100 DAY CHALLENGE

DAY 4 - CHALLENGE

EXPLORE THE FRAMEWORK OF TODO MODELLING

Threat modeling for a ToDo application involves identifying potential security threats, vulnerabilities, and risks related to the application. This process helps in designing secure systems by proactively addressing security concerns. Here's a structured framework for threat modeling a ToDo application:

1. Define Objectives and Scope

- **Objectives**: Determine what you aim to achieve with the threat modeling. For a ToDo application, objectives might include securing user data, preventing unauthorized access, and ensuring data integrity.
- **Scope**: Define the boundaries of the threat model. This could include the entire application, specific modules (e.g., authentication), or interactions with external systems (e.g., third-party APIs).

2. Identify Assets

- **User Data**: Personal information, to-do items, and any other user-related data.
- **Authentication Data**: User credentials, tokens, or session identifiers.
- **Application Code**: Source code and configuration files.
- **Infrastructure**: Servers, databases, and network components.
- **Third-Party Services**: APIs or libraries used in the application.

3. Create an Architecture Diagram

- **System Components**: Map out all components such as the user interface, server-side logic, database, APIs, and third-party services.
- **Data Flow**: Illustrate how data moves between components. For instance, how user input is processed and stored.
- **Trust Boundaries**: Identify boundaries where different levels of trust exist, such as between user input and the application server.

4. Identify Threats

Use a threat modeling methodology such as STRIDE or DREAD to identify potential threats:

- **STRIDE**:
    - **Spoofing**: Unauthorized users accessing the system.
    - **Tampering**: Modification of data in transit or at rest.
    - **Repudiation**: Users denying actions (e.g., deleting to-do items).
    - **Information Disclosure**: Exposure of sensitive data.
    - **Denial of Service**: Overloading the system to make it unavailable.
    - **Elevation of Privilege**: Unauthorized access to higher privilege levels.
- **DREAD**:
    - **Damage Potential**: How harmful a threat could be.
    - **Reproducibility**: How easy it is to replicate the threat.
    - **Exploitability**: How easy it is to exploit the threat.
    - **Affected Users**: How many users are affected.
    - **Discoverability**: How easy it is to discover the threat.

5. Assess Risks

- **Likelihood**: Evaluate the probability of each threat occurring.
- **Impact**: Assess the potential impact of each threat if it were to occur.
- **Risk Level**: Combine likelihood and impact to determine the risk level (e.g., High, Medium, Low).

6. Develop Mitigation Strategies

- **Access Control**: Implement strong authentication and authorization mechanisms.
- **Data Encryption**: Use encryption for sensitive data at rest and in transit.
- **Input Validation**: Validate and sanitize user inputs to prevent injection attacks.
- **Logging and Monitoring**: Implement logging and monitoring to detect and respond to threats.
- **Regular Updates**: Keep software and dependencies up-to-date to address known vulnerabilities.

7. Review and Update

- **Regular Reviews**: Periodically review and update the threat model to address new threats and changes in the application.
- **Feedback Loop**: Incorporate feedback from security assessments and incident responses to improve the threat model.

- **Example Framework for a ToDo Application**

1. **Define Objectives and Scope**
    - Ensure user data confidentiality, integrity, and availability.
2. **Identify Assets**

- User credentials, to-do items, and backend database.

3. **Create an Architecture Diagram**
   - Diagram includes user interface, API server, database, and external APIs.
4. **Identify Threats**
   - Spoofing: Unauthorized users gaining access.
   - Tampering: Alteration of to-do items.
   - Information Disclosure: Unauthorized viewing of private to-dos.
5. **Assess Risks**
   - Spoofing: High likelihood, high impact (High Risk).
   - Tampering: Medium likelihood, high impact (Medium Risk).
6. **Develop Mitigation Strategies**
   - Use multi-factor authentication (MFA).
   - Implement data encryption and validation.
   - Regularly audit and monitor access logs.
7. **Review and Update**
   - Schedule regular threat model reviews and update based on new findings or changes.