XJCO3911 Secure Computing
Coursework 2

This exercise concerns web application vulnerabilities and how they can be fixed. It is worth 15% of your overall grade.

The vulnerable application is Java-based. To run it, you will need JDK 8 or newer installed on your PC. It should work on Linux, Mac and Windows.

The Scenario

You are provided with the source code of a web application in `patients.zip`. This is a crude attempt by an inexperienced developer to implement part of a patient records system. The idea is that GPs in a surgery can log in to the application and search for details of patients that they are currently treating.

The application uses Jetty as a built-in web server. Request processing is done by a Servlet. Data storage is provided by an SQLite 3 database, and queries of the database are done using JDBC. HTML pages are generated using the Freemarker template engine.

Tasks

Analysis of Security Flaws

1. Examine the database used by the application. Amongst other things, this will give you the login credentials and patient details that you need to test the application.

   You can do this on the command line using the `sqlite3` tool: the `.schema` command will tell you the structure of the database and you can issue SQL queries at the command prompt to examine its contents. If you prefer a tool with a GUI, there are many available—e.g., DB Browser.

2. Run the application from the command line using

   ```
   ./gradlew run
   ```

   (On Windows, omit the leading `./`)

3. Visit `http://localhost:8080` in a web browser to interact with the application. Use the information obtained in Step 1 to explore different paths through the application.

4. Experiment with the application to identify a security issue that can be trigged via the web interface. Explore the issue as fully as possible. Explain what the issue is and describe the steps you took to identify it.

5. Identify and discuss one other security issue that cannot be seen via interactions with the web interface but is evident by studying the code of the application.

The analysis is worth **16 marks**.

Implementation of Security Fixes

1. Choose **one** of the security issues identified previously. Modify the application (and, if necessary, the database) to fix these issues.

2. Test the application to make sure that it still works and that it is no longer vulnerable.

3. Describe briefly the changes that you have made to the application. Explain in detail why these changes have fixed the issue.

Your fix and the write-up are together worth **8 marks**.

Deliverables

You need to submit your write-up and the modified application.

Your write-up should be in the form of a PDF file, containing no more than two A4 pages (excluding any cover sheet). This file MUST be named `report.pdf` and it MUST be put in the same directory as the `build.gradle` file.

When you have put this PDF file in the correct location, enter the following command:

```
./gradlew submission
```

This will create a file named `cwk2.zip`, containing everything that needs to be submitted.

Submission

Use Minisign to sign the file `cwk2.zip`, following the same procedure as for Coursework 1.

Submit the files `cwk2.zip` and `cwk2.zip.minisig`, via the link provided for this purpose in Minerva. The deadline for submission is **8 AM on 18 November**.