Internship Progress Report

Name: Saiprakash Bollam
Internship Role: Research Intern
Duration: 6th November 2025 – 19th November 2025
Organization: Computer Science Department, Binghamton University
Supervisor: Zerksis Umrigar
Email: umrigar@binghamton.edu

---

1. Introduction

This week’s work focused on conducting a comprehensive **security audit** of the Smart Contact Manager application after upgrading to **Spring Boot 3.5.0** and Java 21.The goal was to identify deprecated methods, insecure patterns, configuration vulnerabilities, and potential cyber threats that could affect system confidentiality, integrity, and availability.

AI tools played a critical role in **automating codebase analysis**, generating detailed reports, and proposing targeted fixes.

---

2. Objectives

- Identify deprecated and insecure code patterns.
- Detect vulnerabilities related to authentication, authorization, file uploads, logging, and data exposure.
- Apply best practices aligned with **OWASP Top 10 security standards**.
- Strengthen the overall security posture of the application without impacting functionality.

---

3. Key Findings

Security Strengths

- Adoption of modern Jakarta namespace and Spring Security DSL.
- No presence of deprecated `WebSecurityConfigurerAdapter`.
- Strong authentication/authorization structure in place.

Identified Vulnerabilities

- **Hardcoded database credentials** in configuration files.
- **Deprecated header security methods** present in SecurityConfig.

- **Field-based dependency injection**, reducing testability and security clarity.
- **Potential SQL injection** risk through dynamic query concatenation.
- **Logging of sensitive data** in multiple controller and service classes.

---

4. Fixes Implemented

- Replaced field-based and method-level `@Autowired` with **constructor injection**.
- Removed deprecated `xssProtection()`, `contentTypeOptions()`, and `frameOptions()` methods.
- Added **DaoAuthenticationProvider** bean with secure configuration.
- Updated `User.getAuthorities()` to use modern **Java 21 `.toList()` pattern**.
- Parameterized SQL queries, eliminating injection risk.
- Sanitized logging to prevent exposure of sensitive information.

---

5. AI's Role in Development

AI systems such as GitHub Copilot and ChatGPT assisted by:

- Scanning the entire codebase for deprecated or insecure patterns.
- Detecting hardcoded secrets, risky logging, and unsafe input handling.
- Suggesting migration-safe code replacements and patches.
- Generating **multiple structured reports**, including deep technical analysis and executive summaries.
- Automatically validating changes through compilation and testing.

This significantly reduced **manual audit effort, debugging time, and planning overhead**.

---

6. Challenges Faced

- Large number of automated recommendations required **manual verification**.
- Some fixes created temporary **dependency conflicts** in the project.
- Tests needed updates to align with constructor-based dependency injection.
- Maintaining functionality during refactoring required careful sequencing.

---

7. Outcomes & Learning

The security audit resulted in:

- A more **secure, modernized, and testable codebase**.
- Removal of deprecated, insecure, and outdated patterns.
- Strong, explicit authentication configuration with improved maintainability.
- Zero deprecation warnings and successful test execution (~85 tests passed).
- This phase emphasized the importance of **security as a continuous practice** and showcased the value of combining **AI automation with human judgment**.