OWASP Report

Name: Lucas Jacobs
Pcn: 490692
Class: S3 CB-05
Teachers: Jacco Snoerren, Maja Pesic
Date: 12-12-2022
Word count: 1375

Contents

Analyse

	Likelihood	Impact	Risk	Actions Possible	Planned
A01:2021-Broken Access Control	High	Severe	High	Yes	No, for this semester not applicable.
A02:2021-Cryptographic Failures	Unlikely	Severe	Low	Yes, Password hashed plus salting in the database.	N/A
A03:2021-Injection	Unlikely	Moderate	Low	Yes.	Yes
A04:2021-Insecure Design	Unlikely	Severe	Moderate	Yes.	Yes
A05:2021-Security Misconfiguration	Likely	Severe	High	Yes, update outdated configurations.	No, for this semester not applicable (deprecated security files).
A06:2021-Vulnerable and Outdated Components	Likely	Severe	High	Yes, update your versions software	No, for this semester not applicable (deprecated security files).
A07:2021-Identification and Authentication Failures	Unlikely	Severe	Moderate	Yes, strict rules for password	Yes
A08:2021-Software and Data Integrity Failures	Unlikely	Severe	Moderate	Yes	Yes
A09:2021-Security Logging and Monitoring Failures	Likely	Severe	High	Yes	No, for this semester not applicable
A10:2021-Server-Side Request Forgery	High	Moderate	Moderate	Yes	Yes

Likelihood (chance of something happening) rating:

1. Very unlikely
2. Unlikely
3. Likely
4. High

Impact rating:

1. normal (= no impact)
2. mild (= low impact presence)
3. moderate (= between normal and severe)
4. moderately severe (= not the worst case, nor in the middle)
5. severe (= biggest impact)

Risk rating:

1. Low (= zero to none risk)
2. Moderate (= between low and high)
3. High (= biggest risk)

Reasoning

A1: broken access control

Access control ensures that users stay within the bounds of their specified permissions. Failures frequently result in the unauthorized disclosure of information, the modification or deletion of all data, or the execution of business operations outside the user's scope.
Within my application, there are two roles which are ‘USER’ and ‘ADMIN’. Admin has different permissions than a regular user, therefore in my code, there are different configurations so that certain roles have access to do specific operations and some do not. Also the same goes for if some are authenticated or not.
Furthermore, I only specified what needs a connection by preventing a lot of Cross-Origin Resource Sharing (CORS).

A02: Cryptographic Failures

This describes that it is important to protect your data. Especially, it is important to secure information that falls under privacy laws. With my project, there are some sensitive data such as a password. All the passwords are stored as a hash which is also salted, this way when the database is hacked they still cannot access all the accounts.

A03: Injection

The incorrect injection can occur when data is for example not filtered or validated. This can make an application vulnerable to attack. In my application they first go threw validation checks before executing an operation. Moreover, what can also cause injection vulnerabilities, is sensitive data that is passed through an HTTP request. Therefore, what is not safe right now is when you want to chat with someone the Id of the chat will be revealed in the URL, so when someone else enters the same URL they can see the whole chat history. When there is time, I certainly will look at a solution to prevent passing in a parameter.

A04: Insecure Design

To begin with, insecure design is the lack of integration and security controls into the application during the development cycle. In my application, we try to prevent this by testing my code by using the unit and integration tests. With this, it will validate that all critical flows are resistant to potential vulnerabilities. Also in the application, there are a lot of libraries such as the spring security library.

A05: Security Misconfiguration

Security is important regarding your application. We want to prevent outdated software and unnecessary features that being enabled or installed. Also, the values of different frameworks of your application may be not set to secure values.
Regarding my application, there are some deprecated files concerning the web security files. This can be updated at a later stadium of the project but it is not the main purpose of this semester.
In addition, when using certain libraries for my application, they are up-to-date and only implemented when needed.

A06: Vulnerable and Outdated Components

This has to do with vulnerabilities that can occur when software is not updated. Especially it can occur when a business only checks updates every few months leaving the software with unnecessary exposure to fixed vulnerabilities. Regarding my application, this does not apply due to the application only has a development time of half a year. On the other side, if this application of mine will be used in the future it can be needed to update the software before actually publishing.

A07: Identification and Authentication Failures

These failures are vulnerabilities related to the application authentication. This can occur when passwords are weakly stored (plain text, weakly hashed). Also, it can happen when the application permits weak passwords, ineffective multi-factor authentication, permits multiple things such as automated or brute force attacks.
The password is hashed en salted for my application when stored in the database. Furthermore, it allows every password which can lead to this failure. In the future, the application can have automated requirements where you, for example, a password is required valid when it has at least eight characters, one capital, one number, and one special character.

A08: Software and Data Integrity Failures

This can occur when code implementation and other infrastructure lack the ability to protect code against all integrity violations. This can happen when you use libraries or repositories from some untrusted source. A hacker can access the system without authorization. The system will become vulnerable to multiple attacks. For my application, the CI/CD pipeline can get proper segregation, and configuration and has access control in the right places to protect the integrity of the code that is being built and deployed.

A09: Security Logging and Monitoring Failures

You can detect infiltration signs by identifying activity patterns on your networks with the help of logging and monitoring. This way you can detect certain suspicious patterns that can lead to attacks. Therefore, by logging and monitoring, you can take the needed action. For my application, I will use logging and monitoring to see if certain events go as intended.

A10: Server-Side Request Forgery

An attacker targets applications that support using data imports from URLs or allow the application to read certain data of a URL. The attacker is abusing server functionalities to access or modify data. The same thing is being explained at A03, what is not safe right now is when you want to chat with someone the Id of the chat will be revealed in the URL, so when someone else enters the same URL they can see the whole chat history. As mentioned, when there is time I will certainly try to come up with another solution.

Conclusion

Even though you sometimes think ‘my application is pretty secure’, sometimes it can be exactly the opposite.

Is my application ready for the world wide web?
Based on going through OWASP's top 10 and comparing it with my application, there are still some concerns and insecurities.
What can be improved for the future, first of all, password validation? This is important regarding the Identification and Authentication Failures. Secondly, this has to do with Server-Side Request Forgery. When you chat with someone you can see the id of the chat in the URL, this can result in major data leaks or worse because other persons can just type in the URL and see the whole chat history.

Even though my application is not perfectly secure most principles of the OWASP top 10 are covered. To name a few:

- Protecting sensitive data (password hashing).
- Using up-to-date software.
- To prevent a lot of Cross-Origin Resource Sharing (CORS), I only specified what needs a connection.

Therefore, in my opinion, it is sufficient for the state that it is in.

References

Welcome to the OWASP Top 10 - 2021. (2021). From owasp.