Security Document

OGBEIDE GODFREY OSAS. (3785580)

Contents

Indexing

Version	Change
V1	Added 10 OWASP Criterium

Introduction

Security is an important think in development, both for the user and the developers. This is to ensure that a system has certain security associated with it. In this document I will be checking the security of my application with regards to OWASP top 10 Application Security Risks.

Top 10 security risks

	Likelihood	Impact	Risk	Action possible	Planned
A01:2022-Injection	High	Severe	High	Add authentication	Yes
A02:2022-Broken Authentication	Very unlikely	Severe	Low	No password shown	Yes
A02:2022-Sensitive Data Exposure	High	Severe	High	Add validation	N/A
A04:2022-Xml External Entities	Unlikely	Low	Low	Create a design document	N/A
A05:2022-Broken Access Control	Likely	Severe	Moderate	Design test	N/A
A06:2022-Security Misconfigurations	Likely	Severe	High	Update everything	N/A
A07:2022-Identification and Authentication Failures	Likely	Severe	High	Authentication testing	N/A
A08:2022-Software and Data Integrity Failures	Low	Low	Low	N/A	N/A
A09:2022-Security Logging and Monitoring Failures	High	Severe	High	N/A	N/A
A10:2022-Server-Side Request Forgery	High	Severe	High	N/A	N/A

A1:2022-Injection

- Is the risk applicable for the application? No, it does not
- How did the risk impact your implementation? How does this effect the risk?

  In my application sql injections have been handled with the use of JPA and this ensures no manually written queries have variables as parameters

A2:2022-Broken Authentication

- Is the risk applicable for the application?
  No this does not

- How did the risk impact your implementation?

```
.sessionManagement()
    .sessionCreationPolicy(SessionCreationPolicy.STATELESS)
```

This code states my session policy should be stateless which means my backend will not store any information about the logged in user in the memory.

A3:2022-Sensitive Data Exposure

- Is the risk applicable for the application?
  No this does not

- How did the risk impact your implementation?
  My application handles this by making use of both password encryption when the user is being signed up and JSON Web Tokens(JWT) for log in security, which protects user data.

  Token Generation

```
public String generateToken(Authentication authentication) {

    UserPrincipal userPrincipal = (UserPrincipal) authentication.getPrincipal();

    Date now = new Date();
    Date expiryDate = new Date(now.getTime() + jwtExpirationInMs);

    return Jwts.builder()
            .setSubject(Long.toString(userPrincipal.getId()))
            .setIssuedAt(new Date())
            .setExpiration(expiryDate)
            .signWith(SignatureAlgorithm.HS512, jwtSecret)
            .compact();
}
```

  Password Encryption

```
user.setPassword(encoder.encode(signupRequest.getPassword()));
```

A4:2022-XML External Entities (XXE)

- Is the risk applicable for the application?
  No this is not a risk to my application.

- How did the risk impact your implementation?

```
@RequestMapping("user")
public ResponseEntity<?> userLogin(@RequestBody SignUpRequest signupRequest) {
    try {
        User user = IUserService.signUpUser(signupRequest);
        return  ResponseEntity.ok(user);
    }catch(Exception e ) {
        return ResponseEntity.badRequest().body(new ApiResponse(e.getMessage(), ""));
    }
}
```

A5:2022-Broken Access Control

- Is the risk applicable for the application?
  No this does not affect my application.

- How did the risk impact your implementation?

  My application handles this using Authentication and Authorization with the integration of spring security. With Authorization added, certain api calls are restricted to the admin user.

A6:2022-Security Misconfiguration

- Is the risk applicable for the application?
  No this is not a risk to my application

- How did the risk impact your implementation?
  Authentication and Authorization with JWT access tokens as well as password encoding, this ensures every access variable is distinct.

A7:2022-Cross-Site Scripting XSS

- Is the risk applicable for the application? Yes, this not yet handled in my application, but I will research this and apply it in my application.

A8:2022-Insecure Deserialization:

- Is the risk applicable for the application?
  No this not a risk to my application.

- How did the risk impact your implementation?

```
if (productData.isPresent()) {
    return new ResponseEntity<>(productData.get(), HttpStatus.OK);
} else {
    return new ResponseEntity<>(HttpStatus.NOT_FOUND);
}
```

This is handled in my application this by not sending null datatypes directly as a return value in my controllers.

A9:2022-Using Components with Known Vulnerabilities

- Is the risk applicable for the application?
  No, It does not.

- How did the risk impact your implementation?

  My application does not make use of vulnerable 3rd party components. I do not make use of any 3rd party APIs of some sort.

A10:2022-Insufficient Logging & Monitoring

- Is the risk applicable for the application?
  This is not a risk in my application.

- How did the risk impact your implementation?

```
}catch(Exception e )
    {
     logger.info("Error in authenticateUser ",e);
     return ResponseEntity.badRequest().body(new ApiResponse(e.getMessage(), ""));
    }
```