Security Design

Quantum TLC

Table of Contents

Glossary

Term	Definition
Secure Software Development Lifecycle	defines security requirements and tasks that must be considered and addressed within every system, project, or application that is created or updated to address a business need.

Introduction

This document will cover the security measurements that the tournament system is going to cover, with the included technologies that are going to be used. Moreover, this will include what is considered to be a secure website that also looks into zero-tolerance of cheating.

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 Quantum TLC 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.

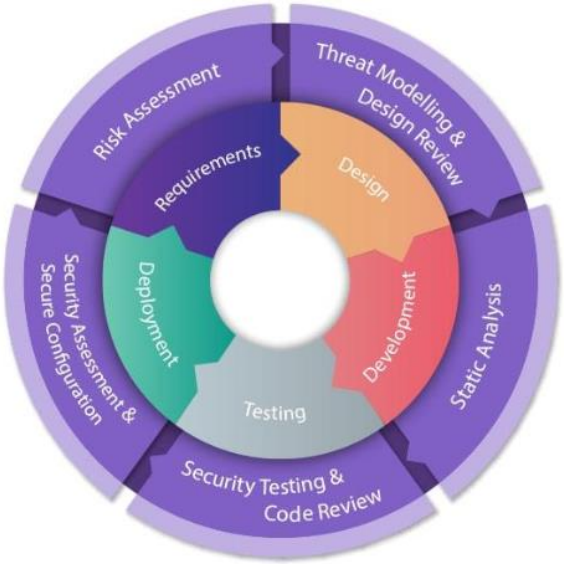

Figure 1: Secure Software Development Life Cycle (SSDLC)

- **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 Quantum TLC a more trustworthy application. (Dabah, 2023)

Analysis

The group looked into several sources to set what requirements are important for this application. The main sources are CIP, OWASP, and Microsoft's Security Engineering. Therefore, the following points are important and need to be considered.

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 Quantum TLC 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.

OWASP

With the usage of OWASP top 10, we have protection guidance of what risks, industry standards, and best practices need to be implemented for the application. This security list will show the list of most serious security vulnerabilities.

- **A01-Broken Access Control**: Important for our application to separate the functionalities that are only accessible to the admin strictly and prevent misconfigurations. So the main thing is to only give authorization to the people who are authorized to do a certain functionality.
- **A02-Cryptographic Failures:** Ensure that we have secured sensitive data, so use industry standards for encrypting passwords.
- **A03-Injection:** A vulnerability can occur when attackers try to put in SQL queries, code snippets, or commands into the form of web application forms or URLs. This can expose data, or even manipulate application behavior.
- **A04-Insecure Design:** When application design can be exploited by hackers to find information or even break the system.
- **A05-Security Misconfiguration**
- **A06-Vulnerable and Outdated Components:** There can be vulnerabilities occur when we use external sources that are not up to date.
- **A07- Identification and Authentication Failures:** Weak passwords and session IDs without a certain time period of validation it can exposes risks.
- **A08-Software and Data Integrity Failures:** When using external sources they may not be verified by the developer which can lead to serious issues.
- **A09-Security Logging and Monitoring Failures:** When developers are not notified of vulnerabilities and failures this can lead to substantial damage.
- **A10-Server-Side Request Forgery (SSRF):** This can occur when an attacker can manipulate the input field or parameters within the application to trick the server to gain sensitive data and other resources.

(Nir, 2023)

Microsoft’s Security Engineering

- Making use of approved tools (verified companies that are reliable)
- Define Security Requirements: The team will continually look at SDLC and update and reflect upon what security requirements are important to prioritize during a sprint.
- Perform Static Analysis Security Testing (SAST)
- Perform Threat Modeling

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

Requirements

Password Security

- Verify that the password is properly encrypted and stored in a safe place.
- The password needs to follow certain standards that need to be eight characters long, with a specific limit, and it is required to have a special character inside of it.

Authentication

- The application needs to make use of an authentication mechanism that is known to be secure and long-lived supported.
- All the communication between all the components needs to have verification that the user who wants to access the functions or data is authorized to.
- The application authentication system needs to be easily extended with improved authentication practices.

Data Protection

- Handling sensitive information needs to comply with the GDPR rules, with an actual look at privacy for personal data also known by the term personally identifiable information (PII).
- Verify that on the client side (Browser), there is no sensitive data stored.
- Sensitive data is stored securely stored and accessed properly.

Secure Software Development Lifecycle (SSDLC)

- All the tasks (user stories) that the team is working on need to have a security aspect considered inside. This means that when the function needs to be implemented, security reasons are taken into account.
- Make sure each iteration of the threat modeling is applied, to consider threats and vulnerabilities that can occur in the software system during the functionalities that are planned to work on.
- Verify that the team is performing coding reviews before further deployment.
- Making sure that all the requirements and applications that are used for security are available for each developer.

Data Input and Output

- All the data needs to be verified to a certain input validation.
- Users can only do functionalities that they are authorized to do.

Error, Auditing, Logging handling

- Verify the error logs have a standard format that does not contain any unneeded information. This cannot include any sensitive information that exceed GDPR.

Configuration

- Verify that the configuration files inside of the services do not include any sensitive data such as plain passwords or connection keys.
- Verify that in the CI/CD pipeline is capable of building and deploying, in a secure way.
- Using containerization, to separate components from each other.

Third-party

- Verify that the external sources from third-party companies are trusted.
- Regularly check third-party companies' software for any updates.

Cheating

- Verify that regulations regarding the set rules are being punished when not adhered.

Design

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

Testing

- Regular coding reviews, critically looking at potential threads in the application.
- Snyck, finding security vulnerabilities inside of the front end.
- SonarQube, analyzing the code and finding vulnerabilities inside of the back end.
- Audit Checks (Dependency check).
- Trivy for scanning vulnerabilities.

Monitoring

- Prometheus and Grafana: Monitoring the container environment in 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

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/

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.