Security Review

Nadko Dentals

|  |  |  |
| --- | --- | --- |
| Version | Date | State |
| 1.0 | 20.06.2021 | Initial Release |

Contents

[OWASP TOP 10 2](#_Toc75124599)

[Injection ✅ 2](#_Toc75124600)

[Broken Authentication ✅ 2](#_Toc75124601)

[Sensitive Data Exposure ✅ 2](#_Toc75124602)

[XML External Entities ✅ 2](#_Toc75124603)

[Broken access control ✅ 2](#_Toc75124604)

[Security Misconfiguration ✅ 2](#_Toc75124605)

[Cross-Site Scripting XSS ❌ 2](#_Toc75124606)

[Insecure Deserialization ❌ 2](#_Toc75124607)

[Using Components with Known Vulnerabilities ❌ 3](#_Toc75124608)

[Insufficient Logging & Monitoring ❌ 3](#_Toc75124609)

# OWASP TOP 10

## Injection ✅

Nadko Dentals Is protected from SQL injection on account of it not using any SQL native queries. Instead database communication is handled by methods like \*reposiroty\*.save().

## Broken Authentication ✅

Authentication in Nadko Dentals is handled by JSON Web Tokens therefore no actions are authorized either in the front end or in the Rest API unless they come with a header featuring an active JWT token.

## Sensitive Data Exposure ✅

Password protection in “NadkoDentals” is handled by Bcrypt type encryption provided by the Spring framework’s PasswordEncoder interface. No other sensitive data is encrypted.

## XML External Entities ✅

Nadko Dentals doesn’t handle external XML files therefore isn’t affected by this vulnerability.

## Broken access control ✅

Authorization is handled by the back-end. Specifically the @PreAuthorize() Spring Framework tag and the Spring framework class authorizeRequests() method

## Security Misconfiguration ✅

Nadko Dentals’ security is setup well and without needless vulnerability however it has not been thoroughly tested due to the lack of expertise in that field by the developer.

## Cross-Site Scripting XSS ❌

Nadko Dentals is vulnerable to XSS as many fields are not regex validated to exclude a JS injection.

## Insecure Deserialization ❌

Many fields are not regex validated therefore Nadko Dentals is vulnerable in this regard

## Using Components with Known Vulnerabilities ❌

Nadko Dentals does use some input files that use or override a deprecated API.

## Insufficient Logging & Monitoring ❌

Nadko Dentals implements no logging or monitoring