STRIDE Security Threat Modeling Report

Project Overview

The project involves developing a client management system using Flask for the web application and PocketBase for user management and backend data storage. The system supports both admin and client functionalities, with a focus on security and user data management.

---

1. Spoofing

- **Threat:** Unauthorized users may impersonate legitimate clients or admins to gain access to sensitive data or perform malicious actions.
- **Identified Risks:**
    - Unauthorized users might attempt to log in by exploiting weak or predictable passwords through brute force.
- **Potential Attack Scenarios:**
    - Attackers could use brute-force methods to guess weak passwords and gain access to client or admin accounts.
- **Potential Impact:**
    - Attackers could gain access to sensitive client or management data, or perform malicious operations.
- **Existing Mitigations:**
    - Password policies mandate a minimum length of 10 characters.
- **Risk Level:** Medium
- **Recommended Mitigations:**
    - Implement Multi-Factor Authentication (MFA).
    - Apply anti-bot strategies after multiple failed login attempts.
    - Regularly conduct security audits and penetration tests.

---

2. Tampering

- **Threat:** Attackers could tamper with data either in transit or at rest, leading to unauthorized modifications of client information, project files, or system configurations.

- **Identified Risks:**
    - Data could be intercepted and altered during transmission, or the database on the PocketBase server could be compromised.
- **Potential Attack Scenarios:**
    - Man-in-the-Middle (MITM) attacks could modify data packets during transmission, changing client information or project files.
    - The unencrypted database on the PocketBase server could be tampered with, causing data loss or corruption.
- **Potential Impact:**
    - Client information might become inconsistent or invalid.
    - Data loss could disrupt the website's operation.
- **Existing Mitigations:**
    - The use of API-based transmission.
    - PocketBase's API prevents remote threats and SQL injection.
- **Risk Level:**
    - Data Transmission: Medium
    - Server Database: High
- **Recommended Mitigations:**
    - Implement HTTPS encryption for all data transmissions and ensure data integrity with checks like HMAC.
    - Place the PocketBase server on a secure, protected server and regularly encrypt the database using different methods.

---

3. Repudiation

- **Threat:** Users could deny performing specific actions, such as modifying or deleting files, leading to accountability issues.
- **Identified Risks:**
    - Users might manipulate logs stored on the PocketBase server to deny performing certain actions.
- **Potential Attack Scenarios:**

    - Users could perform operations and then modify the logs to claim that they never executed those actions.
- **Potential Impact:**
    - Accountability becomes difficult, which could impact legal or compliance requirements.
- **Existing Mitigations:**
    - PocketBase's API prevents remote threats and SQL injection.
- **Risk Level:** High
- **Recommended Mitigations:**
    - Store the PocketBase server on a secure server.
    - Implement tamper-resistant logging mechanisms, such as Write Once Read Many (WORM) storage.

---

4. Information Disclosure

- **Threat:** Unauthorized users could access sensitive information, such as client details, project files, or admin data.
- **Identified Risks:**
    - Data might be intercepted during transmission, leading to the exposure of sensitive information.
- **Potential Attack Scenarios:**
    - Attackers might intercept data packets during transmission, leading to the disclosure of sensitive information or project files.
- **Potential Impact:**
    - Users' privacy could be compromised, and sensitive data might be exposed.
- **Existing Mitigations:**
    - Localized deployment using a local area network (LAN).
- **Risk Level:** Low
- **Recommended Mitigations:**

- For future cloud deployments, encrypt user information during transmission and storage.
- Apply additional encryption for highly sensitive data.

---

5. Denial of Service (DoS)

- **Threat:** Attackers could flood the system with requests, causing it to become unavailable to legitimate users.
- **Identified Risks:**
  - The system could be overwhelmed by a high volume of requests, leading to server overload.
- **Potential Attack Scenarios:**
  - Attackers could send a massive number of requests to exhaust server resources, causing the system to crash or become unavailable.
- **Potential Impact:**
  - Legitimate users might be unable to access the system, disrupting business continuity.
- **Existing Mitigations:**
  - There is currently a lack of rate limiting and Web Application Firewall (WAF).
- **Risk Level:** Low
- **Recommended Mitigations:**
  - Implement rate limiting to limit the number of API requests from a single source.
  - Deploy a Web Application Firewall (WAF) to filter and block malicious traffic.

---

6. Elevation of Privilege

- **Threat:** A lower-privileged user (e.g., a client) could exploit vulnerabilities to gain unauthorized administrative privileges.
- **Identified Risks:**

    - Improper configuration of user roles and permissions might allow users to access or modify other users' resources.
    - Attackers might directly modify user settings from the backend server.
- **Potential Attack Scenarios:**
    - Users could gain access to other users' resources due to incorrect permission settings.
    - Attackers could directly modify user permissions from the server, escalating their privileges.
- **Potential Impact:**
    - Other users' resources could be exposed or altered.
    - User permissions might be incorrectly set, leading to unauthorized access.
- **Existing Mitigations:**
    - PocketBase's remote API prevents users from bypassing front-end restrictions to directly access sensitive resources.
- **Risk Level:**
    - User Permissions: Low
    - Server Configuration: High
- **Recommended Mitigations:**
    - Ensure precise configuration of user permissions on the PocketBase server to prevent privilege escalation.
    - Set up alerts for privilege escalation attempts and regularly back up and encrypt user settings files.

Conclusion

The STRIDE analysis has identified key security risks in the system's current design, specifically focusing on Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. Each identified risk has been addressed with specific mitigation strategies that align with industry best practices. The recommendations provided should be implemented to minimize these risks effectively, ensuring the security and integrity of the client management system.