

LECTURE 16


Software requirements Engineering

Threat Modeling and Security-Critical System Requirements

Instructor:

Dr. Natalia Chaudhry,

Assistant Professor, PUCIT, University of the Punjab, Lahore.

- 
- Threat modeling is a structured approach to identifying, assessing, and mitigating potential threats and vulnerabilities within a system.
 - It involves systematically analyzing a system's components, data flows, dependencies, and potential attack vectors to anticipate and prevent security breaches.

Let's consider a scenario of developing a mobile banking application for a financial institution. We'll walk through the steps of threat modeling for this system.

Step 1: System Understanding

- Identify the components of the mobile banking application: user interface, client-side application, server-side application, databases, external APIs (e.g., for payment processing), and network infrastructure.

Step 2: Threat Identification

- Threat: Unauthorized Access to User Accounts
 - Attack Vector: Weak Authentication Mechanism
 - Vulnerability: Insufficient password complexity requirements or lack of multi-factor authentication.
- Threat: Data Interception
 - Attack Vector: Man-in-the-Middle (MitM) Attack
 - Vulnerability: Insecure transmission of sensitive data over unencrypted channels.
- Threat: Data Tampering
 - Attack Vector: Client-Side Injection Attacks
 - Vulnerability: Lack of input validation and sanitization, allowing attackers to manipulate data sent from the client to the server.

Step 3: Vulnerability Analysis

- Analyze the potential impact of each threat:
 - Unauthorized access to user accounts could lead to financial loss, identity theft, and damage to the bank's reputation.
 - Data interception could result in the exposure of sensitive customer information, such as account numbers and transaction details.
 - Data tampering could lead to unauthorized transactions, account manipulation, or the injection of malicious code into the application.

Step 4: Risk Assessment

- Assess the likelihood and impact of each threat:
 - High likelihood and high impact: Unauthorized Access to User Accounts
 - Medium likelihood and high impact: Data Interception
 - Low likelihood and medium impact: Data Tampering

Step 5: Mitigation Strategies


- Implement strong authentication mechanisms, such as biometric authentication or one-time passwords, to prevent unauthorized access to user accounts.
- Encrypt sensitive data transmitted between the mobile application and the server using secure protocols like HTTPS.
- Implement input validation and sanitization to prevent client-side injection attacks, such as SQL injection or cross-site scripting (XSS).

Step 6: Validation and Iteration

- Validate the effectiveness of mitigation strategies through security testing, penetration testing, and code reviews.
- Iterate on the threat model based on feedback from security assessments and updates to the system architecture or functionality.



Security-critical system requirements




Security-critical system requirements are specifications that define the security properties and capabilities that a system must possess to protect against threats and vulnerabilities.

Key considerations when defining security-critical system requirements include:

Confidentiality: Ensuring that sensitive information is only accessible to authorized users and protected from unauthorized access or disclosure.

Integrity: Guaranteeing the accuracy and trustworthiness of data by preventing unauthorized modifications or tampering.

Availability: Ensuring that the system and its resources remain accessible and operational to authorized users, even in the face of attacks or failures.



Authentication and Authorization: Verifying the identities of users and determining their permissions and privileges within the system to control access to resources and functionalities.

Auditability and Accountability: Enabling the monitoring, logging, and auditing of system activities to trace security-related events and hold users accountable for their actions.

Resilience and Recovery: Implementing mechanisms to detect and respond to security incidents, as well as facilitating the recovery of the system to a secure state following a breach or disruption.

Confidentiality


In the context of our online banking application, confidentiality ensures that sensitive financial information, such as account balances, transaction history, and personal details, is kept secret and only accessible to authorized individuals.

Authentication

Authentication verifies the identity of users attempting to access the online banking application. It ensures that only legitimate users with valid credentials are granted access to their accounts and financial information.

Authorization

Authorization determines what actions or operations authenticated users are allowed to perform within the online banking application. It specifies the permissions or privileges granted to users based on their role or level of access.



Let's consider a real-life scenario involving the development of a secure e-commerce platform for a retail company. Company wants to develop an e-commerce platform to expand its online presence and reach a wider customer base. The platform will allow customers to browse products, make purchases, and manage their accounts online.

1. Confidentiality:

Requirement: Customer personal and payment information must be encrypted during transmission and storage to prevent unauthorized access.

Rationale: Protecting confidentiality ensures that sensitive customer data remains confidential and is not accessible to unauthorized parties, reducing the risk of data breaches and identity theft.

2. Integrity:

Requirement: Data integrity checks must be performed to ensure that product information, prices, and order details are not tampered with during transmission or processing.

Rationale: Maintaining data integrity ensures the accuracy and trustworthiness of information presented to customers, reducing the risk of fraudulent activities and customer disputes.

3. Availability:

Requirement: The e-commerce platform must be highly available, with minimal downtime and fast response times, even during peak traffic periods or in the event of DDoS attacks.

Rationale: Ensuring availability is critical to providing a seamless shopping experience for customers and maintaining business continuity, even in the face of malicious attacks or technical failures.

4. Authentication and Authorization:

Requirement: Users must authenticate themselves using strong authentication mechanisms, such as username/password combined with multi-factor authentication (e.g., OTP), before accessing sensitive account information or making purchases.

Rationale: Strong authentication helps verify the identities of users and prevent unauthorized access to customer accounts, reducing the risk of account takeover and fraud.

5. Auditability and Accountability:

Requirement: All user activities, including login attempts, order placements, and account modifications, must be logged and auditable to trace security-related events and hold users accountable for their actions.

Rationale: Maintaining audit logs enables the detection and investigation of security incidents, as well as compliance with regulatory requirements, enhancing transparency and accountability.

6. Resilience and Recovery:

Requirement: The e-commerce platform must have robust security measures in place to detect and respond to security incidents, such as intrusion detection systems (IDS), intrusion prevention systems (IPS), and automated failover mechanisms.

Rationale: Implementing resilience and recovery mechanisms helps minimize the impact of security breaches and quickly restore the platform to a secure state, reducing downtime and mitigating financial losses.



Open Research Problems in Requirements Engineering

Requirements Elicitation and Prioritization:


Challenges: Eliciting requirements from diverse stakeholders with varying backgrounds, interests, and levels of domain expertise can lead to conflicting viewpoints and incomplete or ambiguous requirements. Prioritizing requirements effectively requires balancing competing interests and constraints, such as cost, schedule, and stakeholder priorities.

Grey Areas: Techniques for effectively eliciting requirements from non-technical stakeholders, such as end-users or domain experts, may require further exploration. Additionally, methods for prioritizing requirements in dynamic and uncertain environments, such as Agile development, may need refinement.

Requirements Validation and Verification:

Challenges: Validating requirements to ensure they meet stakeholder expectations and system constraints requires identifying inconsistencies, ambiguities, and conflicts early in the development process. However, traditional validation techniques may be time-consuming and labor-intensive.

- *Grey Areas:* Developing automated techniques for requirements validation, such as natural language processing (NLP) and machine learning (ML) approaches, is an emerging area of research.
- NLP techniques can be used to analyze natural language requirements documents written in English or other human languages. NLP techniques for requirements validation may involve parsing requirements documents to extract key information, identifying linguistic patterns or structures indicative of requirements errors or inconsistencies, and performing semantic analysis to understand the meaning and context of requirements statements.

- 
- In the context of requirements validation, ML approaches can be used to train models on annotated datasets of requirements documents and their associated quality attributes.
 - ML models for requirements validation may include classification models that classify requirements statements as correct or incorrect based on predefined criteria, regression models that predict the likelihood of requirements defects or ambiguities, and clustering models that group similar requirements together for analysis.

Requirements Engineering for Emerging Technologies:

Challenges: Emerging technologies such as artificial intelligence (AI), blockchain, and the Internet of Things (IoT) introduce unique requirements engineering challenges due to their complexity, uncertainty, and rapid evolution.

Grey Areas: Research is needed to explore how requirements engineering practices can adapt to address the specific needs and constraints of emerging technologies. This includes investigating techniques for eliciting, modeling, validating, and managing requirements for AI-driven systems, blockchain applications, and IoT devices.

Requirements Engineering in Agile and DevOps Environments:

Challenges: Agile and DevOps methodologies emphasize rapid development cycles, continuous integration, and customer collaboration, posing challenges for traditional requirements engineering practices.

Grey Areas: Research is needed to investigate how requirements engineering can be effectively integrated into Agile and DevOps processes. This includes exploring techniques for iterative requirements elicitation, lightweight documentation, and automated testing to support rapid development cycles while ensuring stakeholder satisfaction and system quality.