Security Report

Fontys
University of Applied Sciences

09/06/2024
Version: 1.0

Esther Wolfs: 3329984

Introduction

This document will cover the security measures that will be taken for Concert Meetup. It will include research into what security by design means and it will look into the OWASP top 10 for high vulnerabilities.

Security by design

When creating an application it is important to keep security in mind from the beginning. Security by design means that security is a focus point during the whole software development lifecycle.

OWASP

The OWASP top 10 is a standard awareness document for developers and web application security. This list shows the biggest security vulnerabilities.This list was last updated in 2021.

A01:2021 Broken Access Control

Broken access control is the most serious security risk. Access control enforces policy such that users cannot act outside of their intended permissions. To prevent this authentication and authorisation is implemented. Certain endpoints are only available to logged in users and are only available to certain roles, only admins can create and update concerts. To access personal data the id of the logged in user is checked and only available if the id that is sent with the token matches the id in the database.

In the front end certain routes are protected, which means they are not accessible to users who are not logged in. This means a user can't just type in the url of a page and still visit it when they are not logged in.

Another broken access control risk is manipulation of JWTs. The JWT is sent with the request where authorisation is needed. By manipulating the JWT the user can change their role from a normal user to an admin user and gaining more privileges that way. For authentication and authorisation Auth0 is used, instead of self creating and validating a JWT.

CORS is enabled in the application, meaning only allowed origins can make requests to the API.

A02:2021 Cryptographic failures

The number two biggest risk is cryptographic failures. This means sensitive data can be exposed, due to not storing sensitive data the right way or failures related to cryptography. The most common weaknesses are hard-coded passwords or a broken or risky crypto algorithm.

Auth0 is used for authentication and authorisation, this means sensitive login information like emails and passwords are stored in their database, and not the database of concert meetup.

This is a risk because it is never certain what kind of encryption they use for their databases, but on the other hand it is a professional company specialised in making login systems so they have more knowledge.

Sensitive data that is unnecessary is not stored in the concert meetup database, like the emails of users. By not storing unnecessary sensitive data it is impossible to steal the data.

There are no hard-coded passwords or other sensitive data like connection strings in the Concert Meetup code. All passwords are stored as environment variables and Azure key vault is used.

A03:2021 Injection

SQL injection is a common weakness. This means that malicious users can enter SQL statements in forms in the web application that get sent to the backend to perform queries on the database. To prevent this the data should be validated before it gets sent to the backend. Entity Framework is used as an ORM, no direct sql queries are used. This however does not completely eliminate the risk of SQL injection. Data should also be limited to prevent all data from being returned.

Cross site scripting is also a risk, this is a type of injection where malicious scripts are injected into otherwise trusted websites. One of the most common XSS attacks is a DOM based attack. React is used as the frontend for the Concert Meetup application, it already has some built in XSS prevention.

A04:2021 Insecure design

This risk is related to design issues and architectural flaws. Examples of common weaknesses are generating error messages containing sensitive data or unprotected storage of credentials. Secure design is a culture and methodology that constantly evaluates threats and ensures that code is robustly designed and tested to prevent known attack methods. To prevent this the application should be constantly tested and monitored for security issues. Writing unit and integration tests should be included to test and validate all critical flows. The error messages only show the minimal information so that the user knows what went wrong, but it does not contain any sensitive data.

Unit tests are made with a high test coverage, to ensure the quality of the application. Sonarcloud is used to scan for security issues and vulnerabilities each time new code is pushed.

A05:2021 Security misconfiguration

The fifth risk in the owasp top 10 is security misconfiguration. This can happen when important security features or configurations are disabled, or unnecessary features are enabled. Normal users have different privileges from admin users, meaning not everyone can access all features. Users have to choose a strong password and have to enter their own password when creating the account. No default passwords are used. Users also have the option to change their password if they want to and if they have verified their email they can also request a new password when they forget it.

A06:2021 Vulnerable and outdated components

Using third party components and libraries can be a risk, as they are not always updated and maintained by the owner. The dependencies should also be updated to a new version when it is available. The use of outdated browsers or operating systems by the users of the application can also be a risk for the security of Concert Meetup.

A dependency check tool will be used to analyse dependency risks and automate this process.

Unused dependencies should be removed from the application, as they can become a security risk. Right now the Concert Meetup has no installed dependencies that are not being used.

A07:2021 Identification and Authentication failures

Confirmation of the user's identity, authentication and session management is critical to protect against authentication related attacks.

One way to prevent this is to prevent users from choosing weak passwords like "password" or "admin". To prevent this a check should be made against the top 10.000 common passwords and a password length and complexity check should be according to the National Institute of Standards and Technology guidelines. Using multi factor authentication can also lessen the risk of identification and authentication failures, however it should be done right.

A08:2021 Software and data integrity failures

This common risk is related to code and infrastructure that does not protect against integrity violations. An insecure CI/CD can introduce the potential for unauthorised access, malicious code or system compromise. In the CI/CD pipeline no hardcoded passwords or tokens are used, they are stored as github actions variables. To prevent this a digital signature should be used to verify the software or data is from the expected source and not altered. Libraries and dependencies that are used should be checked. The CI/CD pipeline should also have proper segregation, configuration and access control. When working in a team new code should be reviewed before it gets pushed.

A09:2021 Security logging and monitoring failures

Detecting and responding to security breaches is critical. Logging and monitoring should be sufficient. Warnings and errors should log clear and understandable messages, these logs should not only be stored locally. The API should also be monitored for suspicious activity. The log data should be encoded to prevent malicious injections or attacks on the logging or monitoring systems.

By using Auth0 login attempts are automatically monitored and logged. It is possible to see which users tried to login and if the login was successful. It also generates a report of failed login attempts. This means brute force password hacking can be detected.

A10:2021 Server side request forgery

The last security risk in the OWASP top 10 is server side request forgery. SSRF flaws occur whenever a web application is fetching a remote resource without validating the user supplied URL. By enforcing a “deny by default” firewall policies or network access control rules all but essential traffic can be blocked. Azure has a lot of firewall and security settings, like which IP addresses can make a connection to the database. By using these settings the application can become more secure.

Conclusion

There are a lot of common security issues and weaknesses. By looking at the top 10 most common/dangerous risks the application can be made more secure. It is important to keep the security of the application in mind when designing the architecture, so it is as secure as possible right from the beginning.

References

OWASP. (2021). *OWASP Top 10*. OWASP Top 10:2021. Retrieved June 10, 2024, from https://owasp.org/Top10/