# AutoSecure: AI-Powered GitHub Agent for Automated Java Vulnerability Remediation

## Overview

Legacy Java applications in financial institutions often carry outdated dependencies, deprecated API usage, and unresolved security vulnerabilities — exposing them to compliance risks, technical debt, and security breaches.  
  
AutoSecure is an AI-integrated GitHub agent that scans, remediates, and upgrades Java projects. It proactively:  
- Identifies security vulnerabilities (CVEs)  
- Replaces deprecated methods  
- Updates frameworks like Spring Boot and libraries  
- Validates refactored code  
- Creates pull requests for safe, secure codebase modernization  
  
By combining static analysis, AI-based refactoring, and DevSecOps workflows, AutoSecure helps teams reduce vulnerability exposure, accelerate modernization, and avoid regressions — a critical need for highly regulated sectors like banking.

## Problem Space

- Security Debt: Teams defer upgrades fearing functionality breakage.  
- Compliance Pressure: Financial services face regulatory mandates for secure software.  
- Developer Load: Manually reviewing and replacing deprecated code across large codebases is time-consuming and error-prone.  
- Missed CVEs: Many known vulnerabilities persist due to inconsistent upgrade cycles.

## Solution Summary

AutoSecure introduces a secure, automated remediation pipeline by:  
1. Scanning codebases for vulnerabilities (CVEs) and deprecated Java/Spring APIs.  
2. Mapping deprecated methods to safe alternatives using LLM (GPT/OpenAI).  
3. Updating build configurations (Maven/Gradle) with secure versions.  
4. Validating through test execution and safe refactor previews.  
5. Creating Pull Requests with changelogs and impact summaries.  
  
This transforms vulnerability remediation into a low-friction, repeatable, AI-assisted task — making modern DevSecOps truly continuous.

## Core Features

1. Vulnerability & Deprecation Detection  
- SBOM-based scan (via OWASP Dependency-Check, snyk, or Maven plugins).  
- Deprecated Java/Spring APIs identified through AST analysis or GPT-based search.  
  
2. Code Upgrade Engine  
- Uses LLMs (e.g., GPT-4, GitHub Copilot) to refactor deprecated methods.  
- Context-aware replacement based on method signatures and comments.  
  
3. Safe Validation Layer  
- Compilation check using Maven/Gradle CLI  
- Optionally runs unit/integration tests (JUnit/TestNG)  
- Logs changelog of updates and regression check outcomes  
  
4. GitHub Integration  
- Creates secure PR with updated code, vulnerability fixes, and confidence levels  
- Triggerable via GitHub Action or CLI

## Tech Stack

| Layer | Tools |  
|--------------|---------------------------------------|  
| Scanning | OWASP Dependency Check, Snyk CLI |  
| Code Parsing | JavaParser, Refaster, AST tools |  
| AI Suggestions| GPT-4 / Copilot API |  
| Build & Test | Maven, Gradle, JUnit |  
| Integration | GitHub API, GitHub Actions |  
| Optional UI | React/CLI Dashboard |

## Target Impact for Danske Bank

| Impact Area | Value |  
|------------------|------------------------------------------------------------|  
| Security | Proactively remediates known vulnerabilities before audits |  
| Compliance | Supports adherence to ISO/PCI/SOX standards |  
| Efficiency | Saves ~80% dev effort on upgrades |  
| Risk Reduction | Prevents regressions through testing |  
| Innovation | Introduces AI into code maintenance pipelines |

## Comparison with Traditional Remediation

| Criteria | Manual Approach | AutoSecure |  
|-----------------|------------------------------|-----------------------------|  
| Accuracy | Human-dependent | CVE + AI-backed |  
| Time | Days per upgrade | Minutes per scan |  
| Refactoring | Risk of missed APIs | LLM-driven code replacement |  
| Security | Reactive | Proactive & Continuous |  
| Regression | Manual testing | Auto validation |

## Implementation Plan (Hackathon Scope)

\*\*Phase 1: MVP\*\*  
- Input: GitHub repo URL  
- Actions: Scan → Update dependencies → Detect deprecated code  
- Output: GitHub PR with updated code + report  
  
\*\*Phase 2: AI-assisted Refactor (LLM Integration)\*\*  
- Input: Deprecated method  
- Action: GPT suggests equivalent replacement  
- Output: Auto code change with fallback for manual review

## Optional Enhancements

- Chatbot interface: “Ask Agent” about deprecated usage  
- Slack alerting: Notify when vulnerability is fixed via PR  
- Integration with Jenkins pipeline

## Conclusion

AutoSecure combines AI, DevSecOps, and GitHub workflows to solve a real, pressing problem in enterprise software security. Especially in regulated environments like Danske Bank, this agent reduces risk, saves time, and ensures Java applications are continuously modernized — without breaking functionality.  
  
Tagline: “Upgrade. Secure. Repeat. — With AutoSecure.”