TDT4237 Software Security - Exercise 2

Introduction

In this exercise your task is to fix the security vulnerabilities found in Exercise 1, and extend the functionality of the application, without introducing new security vulnerabilities in the process:

1) Fix the vulnerabilities you found in Exercise 1.
2) Extend the application with the following functionality:
    a) Users should be able to rate and review books, and mark each others' reviews as helpful or unhelpful.
    b) Users should be able to make and publish lists of books. Users should be able to give each list a title and a description.
    c) Users should be able to edit and cancel orders that are not shipped yet. Due to accounting requirements, existing orders should never be updated in the database; Edited/canceled orders must instead be added as new orders with "negative" items.

The delivery deadline for Exercise 2 is Friday October 25th 1800.

Rules

1. The improved application must retain all original functionality. Securing by removing functionality is not allowed and neither is taking the application offline. The exception to this: You *are* allowed to remove: /debug/mail_log.jsp.
2. The improved application must run on the same technology stack, and on the same server. This means: Java, JSP, Glassfish, MySQL. If in doubt, ask.
3. You are not allowed to use third-party libraries to implement the features, except when specifically approved by the course staff. We will maintain a public list of whitelisted libraries. If you want to use a library that is not on this list, email tdt4237@idi.ntnu.no and we will consider including it on the whitelist.

Keep in mind that in Exercise 3 another group will attempt to break into your application, so make sure they can't. ;)

Expected deliverables

1) The source code for your improved application and an SQL dump of the database.
2) A report describing what you have done to complete the exercise. This report should - as a minimum - contain the following:
    a) A list of the discovered vulnerabilities from Exercise 1.
    b) For each vulnerability: Details on what you have done to fix it.
    c) Details on design and implementation of the added functionality. (Hint: you should use threat modeling to explore potential weaknesses in the added functionality. Use these models as a basis to identify how to create a secure implementation of the new functionality. Risk analysis is a useful tool to prioritize.)
3) The application deployed to the course application server.