AI-Powered Secure SDLC Compliance Auditor

1. Title of the Project

AI-Powered Secure SDLC Compliance Auditor: An Intelligent System for Automated Security Assessment and Compliance Monitoring in Software Development Lifecycle

2. Team Information

- **Team Leader:** Tauqeer Ahmad (Student ID: 22i-2078) - Security Architecture & AI Integration
- **Security Analyst:** Muazam Ali (Student ID: 22i-1734) - Vulnerability Assessment & Risk Analysis
- **Backend Developer:** MustanSir Hussain (Student ID: 22i-1764) - System Development & Database Design
- **Frontend Developer:** Saad Mehmood (Student ID: 22i-1655) - User Interface & Dashboard Development

3. Problem Statement

Organizations struggle to maintain consistent security practices throughout their Software Development Lifecycle (SDLC), leading to security vulnerabilities, compliance violations, and increased cyber risk exposure. Traditional manual security audits are time-consuming, error-prone, and often occur too late in the development process. Current tools lack intelligent analysis capabilities and fail to provide comprehensive SDLC security assessment aligned with industry frameworks like NIST SSDF, OWASP SAMM, and ISO 27001.

The absence of automated, intelligent security compliance monitoring results in delayed vulnerability detection, inconsistent security implementation, and difficulty in demonstrating regulatory compliance. Development teams need an AI-powered solution that can continuously

monitor, assess, and guide secure software development practices across all SDLC phases while providing actionable insights for compliance adherence.

4. Objectives of the Project

- **Primary Goal:** Develop an AI-powered system that automatically audits and monitors secure SDLC compliance across all development phases
- **Automated Risk Assessment:** Implement intelligent vulnerability detection and risk scoring mechanisms
- **Compliance Monitoring:** Ensure adherence to security frameworks (NIST SSDF, OWASP SAMM, ISO 27001)
- **Real-time Feedback:** Provide continuous security guidance and recommendations to development teams
- **Dashboard Analytics:** Create comprehensive reporting and visualization for security metrics
- **Integration Capability:** Seamlessly integrate with existing development tools and CI/CD pipelines

5. Proposed Solution

System Overview

The AI-Powered Secure SDLC Compliance Auditor is an intelligent platform that leverages machine learning algorithms, natural language processing, and security knowledge bases to automatically assess, monitor, and guide secure software development practices.

Core Security Principles

- **Defense in Depth:** Multi-layered security assessment across all SDLC phases
- **Principle of Least Privilege:** Role-based access control for audit system users
- **Secure by Design:** Built-in security controls and encrypted data handling
- **Continuous Monitoring:** Real-time security posture assessment and alerting

- **Risk-Based Approach:** Intelligent prioritization based on threat severity and business impact

Key Features

- **AI-Powered Code Analysis:** Machine learning models for vulnerability detection and secure coding pattern recognition
- **Compliance Framework Mapping:** Automated mapping to NIST SSDF, OWASP SAMM, BSIMM, and ISO 27001 requirements
- **Threat Intelligence Integration:** Real-time threat feed integration for emerging vulnerability detection
- **Automated Report Generation:** Comprehensive compliance reports with remediation recommendations
- **Dashboard Visualization:** Interactive security metrics and compliance status dashboards

Threats & Countermeasures

Threat	Impact	Countermeasure
Data Breach of Audit Logs	High	AES-256 encryption, secure key management, access logging
Unauthorized System Access	High	Multi-factor authentication, role-based access control, session management
AI Model Poisoning	Medium	Model validation, secure training data pipelines, anomaly detection
False Positive/Negative Results	Medium	Continuous model training, human-in-the-loop validation, confidence scoring

6. Methodology

Secure Development Approach

Following Secure-SDLC methodology with integrated security activities:

Phase 1: Planning & Requirements (Weeks 1-2)

- Threat modeling and security requirements analysis
- Risk assessment and compliance framework selection
- Architecture security review and design principles definition

Phase 2: Design & Architecture (Weeks 3-4)

- Secure architecture design with security controls mapping
- AI model architecture design and security considerations
- Database security design and encryption strategy implementation

Phase 3: Implementation (Weeks 5-8)

- Secure coding practices with automated code review integration
- AI model development with bias detection and validation
- Security testing integration (SAST, DAST, dependency scanning)

Phase 4: Testing & Validation (Weeks 9-10)

- Comprehensive security testing including penetration testing
- AI model accuracy validation and false positive/negative analysis
- User acceptance testing with security focus

Phase 5: Deployment & Monitoring (Weeks 11-12)

- Secure deployment with infrastructure security hardening
- Continuous monitoring setup and alerting configuration
- Security incident response procedures implementation

7. Tools and Technologies

Programming Languages & Frameworks

- **Backend:** Python (Flask/Django), Node.js for API development
- **Frontend:** React.js with TypeScript for dashboard development
- **AI/ML:** TensorFlow, PyTorch, scikit-learn for machine learning models
- **Database:** PostgreSQL with encryption, Redis for caching

Security Tools Integration

- **Static Analysis:** SonarQube, Checkmarx, Veracode API integration
- **Dynamic Analysis:** OWASP ZAP, Burp Suite API integration
- **Container Security:** Docker Bench Security, Trivy scanner
- **Infrastructure:** Terraform for secure infrastructure as code

Development & Deployment

- **Version Control:** Git with signed commits and branch protection
- **CI/CD:** Jenkins/GitHub Actions with security pipeline integration
- **Containerization:** Docker with security scanning and hardened images
- **Cloud Platform:** AWS/Azure with security best practices implementation

8. Expected Deliverables

Technical Deliverables

- **Functional Prototype:** Complete AI-powered SDLC compliance auditor system
- **AI Models:** Trained machine learning models for vulnerability detection and compliance assessment
- **Web Dashboard:** Interactive compliance monitoring and reporting interface
- **API Documentation:** Comprehensive REST API documentation for system integration
- **Database Schema:** Secure database design with audit trail capabilities

Documentation Deliverables

- **Security Architecture Document:** Detailed system security design and controls

- **User Manual:** Complete guide for system operation and configuration
- **Compliance Mapping Document:** Framework alignment documentation (NIST SSDF, OWASP SAMM)
- **Testing Report:** Security testing results and vulnerability assessment findings
- **Deployment Guide:** Secure installation and configuration procedures

Presentation Materials

- **Final Presentation:** Comprehensive project demonstration and results analysis
- **Security Demo:** Live demonstration of vulnerability detection and compliance assessment
- **Performance Metrics:** System accuracy, false positive rates, and compliance coverage analysis

9. Timeline

Week	Phase	Key Milestones	Deliverables
1-2	Planning	Requirements gathering, threat modeling	Security requirements document, project plan
3-4	Design	Architecture design, AI model design	System architecture, database schema
5-6	Core Development	Backend API development, AI model implementation	Core system prototype, initial AI models
7-8	Integration	Frontend development, tool integrations	Complete system integration, dashboard
9-10	Testing	Security testing, model validation	Testing report, validated AI models
11-12	Deployment	System deployment, documentation	Final system, complete documentation

Critical Milestones

- **Week 2:** Security requirements and architecture approval
- **Week 6:** Core AI models and backend API completion
- **Week 8:** Full system integration and initial testing
- **Week 10:** Security testing completion and vulnerability remediation
- **Week 12:** Final system deployment and project presentation

10. References

Security Frameworks & Standards

- NIST SP 800-218: Secure Software Development Framework (SSDF) v1.1
- OWASP Software Assurance Maturity Model (SAMM) v2.0
- ISO/IEC 27001:2022 Information Security Management Systems
- Building Security In Maturity Model (BSIMM) v13

Academic & Industry Resources

- McGraw, G. (2022). "AI Security: Challenges and Solutions." IEEE Security & Privacy Magazine
- OWASP Top 10 2021: A10-2021 Server-Side Request Forgery and security risks
- NIST Cybersecurity Framework 2.0: Artificial Intelligence Risk Management
- Saltzer, J.H. & Schroeder, M.D. "The Protection of Information in Computer Systems"

Technical Documentation

- TensorFlow Security Guidelines for ML Model Development
- OWASP AI Security and Privacy Guide v1.0
- Cloud Security Alliance: Security Guidance for Critical Areas of Focus in Cloud Computing v4.0
- SANS Secure Coding Practices Quick Reference Guide