# Security Requirements & Threat Modeling – Comprehensive Guide with Real-Time Example

## 🔐 What Are Security Requirements?

**Security Requirements** are explicit statements outlining the safeguards an application must implement to prevent, detect, or respond to threats. These requirements align with organizational policies, industry regulations, and risk assessments.

## 🧩 Types of Security Requirements

| Category | Example Requirements |
| --- | --- |
| **Authentication** | Use Multi-Factor Authentication (MFA), strong password policies |
| **Authorization** | Role-Based Access Control (RBAC), least privilege principles |
| **Data Protection** | Encrypt data at rest and in transit, use secure hashing |
| **Input Validation** | Validate all inputs to prevent XSS, SQL Injection |
| **Audit Logging** | Track all critical user actions, failed login attempts |
| **Availability** | Implement DDoS protection, backup and recovery policies |

## 🧠 What Is Threat Modeling?

**Threat Modeling** is the structured process of identifying potential security threats and determining the appropriate mitigation strategies. It is best done during the **design phase** of the Software Development Life Cycle (SDLC).

### 🧰 Common Threat Modeling Frameworks:

* **STRIDE** – Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege
* **DREAD** – Damage, Reproducibility, Exploitability, Affected Users, Discoverability

## 🧪 STRIDE Model Explained

| Threat Type | Description | Example |
| --- | --- | --- |
| **Spoofing** | Pretending to be another user | Login using stolen credentials |
| **Tampering** | Modifying data or code | Altering cookies in the browser |
| **Repudiation** | Denying an action without traceability | Claiming a transaction was never made |
| **Information Disclosure** | Exposing sensitive information | Credit card numbers in logs |
| **Denial of Service (DoS)** | Making service unavailable | Sending large requests repeatedly |
| **Elevation of Privilege** | Gaining higher access rights | Regular user becomes admin |

## 💼 Real-Time Example: Online Banking Application

### Step 1: Security Requirements

* All users must log in with MFA
* Passwords stored using bcrypt hashing
* Transfer limits enforced by role
* Transactions must be logged and auditable

### Step 2: Threat Modeling on “Funds Transfer” Feature

| Threat | Scenario | Mitigation |
| --- | --- | --- |
| Spoofing | Attacker uses stolen credentials | Enforce MFA and device fingerprinting |
| Tampering | Change transfer amount via form manipulation | Use server-side validation and checksums |
| Information Disclosure | Displaying account balance in browser console | Avoid exposing sensitive data to client |
| DoS | Bots sending thousands of transfer requests | Implement CAPTCHA and rate-limiting |
| Privilege Escalation | User accesses admin-only transfer controls | Apply RBAC and access auditing |

## 🔁 Integration into SDLC

| SDLC Phase | Security Activity |
| --- | --- |
| Requirements | Define security policies and rules |
| Design | Perform threat modeling and architectural risk assessment |
| Implementation | Apply secure coding and static analysis |
| Testing | Perform pen testing, DAST, and security unit tests |
| Deployment | Use secure configurations and infrastructure scanning tools |
| Maintenance | Monitor logs, rotate secrets, scan for vulnerabilities regularly |

## ✅ Conclusion

Security Requirements and Threat Modeling form the **foundation of secure software development**. When implemented early, they reduce risks, lower remediation costs, and improve the trustworthiness of the application.

To stay secure in today’s fast-moving tech landscape, organizations must treat security as a **continuous process**, not a one-time checklist.

Would you like this exported as a PDF or Word document?