UniChat – Guideline for Secure Development Lifecycle and Security Activities

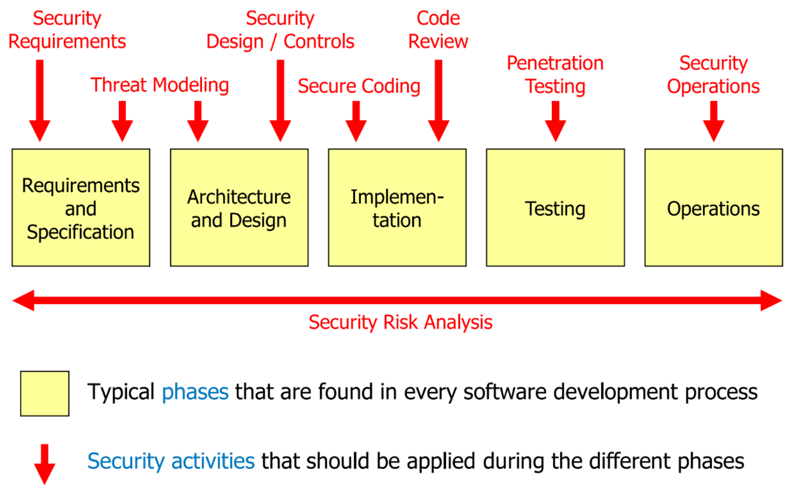
** This document is not finalized due to time constraints **

This document provides a comprehensive guideline for the team behind the UniChat desktop application. It emphasizes the importance of security throughout the software development lifecycle (SDLC) and outlines various security activities to be performed during different phases of the development process.

Secure Software Development Lifecycle (SSDLC) Overview

Security must be an integral part of the entire software development process, not just an afterthought. Each sprint comes with new tasks, which should be accompanied by the question: "How can we implement this functionality securely?" Each sprint ends with an assessment and documentation of the results in terms of security. This assessment and documentation serve as a reference for the quality of the software in terms of security.

This is how a classical software develeopment lifecycle looks like:


Security Requirements
Threat Modeling
Security Design / Controls
Secure Coding
Code Review
Penetration Testing
Security Operations
Requirements and Specification
Architecture and Design
Implemen-tation
Testing
Operations
Security Risk Analysis
Typical phases that are found in every software development process
Security activities that should be applied during the different phases

Overview of the Classical SDL Integrated with Security Activities

1. **Requirements Analysis**
2. **Design**
3. **Implementation (Coding)**
4. **Testing**
5. **Deployment**
6. **Maintenance**

Relevant Definitions

An application is considered "secure" if Confidentiality, Integrity, and Availability (CIA) are satisfied to a reasonable degree:

- **Confidentiality**: Protect sensitive data from unauthorized read access.
    - Example: Contents from chats, user credentials, and information.
- **Integrity**: Protect data and systems from unauthorized modification.
    - Example: Protect access to chats and their contents.
- **Availability**: Ensure information is available when needed.
    - Example: UniChat should be operational and accessible at any time, 24/7.

In terms of security goals, Confidentiality and Integrity are prioritized higher in our case.

Security Activities Throughout the SDLC

Security activities should be integrated into each phase of the SDLC to ensure a robust security posture for the UniChat application.

1. Requirements Analysis

Security Requirements

Defining security requirements is the first security-related activity. This is based on the functional requirements and using (generic) checklists. The checklists provide common and basic security requirements.

Useful checklists that may help to specify security requirements and controls are listed in the Appendix.

Action Points:

- Define security requirements alongside functional requirements.
- Use checklists to ensure comprehensive security coverage.
- Document all security requirements clearly.

2. Design

Threat Modeling

The purpose of Threat Modeling is to identify security design flaws or conceptual security problems. This is based on the security requirements and controls that have already been defined.

Action Points:

- Identify possible threats against the system.
- Identify vulnerabilities in the system design based on these threats.
- Document additional security requirements from the identified vulnerabilities.

Steps for Threat Modeling:

1. Express the business goal in a few sentences.
2. Driven by the business goal, identify a few security goals.
3. Think like an attacker and identify potential attack vectors.

3. Implementation (Coding)

Secure Coding

This process is about implementing the source code in a secure way and being "careful" when writing code. No security bugs should be introduced during the actual implementation.

Action Points:

- Follow secure coding guidelines and best practices.
- Regularly review and update coding standards to address new security threats.
- Conduct training sessions for developers on secure coding practices.

4. Testing

Code Review

The goal of this process is to detect security bugs by inspecting the code and searching for security problems.

Penetration Testing

Penetration testing involves simulating attacks on your system to identify and fix security vulnerabilities.

Action Points:

- Conduct regular code reviews to identify and fix security issues.
- Perform penetration testing to identify potential vulnerabilities.
- Document and remediate any discovered issues promptly.

5. Deployment

Security Operations

Security operations involve monitoring, managing, and responding to security incidents to ensure ongoing protection.

Action Points:

- Implement security monitoring tools to detect and respond to security incidents.
- Establish incident response procedures and train the team on them.
- Regularly update and patch the system to address new vulnerabilities.

6. Maintenance

Ongoing Security Activities

Maintenance involves ongoing security activities to ensure the application remains secure over time.

Action Points:

- Continuously monitor the system for security threats.
- Regularly update security controls and policies.
- Conduct periodic security audits and assessments.

Notes

Current Phase: Implementation

We are currently in the "Implementation" phase, meaning each sprint can be followed by the following "Unified Process (UP)" pattern:

- Security requirements are defined simultaneously with the "normal" requirements.
- Threat Modeling is performed in parallel with or prior to security requirements.

General Guidelines for Secure Software Development

Fundamental Security Principles

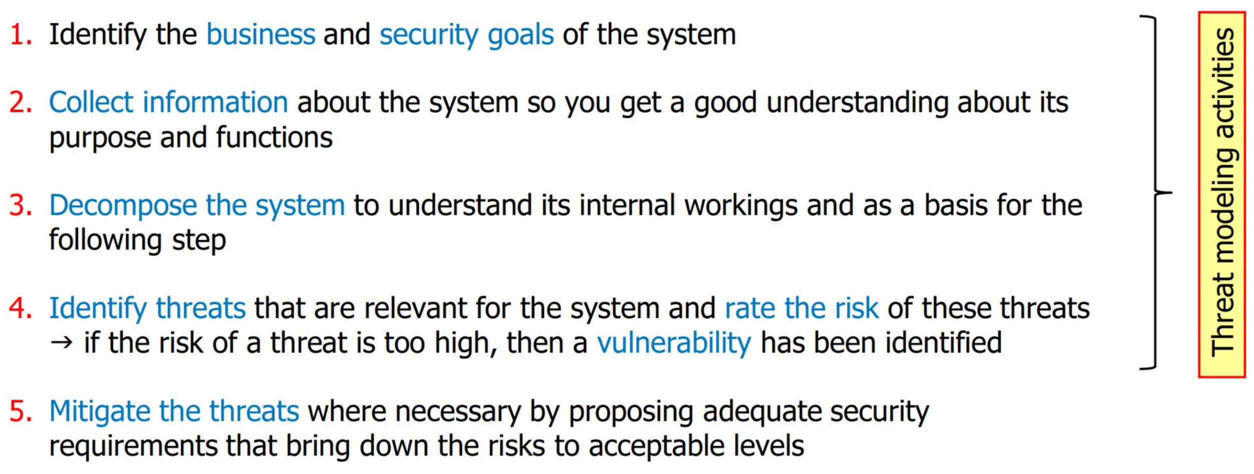
- Security is determined by the weakest component.
  - "Identifying the weak link is usually done by performing threat modeling and penetration tests in combination with risk analysis."
- New functionality introduces new threats.
  - Functional requirements lead to security requirements (based on a checklist or functional aspects of the system) and are elaborated further during later sprints.

Strategy

- The project has reached a certain stage, prompting Threat Modeling and Security Requirement definition.
- Based on these, Security Design / Controls activities follow to determine how to implement the Security Requirements concretely.

Threat Modeling for Our Case

Threat Modeling is an extremely valuable but also time-consuming assessment.
Following are the 5 steps to Microsoft's threat modeling process:



1. Identify the business and security goals of the system
2. Collect information about the system so you get a good understanding about its purpose and functions
3. Decompose the system to understand its internal workings and as a basis for the following step
4. Identify threats that are relevant for the system and rate the risk of these threats
   → if the risk of a threat is too high, then a vulnerability has been identified
5. Mitigate the threats where necessary by proposing adequate security requirements that bring down the risks to acceptable levels

As already mentione; The purpose of Threat Modeling is to **identify security design flaws** / conceptual security problems. Following is an uncomplete list of likely threats for the UniChat Desktop App:

- **Spoofing**: Use certificate pinning.
- **Tampering**: Implement hashes to check the integrity of messages and data sent.
- **Repudiation**: Implement logging and digital signatures.
- **Information Disclosure**: Encrypt sensitive data at rest and in transit.
- **Denial of Service (DoS)**: Implement rate limiting and resource management.
- **Elevation of Privilege**: Ensure proper access control and least privilege.

Define Security Requirements and Threat Modeling

Security requirements must be defined early in the development lifecycle, often using a generic checklist or based on the system's functional aspects.

Develop Security Design / Controls

Develop specific security measures to secure the system adequately.

Secure Coding & Code Review

Start with the question: "How can I design the code safely?" End with the question: "Have I ensured the code is secure?"

Business Goals

UniChat unites various messenger apps in one and allows the user to communicate seamlessly across different platforms.

Security Goals

The goal of penetration/security testing is to verify that the specified security requirements are fulfilled and implemented. Security requirements lay the basis for security controls.

This document serves as a comprehensive guide for developers working on the UniChat project to ensure that security is integrated into every phase of the software development lifecycle.

Appendix

Links for useful checklists:

- OWASP Web Application Security Requirements
- SANS Securing Web Applications
- Microsoft Security Development Lifecycle
- Web Developer Security Checklist
- Probely Security Checklist
- Security for Startups Controls Checklist