Security Design

Individual Project: FestivalConnect

Name: Lucas Jacobs
Class: S-A-RB06
PCN: 490692
Student number: 4607368
Technical teachers: Felipe Ebert, Bartosz Paszkowski
Semester coach: Gerard Elbers

Table of Contents

Glossary

Term	Definition
Transport layer security (TLS)	A cryptographic protocol that provides end-to-end security of data sent between applications over the Internet.
Input validation	The process of testing input received by the application for compliance against a standard defined within the application.
RBAC	Role-based access control (RBAC), also known as role-based security, is a mechanism that restricts system access. It involves setting permissions and privileges to enable access to authorized users.
XSS	Cross-site scripting (XSS) is an attack in which an attacker injects malicious executable scripts into the code of a trusted application or website.
SQL Injection	SQL injection (SQLi) is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database.
SSRF	Involves an attacker abusing server functionality to access or modify resources.

Introduction

This document will outline what will be important during the software development life cycle (SDLC) of FestivalConnect. This is done by first investigating what vulnerabilities can occur, how the system can be attacked, and eventually making choices for security that are well-argued.

Understanding Security-By-Design

Security by design (SbD) is the phase in your software development where you consider security for relevant activities in your project. This is done to make the application free of vulnerabilities and prevent attacks by doing continuous testing, authentication precautions, and the best programming practices. It becomes more important by the day, internet of things is rapidly evolving, and mobility is becoming more important than ever. With devices such as phones, tablets, and laptops you want to access FestivalConnect securely. (Wigmore, n.d.) Therefore, we need to have SbD to ensure that all the components that we create are from the get-go with security in mind. This means that there is a proactive approach to integrating security from the start. (Nath, 2022)
For the SDLC to consider the SbD, we want to have security kept in mind the whole cycle. So for each cycle in my sprint the following needs to be considered.

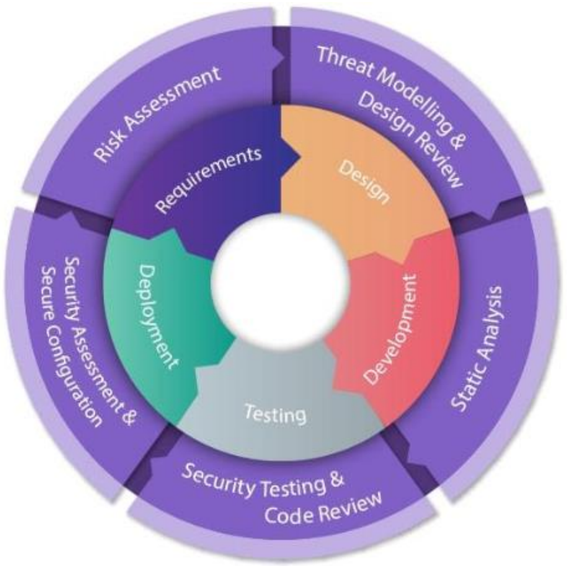
Risk Assessment
Threat Modelling & Design Review
Static Analysis
Security Testing & Code Review
Security Assessment & Secure Configuration
Requirements
Design
Development
Testing
Deployment

Figure 1: Secure Software Development Life Cycle (SSDLC)

- **Requirements**: setting up the needed functional requirements related to security and what security considerations are needed to complete this functionality.
- **Design**: Updating the design of the application. The non-functional requirements need to be considered. Throughout the sprints update the information.
- **Development**: Critically looking at my code and following coding guidelines. This can be done manually or using some automated tools.
- **Verification**: Have automated tests and deployment to verify design and requirements. This will be automated in the CI/CD pipeline.
- **Maintenance**: Keeping an eye on that there can still be vulnerabilities that slip through. So keep looking for improvements.

(Secure Software Development Lifecycle (SSDLC), n.d.)

Relation between SbD and Privacy

The relation between security by design and privacy lies in protecting sensitive information. This means that we need to implement security measures in the software design. Therefore, developers need to design, to protect user data from attacks of for example unauthorized access, breaches, and misuse. So privacy will regulate security in the sense of knowing how the user data needs to be handled and transferred securely. Finally, these two work together to make FestivalConnect a more trustworthy application. (Dabah, 2023)

Analysis

Boundary Crossing

Within my technical design (Jacobs, 2024, Technical Design, C4) will show the diagram of how the architecture is structured of FestivalConnect. The following shows what the data entry and outputs are with the potential security risks that can occur.

Data entry points:

- **User Login**: With a login in FestivalConnect you need to provide your username and password.
- **User Registration**: Making an account in FestivalConnect will be with basic user information.
- **Community Creation**: Festival organizer creates a community within FestivalConnect.
- **User Interaction**: Post comments, join a community and update their profile.
- **Notification delivery**: When a notification needs to be sent to a user due to a new event or community updates.

Data output points:

- **User authentication response**: When a user tries to log in to FestivalConnect, the response will be sent on the login, either granting or denying access.
- **Registration confirmation**: The FestivalConnect system will approve a successful user registration and send a confirmation message to the user.
- **User interaction feedback**: FestivalConnect sees user interaction, such as posting comments or joining a community.
- **Notification delivery feedback**: When the FestivalConnect system will confirm a successful delivery of a notification to a user.

Potential Security Risks:

- **Unauthorized access**: during login or registration, there is a risk that unauthorized users gain access to the system.
- **Data interception**: Data that is being transmitted during login, registration, or other user interaction can be intercepted by an attacker.
- **Data manipulation**: Attackers may want to manipulate data during a request, such as modifying user profiles or community information.
- **Denial of Service**: An attacker can do a DoS attack which can result in service downtime.
- **Data breaches**: Without taking proper cause, weakness in security can lead to data breaches, which can leak sensitive data of users to unauthorized persons.
- **Integration vulnerabilities**: When there is going to be the usage of an external API can lead to attackers finding a way to gain access to the system or to manipulate the data.

Secure the data:

- Encrypting the data transitions to secure them so that they remain confidential and cannot be easily intercepted or manipulated. For the front end, FestivalConnect uses Axios to make requests, such as HTTP and HTTPS. Axios will use the underlying node's HTTP and HTTPS to use TLS encryption to secure communication. (vcsjones, 2018) All the incoming requests in the API gateway will be fully decrypted to process the HTTP request
- **Authentication mechanisms**: Look at the best practices to implement authentication, and verify the identity of users with the use of for example RBAC, to prevent unauthorized access which will give the option to restrict access to sensitive data to other users..
- **Input validation**: Having proper checks on if the data is correct before sending a request. This will prevent XSS, SQL Injection, and other attacks that can significantly impact FestivalConnect. (Input Validation Cheat Sheet, n.d.)
- Monitoring to track vulnerabilities such as DoS attacks.

Role-Based vs Credentials

Role-based access controls (RBAC) and permission-based access controls (PBAC) are needed for law firms to manage the data of FestivalConnect securely. RBAC is based on gaining access to predefined roles like the festival-goer, festival organizer, and admin, which increases security. PBAC will allow for more granular control over individual access rights, which can assure each user has access to their specific responsibilities. (Cheng, n.d.)
Credentials are used to check the user's identity, such as the username, password, etc. These will be used during the authentication process to confirm that the user is who they claim to be.
So to conclude the final thing, credentials are used to authenticate users, while roles are used to authorize users by assigning them permissions to even further detail PBAC rules.

Security by Design: Investigation

FestivalConnect looked into multiple sources to investigate SbD to look at multiple sources such as CIP, OWASP, and Microsoft’s Security Engineering. The following points are important for FestivalConnect.

CIP

- **SSD-1: Hardening of technical components:** All the unused things of the system such as inactive user accounts will be removed to reduce the number of potential attacks.
- **SSD-4: Session encryption:** Important to make sure that data deliveries are securely transmitted.
- **SSD-5 Determine the identity of an external user:** Authentication and authorization are important to prevent unauthorized applications from accessing the system.
- **SSD-6 Establish the identity of an internal user:** Important to prevent unauthorized access.
- **SSD-7: Segregation of functions**: With the use of this, it will prevent unauthorized access to sensitive data to make sure that users only have access to what is needed for their role.
- **SSD-8: Least Privilege:** Beneficial for having users granted minimum level access required to perform their tasks.
- **SSD-11: System Use Notification:** Give more clarity to the user to give more transparency and security of what the user can access.
- **SSD-**15: Separation of Presentation, Application, and Data: Each service will have a layered structure to ensure confidentiality of the application logic and data.
- **SSD-19: Normalization of input:** With normalization of input, will prevent abuse of data, loss, or manipulation of the application logic.
- **SSD-22: Input Validation**: validate the user input first, to prevent manipulation which can affect the application in a harmful way.
- **SSD-27: Error handling:** Good error handling is important to detect security vulnerabilities.
- **SSD-28: Comment (lines):** The code that is exposed to the user, does not contain any comments, to prevent leaks on sensitive parts of the system.

(Koers, 2014)

OWASP

FestivalConnect looked at the OWASP Top Ten of 2023, since 2024 is being revealed in September 2024.

	Likelihood	Impact	Risk
A01-Broken Access Control	High	Severe	High
A02-Cryptographic Failures	Unlikely	Severe	Low
A03-Injection	Unlikely	Moderate	Low
A04-Insecure Design	Unlikely	Severe	Moderate
A05-Security Misconfiguration	Likely	Severe	High
A06-Vulnerable and Outdated Components	Likely	Severe	High
A07- Identification and Authentication Failures	Unlikely	Severe	Moderate
A08-Software and Data Integrity Failures	Unlikely	Severe	Moderate
A09-Security Logging and Monitoring Failures	Likely	Severe	High
A10-Server-Side Request Forgery (SSRF)	High	Moderate	Moderate

(Nir, 2023)

Legend:

Likelihood of something is occuring:

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

Impact Rating:

1. Normal (No impact)
2. Mild (Low Impact Presence)
3. Moderate (Notable Impact)
4. Moderately Severe (Really Notable Impact)
5. Severe (Biggest Impact)

Risk rating:

1. Low
2. Moderate
3. High

A01-Broken Access Control
With the authorization, we separate the functionalities that users can do, with bounding them to specified permissions. When this fails, unauthorized users can access information or even modify and delete data, with also having the possibility to do business logic that they usually cannot access. The application has three roles: festival goer, festival organizer, and admin. Festival organizers and admins have other permissions than a regular user. This is configured in the code that roles have access to certain specific operations and some do not. Furthermore, in the API gateway authenticated functionalities are also checked with a policy. Also by keeping in mind the SSDLC, to consider the access control requirements early on in the design process.

A02-Cryptographic Failures
It is important to secure and protect sensitive information. How FestivalConnect handles sensitive data is listed in (Jacobs, 2024, Data Storage and GDPR Compliance Strategy). Furthermore, FestivalConnect encrypts passwords with hashing and salting, keeping into account the requirements for creating a password. Moreover, having a key vault that prevents having a hard-coded password in the code.

A03-Injection
Not properly validating input can lead to attackers injecting data such as SQL queries or code snippets. As given in SSD-22 the application will do validation checks before passing it through the back end. Also, in the requirements in the next chapter, the input and output data entries will be listed down to properly check them.

A04-Insecure Design
Flaws in the design of the application will be tried to be avoided to do the following:

1. Doing testing like unit tests, integration tests, E2E, and performance tests.
2. Addressing vulnerabilities early on the SSDLC.
3. Automatically having security checks in the pipeline.

A05-Security Misconfiguration
To prevent security misconfiguration in configs we will use automated deployment to have a centralized and consistent configuration of an application deployment. Also, the other security requirements are taken into account that are listed in the chapter ‘Requirments’.

A06-Vulnerable and Outdated Components
FestivalConnect uses a lot of third-party services. Therefore to securely do this, we will frequently check on updates for the services and only use services from trustworthy companies, this is also specified in what needs to be verified in the requirements chapter.

A07- Identification and Authentication Failures
FestivalConnect will provide proper Authentication and setup requirements for passwords. This is further specified in the chapter requirements. When there are weak passwords, the application is vulnerable when the attacker has access to the database, because, with brute force attacks, they can easily use their credentials.

A08-Software and Data Integrity Failures
When using plugins or libraries that are not verified on integrity, when one of these fails, we can be left with leaking unauthorized information or other problems. Therefore, data validation and sanitization are important, as doing frequent code reviews, and in the CICD pipeline having proper segregation.

A09-Security Logging and Monitoring Failures
Using Prometheus in the Kubernetes services for logging and monitoring to see if certain events go as intended.

A10-Server-Side Request Forgery (SSRF)
Attackers can attack FestivalConnect when we use data inside of the URLs or when we allow the application to read a certain data of a URL. When not properly configured, attackers can abuse this and modify or access daa. Therefore, input validation and safe libraries.

(Nir, 2023)

Microsoft's Security Engineering

- Perform Penetration Testing
- Use Approved Tools
- Perform Static Analysis Security Testing (SAST)

(What are the Microsoft SDL practices?, n.d.)

Requirements

Data Protection

- Encrypt private data, that is relevant to regulations of GDRP, and for Personal Identifiable Information (PII).
- Make sure that sensitive data is securely stored, to have secure access, create, and store.
- All the data that is stored on the client side (browser) does not have sensitive data.

Password Security:

- Input validation that forces a strong password, with a minimum length of 8, special characters requirements, and a maximum amount of characters.
- Confirm to have an encrypted password and save the password in a safe place.

Input and Output Data Entry

- Verify that users only have access to the functions that they are only meant to do.
- Verify that the tools that are being used for the secure transit of requests are approved.
- The layers between the communication are all authenticated.
- Verify that the data is validated against an input validation, to have a specific format for example for an e-mail address.
- Have a rate limit to prevent brute force attacks.

Error, Auditing, Logging handling

- Confirm that A standard format for the error and logging is used across the system.
- Make sure that the logging and errors do not include sensitive data, which can exceed GDPR.

Configuration

- Confirm to have an application that is built and can be deployed, which is done in a secure way, which is also automated in a CI/CD pipeline.
- A step in the pipeline that covers security.
- Use containerization for application to isolate the application components from each other.

Authentication

- The authentication tools used are approved and industry standards are known as secure.
- Can be easily extended with extra authentication.

Third-Party Service

- Check that the applications used by third-party companies are trusted.
- Regularly review and update third-party services to ensure they meet your security standards.

Design

- Validating application implementations of security with the coding practices of OWASP.
- Using Azure Key Vault to store passwords, tokens, etc.

Testing

- SonarQube to do Static Analysis Security Testing.
- Trivy: Vulnerability Checks in Pipeline

Monitoring

- Prometheus and Grafana: Monitoring the deployment in azure using Kubernetes.

References

Cheng, R. (n.d.). *What Firms Should Know About Role-Based & Permission-Based Access Controls*. Retrieved from practicepanther: https://www.practicepanther.com/blog/role-based-versus-permission-based-access-controls/#:~:text=Role%2Dbased%20access%20controls%20grant,differences%2C%20benefits%2C%20and%20drawbacks.

Dabah, G. (2023, 02 07). *Privacy by Design vs Security by Design: What's the Difference?* Retrieved from piiano: https://www.piiano.com/blog/privacy-by-design-vs-security-by-design

Input Validation Cheat Sheet. (n.d.). Retrieved from cheatsheetseries.owasp: https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html

Koers, M. (2014). *Grip on Secure Software Development (SSD).*

Nath, O. (2022, 07 22). *Security By Design: What Is It and How to Do It Right?* Retrieved from spiceworks: https://www.spiceworks.com/it-security/cyber-risk-management/articles/what-is-security-by-design/

Nir, O. (2023, 08 27). *OWASP Top Ten 2023 – The Complete Guide*. Retrieved from reflectiz: https://www.reflectiz.com/blog/owasp-top-ten-2023/

Secure Software Development Lifecycle (SSDLC). (n.d.). Retrieved from snyk: https://snyk.io/learn/secure-sdlc/

vcsjones. (2018, 03 09). *TLS version support in axios?* Retrieved from stackoverflow: https://stackoverflow.com/questions/49197820/tls-version-support-in-axios

What are the Microsoft SDL practices? (n.d.). Retrieved from microsoft: https://www.microsoft.com/en-us/securityengineering/sdl/practices#practice4

Wigmore, I. (n.d.). *security by design*. Retrieved from techtarget: https://www.techtarget.com/whatis/definition/security-by-design#:~:text=Security%20by%20design%20is%20an,adherence%20to%20best%20programming%20practices.

Jacobs, L. (2024). Data Storage and GDPR Compliance Strategy: Practical Guide (Unpublished manuscript), FontysICT.