Vulnerability Fixes Report

COMP47910 Secure Software Engineering

Luis Marron (23202882)

A Lab Journal in part fulfilment of the degree of

MSc. in Advanced Software Engineering

Supervisor: Dr. Liliana Pasquale

UCD School of Computer Science
University College Dublin

August 18, 2025

Table of Contents

Chapter 1: **A01:2021 Broken Access Control**

1.1 CWE-284: Improper Access Control

Description: The BookShop application implements improper access control mechanisms that allow unauthorized users to access restricted functionality and sensitive data. The application lacks proper authentication and authorization checks across multiple endpoints, enabling attackers to bypass security controls and gain unauthorized access to administrative functions and user-specific resources.

CWE Explanation: CWE-284 occurs when the application fails to properly restrict access to functionality or resources, allowing unauthorized users to perform actions or access data that should be protected by proper authentication and authorization mechanisms.

Severity: High

Vulnerability Locations and Type

The application suffered from multiple *Broken / Improper Access Control* weaknesses manifesting as:

- **Vertical privilege escalation**: Security rule protected path pattern `/admin/**`, while the actual admin endpoint controller was mapped to `/admins`. This mismatch let any authenticated non-admin invoke administrative endpoints (e.g. `GET /admins`).
- **Horizontal privilege escalation / Insecure Direct Object Reference (IDOR)**: Cart- and cart-item–related REST endpoints (`/carts/**`, `/cart-items/**`) accepted arbitrary `customerId`, `cartId`, or `itemId` without verifying ownership against the authenticated principal, enabling access/modification of other users' carts.
- **Over-broad data exposure**: User and customer enumeration endpoints (`/users/**`, `/customers/**`) allowed any authenticated user to list or fetch other users/customers and (in the case of `/users`) returned password hashes, breaching least privilege and confidentiality.
- **Unrestricted modification endpoints**: Book management API (`/api/books` POST/PUT/DELETE) was accessible to any authenticated role, permitting non-admin content manipulation.
- **Lack of server-side authorization on web cart actions**: Web MVC endpoints for cart item removal did not assert ownership, allowing crafted requests to remove other users' items.

Mitigation Strategy and Rationale

Defence-in-depth was applied combining **path-based authorization**, **method-level role enforcement**, and **resource ownership (row-level) checks**. This layered approach ensures:

- Misconfigurations in one layer (e.g. path patterns) do not automatically grant access because method-level annotations add a second gate.
- Even with correct role checks, horizontal attacks (guessing IDs) are blocked by explicit ownership validation tying domain object identifiers to the authenticated principal (prevents IDOR).
- Principle of Least Privilege is restored by scoping sensitive endpoints (admin, user lists, data mutations) strictly to the required roles, reducing attack surface.
- Sensitive credential fields are no longer exposed after authorization succeeds, limiting post-auth data leakage channels.

Implemented Security Controls

The following concrete changes were introduced in code (branch `broken-access-control-fixes`):

1. **Corrected admin path mapping**: Updated Spring Security configuration to match the actual controller path `/admins/**` instead of the incorrect `/admin/**`.
2. **Granular authorization rules**: Added HTTP method specific matchers restricting book mutation endpoints (POST/PUT/DELETE on `/api/books/**`) and all `/books/**` (management pages) to role `ADMIN`. User endpoints `/users/**` now require `ADMIN`; customer endpoints enforce authentication plus ownership (see point 3) and reserve modifications for `ADMIN`.
3. **Ownership / row-level checks**: Injected `@AuthenticationPrincipal` into cart and cart item controllers; added helper methods validating that the referenced cart/item's owning customer's username equals the authenticated principal. Requests failing validation now return HTTP 403 (or 404 when appropriate).
4. **Method-level security**: Enabled `@EnableMethodSecurity` and introduced `@PreAuthorize` annotations on controller methods (e.g. book mutations, user controller) to provide a second authorization layer beyond URL pattern matching.
5. **Data minimisation via DTOs**: Replaced direct entity exposure for users and customers with DTOs omitting the password field and limiting attributes to non-sensitive identification data.
6. **Cart web endpoint hardening**: Added ownership validation before allowing removal of a cart item via web MVC endpoint to block cross-user manipulation.
7. **Input encapsulation**: Refactored mutable request payload classes (e.g. add-item request) to use private fields with accessors, preventing accidental uncontrolled field exposure (minor hardening).

Solution Effectiveness

The mitigations directly address the CWE-284 root causes:

- Path correction + role restrictions close vertical privilege escalation vectors.
- Ownership validation eliminates horizontal ID enumeration attacks by binding operations to the authenticated identity.

- Method-level annotations safeguard against future path mapping drift or overly broad ant matchers.
- DTO-based redaction removes unnecessary sensitive data from responses, shrinking impact radius of any residual access gaps.
- Principle of Least Privilege is enforced consistently across both REST and MVC layers, aligning actual access with business intent.

Residual Risk and Next Steps

Remaining improvement opportunities include re-enabling CSRF protection, adding integration tests to automatically enforce authorization policies, centralising ownership checks in a dedicated service / AOP advice, and implementing rate limiting to further reduce brute-force probing of resource identifiers.

Chapter 2: **A02:2021 Cryptographic Failures**

2.1 CWE-326: Inadequate Encryption Strength

Description: The BookShop application stores user passwords in plain text without any cryptographic protection. The database contains user credentials including admin passwords stored as clear text, making them immediately readable if the database is compromised. The application also lacks any password hashing or encryption mechanisms in the authentication process.

CWE Explanation: CWE-326 occurs when the application uses cryptographic algorithms or key sizes that are insufficient to protect sensitive data, making it vulnerable to brute force attacks and unauthorized access.

Severity: Critical

2.2 CWE-522: Insufficiently Protected Credentials

Description: The BookShop application uses Spring Security framework but fails to implement proper credential protection mechanisms. While the framework provides password hashing and encoding capabilities, the application stores passwords in plain text and performs direct string comparisons without any hashing, salting, or other protection mechanisms.

CWE Explanation: CWE-522 occurs when the application uses a protection mechanism that is insufficient to protect credentials, such as using weak hashing algorithms, not using salt, or failing to implement available security features properly.

Severity: Critical

Chapter 3: **A03:2021 Injection**

3.1 CWE-79: Cross-Site Scripting (XSS)

Description: The BookShop application was analyzed for Cross-Site Scripting (XSS) vulnerabilities due to improper handling of user-controlled data in error messages and direct rendering of user input without proper sanitization or encoding.

CWE Explanation: CWE-79 occurs when the application fails to properly validate, sanitize, or encode user-controlled input before including it in output that is sent to other users' browsers, allowing attackers to execute malicious scripts in the context of other users' sessions.

Severity: Low

3.2 CWE-190: Integer Overflow or Wraparound

Description: The BookShop application is vulnerable to integer overflow and underflow attacks due to improper handling of numeric operations without proper validation or overflow checks. This vulnerability allows attackers to manipulate business logic by providing malicious numeric values that cause integer wraparound.

CWE Explanation: CWE-190 occurs when the application performs arithmetic operations on integers without checking for overflow or underflow conditions, allowing attackers to manipulate numeric values to cause unexpected behavior, data corruption, or bypass business logic controls.

Severity: Medium

Chapter 4: **A04:2021 Insecure Design**

4.1 CWE-602: Client-Side Enforcement of Server-Side Security

Description: The BookShop application implements critical security controls on the client-side instead of the server-side, allowing attackers to bypass authentication, authorization, and input validation by making direct API calls to backend endpoints. The application relies on client-side JavaScript to enforce security policies that should be implemented on the server.

CWE Explanation: CWE-602 occurs when the application implements security controls (authentication, authorization, input validation) on the client-side rather than the server-side, making them easily bypassable by attackers who can make direct HTTP requests to backend endpoints.

Severity: High

4.2 CWE-799: Improper Control of Interaction Frequency

Description: The BookShop application lacks any rate limiting or frequency control mechanisms, allowing unlimited interaction attempts with all endpoints. The application does not implement authentication attempt limits, request throttling, or any protection against brute force attacks, making it vulnerable to automated attacks and resource exhaustion.

CWE Explanation: CWE-799 occurs when the application fails to properly control the frequency of interactions, allowing attackers to make unlimited requests that can lead to brute force attacks, resource exhaustion, and denial of service conditions.

Severity: High

4.3 CWE-840: Business Logic Errors

Description: The BookShop application contains multiple business logic errors that violate fundamental application rules and constraints. These include race conditions in the checkout process, lack of duplicate username validation, client-side price calculation vulnerabilities, and missing order validation rules that can lead to inventory overselling, account confusion, and financial manipulation.

CWE Explanation: CWE-840 occurs when the application fails to properly implement business

rules and constraints, allowing attackers to exploit logical flaws in the application's workflow, data validation, and state management to achieve unauthorized outcomes.

Severity: High

4.4 CWE-1173: Improper Use of Validation Framework

Description: The BookShop application completely lacks proper validation framework usage, with no Bean Validation annotations, no @Valid annotations in controllers, and no validation framework dependencies. The application relies solely on minimal client-side validation, allowing malicious or invalid input to bypass security controls and potentially cause data integrity issues, application instability, and security vulnerabilities.

CWE Explanation: CWE-1173 occurs when the application fails to properly use validation frameworks or implements validation incorrectly, allowing invalid or malicious input to bypass security controls, potentially leading to data integrity issues, application instability, and security vulnerabilities.

Severity: High

Chapter 5: **A05:2021 Security Misconfiguration**

5.1 CWE-614: Sensitive Cookie in HTTPS Session Without 'Secure' Attribute

Description: The BookShop application's session cookies (JSESSIONID) are transmitted without the 'Secure' attribute, allowing them to be sent over unencrypted HTTP connections. This vulnerability exposes session tokens to potential interception by attackers through network sniffing, man-in-the-middle attacks, or other network-based attacks.

CWE Explanation: CWE-614 occurs when sensitive cookies are transmitted over insecure channels without proper protection mechanisms, allowing attackers to capture and reuse session tokens to impersonate authenticated users.

Severity: High

5.2 CWE-1275: Sensitive Cookie with Improper SameSite Attribute

Description: The BookShop application's session cookies lack proper SameSite attribute configuration, allowing them to be sent in cross-site requests and enabling various client-side attacks including CSRF.

CWE Explanation: CWE-1275 occurs when cookies are configured without appropriate SameSite restrictions, allowing them to be sent in cross-site requests and enabling various client-side attacks including CSRF.

Severity: High

5.3 CWE-693: Protection Mechanism Failure

Description: The BookShop application includes Spring Security framework but completely disables all protection mechanisms, rendering the security framework ineffective. The application disables CSRF protection, permits all requests without authentication, and fails to implement any of the available security controls, making the protection mechanism completely non-functional.

CWE Explanation: CWE-693 occurs when the application has a protection mechanism in place but fails to use it properly, rendering the security controls ineffective and leaving the application vulnerable to attacks that the protection mechanism was designed to prevent.

Severity: High

5.4 CWE-256: Unprotected Storage of Credentials

Description: The BookShop application stores credentials in unprotected form across multiple locations including configuration files, database initialization scripts, and Docker environment variables. All credentials are stored in plain text without any encryption, hashing, or other protection mechanisms.

CWE Explanation: CWE-256 occurs when the application stores sensitive credentials (passwords, keys, tokens) without proper protection mechanisms, making them vulnerable to unauthorized access and compromise.

Severity: Critical

Chapter 6: **A06:2021 Vulnerable and Outdated Components**

6.1 CWE-269: Improper Privilege Management

Description: The BookShop application exhibits multiple critical privilege management failures including missing authentication for critical admin functions, inconsistent privilege enforcement across endpoints, complete bypass of Spring Security framework, and lack of ownership verification for user resources. These vulnerabilities enable unauthorized access, privilege escalation, and cross-user data manipulation.

CWE Explanation: CWE-269 occurs when the application fails to properly manage privileges, permissions, and access controls, allowing unauthorized users to access restricted functionality or resources that should be protected by proper authentication and authorization mechanisms.

Severity: Critical

6.2 CWE-400: Uncontrolled Resource Consumption

Description: The BookShop application lacks proper resource management controls including database connection pool limits, session timeout limits, request timeout limits, and memory limits. This vulnerability allows attackers to exhaust system resources through unlimited requests, session creation, and large payload attacks, potentially leading to denial of service conditions.

CWE Explanation: CWE-400 occurs when the application fails to properly control resource consumption, allowing attackers to exhaust system resources such as memory, CPU, database connections, or network bandwidth through unlimited or uncontrolled operations, leading to denial of service conditions.

Severity: High

Chapter 7: **A07:2021 Identification and Authentication Failures**

7.1 CWE-287: Improper Authentication

Description: The BookShop application implements fundamentally flawed authentication mechanisms including plain text password storage, direct password comparison without hashing, weak password policies, and complete absence of authentication security controls. The application stores user credentials in plain text in the database and performs direct string comparison during login, making it vulnerable to complete account compromise and unauthorized access.

CWE Explanation: CWE-287 occurs when the application fails to properly verify the identity of users, implement secure authentication mechanisms, or protect authentication credentials, allowing attackers to bypass authentication controls, compromise user accounts, and gain unauthorized access to sensitive data and functionality.

Severity: Critical

7.2 CWE-384: Session Fixation

Description: The BookShop application is vulnerable to session fixation attacks due to the absence of session regeneration after successful authentication. The application uses the same session ID before and after login, allowing attackers to obtain a valid session ID and trick victims into using it, enabling complete session hijacking and unauthorized access to user accounts.

CWE Explanation: CWE-384 occurs when the application fails to regenerate session identifiers after successful authentication, allowing attackers to predict, obtain, or manipulate session IDs to hijack user sessions and gain unauthorized access to sensitive data and functionality.

Severity: High