# High-Level Threat Model (Bank)

**1. System Overview**

A simplified architecture for the bank’s core services might include:

1. **Customer Access Channels**
   * Web application (online banking)
   * Mobile application
   * ATM network
2. **Front-End Services**
   * Web servers and mobile APIs
3. **Application Layer**
   * Transaction processing (e.g., transfers, payments)
   * Authentication services
   * Business logic (e.g., credit checks, account management)
4. **Data Repositories**
   * Core banking database (customer records, account balances, transaction history)
   * Audit logs and security logs
5. **Integration Points**
   * External payment networks (e.g., SWIFT, card payment processors)
   * Third-party services (e.g., credit bureaus, regulatory reporting)

**2. Data Flow**

1. **Customers** initiate requests (logins, fund transfers) through the **web/mobile/ATM interfaces**.
2. **Front-End Services** handle incoming requests, perform initial security checks (e.g., session validation).
3. **Application Layer** processes transactions, authenticates users, and orchestrates business logic.
4. **Data Repositories** store and retrieve account information, transaction data, and audit records.
5. **Integration Points** allow the bank to send/receive data to/from external networks and partners.

**3. STRIDE Analysis**

**3.1 Spoofing**

* **Threats**
  + Attackers impersonating legitimate users (customer or employee) to access accounts.
  + Spoofed network endpoints or fraudulent domain names intercepting traffic.
* **Attack Vectors**
  + Phishing, credential theft, or session hijacking.
  + DNS spoofing or SSL certificate manipulation.
* **Impact**
  + Unauthorised transactions, data theft, regulatory non-compliance.
* **Potential Mitigations**
  + Strong authentication (multi-factor for both customers and employees).
  + Secure channel enforcement (TLS, certificate pinning).
  + Anti-phishing measures and user education.

**3.2 Tampering**

* **Threats**
  + Modification of in-transit transactions or records in the database.
  + Malicious changes to configuration files on critical servers.
* **Attack Vectors**
  + Network man-in-the-middle attacks.
  + Insider threats altering data directly in the system.
* **Impact**
  + Financial fraud, inaccurate records, system instability.
* **Potential Mitigations**
  + End-to-end encryption for data in transit.
  + Integrity checks (checksums, digital signatures) on critical data.
  + Strict change management with role-based access controls for database and configuration changes.

**3.3 Repudiation**

* **Threats**
  + Users or insiders performing illicit activities and denying responsibility.
  + Inadequate logging or missing audit trails making investigations difficult.
* **Attack Vectors**
  + Unlogged privileged actions.
  + Weak non-repudiation controls (e.g., no digital signatures).
* **Impact**
  + Inability to trace and attribute fraudulent transactions.
  + Compliance failures (regulatory audits, legal disputes).
* **Potential Mitigations**
  + Comprehensive logging with secure, tamper-evident storage (e.g., WORM storage).
  + Digital signatures on high-value transactions.
  + Automated monitoring and alerting on anomalous user actions.

**3.4 Information Disclosure**

* **Threats**
  + Data breaches exposing customer details or financial records.
  + Misconfigured external services or storage buckets.
* **Attack Vectors**
  + SQL injection, insecure APIs, or zero-day exploits.
  + Social engineering / spear phishing for privileged access.
* **Impact**
  + Loss of customer trust, regulatory fines (e.g., GDPR), reputational damage.
* **Potential Mitigations**
  + Hardened perimeter with WAF (Web Application Firewall) and thorough input validation.
  + Strong encryption at rest (key management, database field-level encryption).
  + Role-based access with least-privilege principles for internal users and systems.

**3.5 Denial of Service**

* **Threats**
  + Disruption of banking services (online platforms, ATMs, payment systems).
  + Attackers flooding networks or transaction processing systems.
* **Attack Vectors**
  + DDoS attacks from botnets.
  + Application-layer attacks consuming excessive resources.
* **Impact**
  + Service downtime, customer dissatisfaction, potential financial loss.
* **Potential Mitigations**
  + Distributed, resilient infrastructure (load balancers, CDNs, auto-scaling).
  + DDoS protection services (scrubbing centers, network filtering).
  + Throttling and rate-limiting on APIs, plus failover policies.

**3.6 Elevation of Privilege**

* **Threats**
  + Attackers escalating from normal user privileges to system or domain admin.
  + Lateral movement from one compromised service to another.
* **Attack Vectors**
  + Exploiting unpatched OS/application vulnerabilities.
  + Misconfiguration of access control (e.g., excessive IAM permissions).
* **Impact**
  + Full system compromise, large-scale fraud, data exfiltration.
* **Potential Mitigations**
  + Regular vulnerability scanning and patch management.
  + Network segmentation with strict trust boundaries.
  + Privilege management tools (e.g., just-in-time admin access, two-person rule for sensitive actions).

**4. Attack Trees (High-Level Example)**

**Goal**: Unauthorised Funds Transfer

1. **Obtain Valid Credentials**  
   1.1 Phishing or social engineering  
   1.2 Keylogging malware on customer’s machine  
   1.3 Insider or stolen credentials
2. **Bypass Transaction Controls**  
   2.1 Exploit API vulnerabilities  
   2.2 Intercept and modify transaction data in transit
3. **Avoid Detection**  
   3.1 Disable or evade monitoring systems  
   3.2 Alter logs to cover tracks