplied URL. It allows an attacker to coerce the application to send a crafted request to an unexpected destination, even when protected by a firewall, VPN, or another type of network access control list (ACL)  
    
  SSRF can be prevented by segment remote resource access functionality in separate networks to reduce the impact of SSRF. Segment remote resource access functionality in separate networks to reduce the impact of SSRF

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Likelihood | Impact | Risk | Actions possible | Planned |
| A1: Broken access control | Unlikely | Severe | Low | N/a, fixed | Yes |
| A2: Cryptographic failures | Very unlikely | Severe | Low | N/a, fixed. Used passwords are encrypted | Yes |
| A3: Injection | Unlikely | Severe | Low | N/a, fixed.  Usage of JPA prevents SQL injections | Yes |
| A4: Insecure design | Likely | Severe | High | Usage of already ready to use and tested libraries and components | Yes |
| A5: security misconfiguration | Likely | Severe | Medium | Currently only using JWTs and spring security configuration. | No |
| A6: Vulnerable and outdated components | Likely | Severe | High | Remove unused dependencies and actively scan for ones that are not maintained anymore | No |
| A7: Identification and Authentication failures | Likely | Severe | High | Implement 2 factor authentication and implement weak password checks. | No |
| A8: Software and Data integrity Failures | Likely | Severe | Medium | Usage of a secure CI/CD pipeline | Yes |
| A9: Security logging and Monitoring failures | Very likely | Severe | High | Ensure all failures can be logged with sufficient context and that the log data is encoded | No |
| A10: Server side request forgery | Very likely | Severe | High | Segment remote resource access functionality in separate networks to reduce the impact of SSRF | No |