# Secure SDLC – Concepts and Activities with Real-Time Example

## 📘 What is Secure SDLC?

Secure Software Development Life Cycle (**Secure SDLC**) is a framework that integrates **security practices into every phase** of the traditional Software Development Life Cycle. The objective is to **minimize vulnerabilities** and **ensure security** from design to deployment.

## 🔄 Secure SDLC Phases and Key Activities

### 1. 📋 Requirements Phase

**Activities:** - Define security-specific requirements - Determine compliance needs (e.g., GDPR, HIPAA, PCI-DSS) - Identify threat models

**Example:** A healthcare application requires **HIPAA compliance**. Security requirements include encrypted communication, access control, and audit logging.

### 2. 🧠 Design Phase

**Activities:** - Perform **threat modeling** (e.g., STRIDE, DREAD) - Select secure design patterns - Define security controls (authentication, authorization, encryption)

**Example:** For a banking portal, architects implement **Role-Based Access Control (RBAC)** and enforce **data-at-rest encryption**.

### 3. 👨‍💻 Development Phase

**Activities:** - Follow secure coding standards (OWASP, SEI CERT) - Integrate **Static Application Security Testing (SAST)** tools like **SonarQube**, **Veracode** - Peer code reviews with security focus

**Example:** Developers use **input validation** to prevent SQL Injection and Cross-Site Scripting (XSS) attacks.

### 4. 🧪 Testing Phase

**Activities:** - Conduct **Dynamic Application Security Testing (DAST)** - Perform **penetration testing** - Test authentication and authorization workflows

**Example:** A QA team uses **OWASP ZAP** to scan a web application and uncovers an insecure direct object reference (IDOR).

### 5. 🚀 Deployment Phase

**Activities:** - Secure environment setup (firewalls, network segmentation, HTTPS) - Final code audit and security checklist validation - Ensure secure configuration management (e.g., Ansible, Terraform)

**Example:** Before go-live, a production server is hardened using **CIS benchmarks**, and all secrets are stored using **Vault**.

### 6. 📊 Maintenance and Monitoring Phase

**Activities:** - Apply regular patches and updates - Monitor logs and audit trails - Continuous vulnerability scanning and incident response planning

**Example:** Post-deployment, a cloud-based app is monitored using **SIEM tools like Splunk** for unusual login activities.

## 🛡️ Real-Time End-to-End Example: E-Commerce Platform

1. **Requirements:** Define PCI-DSS compliance for online payments.
2. **Design:** Threat model includes MITM attack prevention with SSL.
3. **Development:** Input sanitization to prevent SQL Injection.
4. **Testing:** Use DAST to find session fixation issues.
5. **Deployment:** Use Terraform to provision hardened cloud infra.
6. **Maintenance:** Monitor login attempts and trigger alerts.

## ✅ Benefits of Secure SDLC

* Detects vulnerabilities early (reduces cost of fixing later)
* Ensures regulatory compliance
* Builds user trust and prevents data breaches
* Enables **DevSecOps** with continuous security integration

## 🔁 Integration with DevOps

Secure SDLC fits seamlessly with DevOps workflows: - Automate SAST/DAST in CI/CD pipelines - Enforce security gates before deployment - Use IaC security policies for infrastructure

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