COMP3310 Group Project

S1 2024

Project Overview

The goal of this unit has been to teach you how to design secure applications in all phases of the software development life cycle (SDLC), in line with the expectations of a software or security professional in any organisation. In this project, you will apply the skills you have learned throughout the unit to the assessment of a small web-based application.

You are given a basic photo gallery website written in Python. Your task is to analyse the website for security vulnerabilities and then extend it with features of your choice, using secure application design and development principles.

Assignment marks: 40% of overall grade.
Due dates:

- Presentations (10%): Week 12 during your workshop class.
- Project report and code (30%): Sunday, Jun 2, 11:59 pm (end of Week 13)

Note that although you will do your presentations as a group, the grades are calculated **individually.** This means that every member of your group must be available to present.

Project Description

Cloud Photos is a simple photo-sharing website that allows individuals to upload, manage and delete photos. It has been implemented according to a set of functional requirements and a technical specification. Unfortunately, there were no security design principles explicitly used in the design or implementation of the site. As a security specialist, you have been tasked with performing a security analysis of the website.

Setting up the code: Follow the instructions on iLearn to clone the repository and download it to your local machine. Instructions for running the website are available in the README.md file included with the code.

Functional Specification: The website features incorporate the following functional requirements:

- The landing page (home page) shows a list of photos that have been uploaded to the site.
- Hovering over a photo displays the name of the photo contributor, the caption and the desc the description of the image. The description is optional.
- Users can upload, edit and delete photos from the site.

Technical Specification: The website has been implemented according to the following technical specifications:

- The home page will be accessible at the default url: /
  - The photos will be displayed sorted alphabetically by filename.
- Individual photos will be stored at the url: /uploads/<file> where <file> is the filename of the photo on the filesystem. Eg. /uploads/mountains.jpg
- Metadata (username, caption and description) will be stored in a database which generates a unique id for each image.

- Photos can be edited at the url: /photo/<id>/edit/ where <id> is the unique id of the photo in the database. Only the caption, user and description can be edited.
- Photos can be deleted at the url: /photo/<id>/delete where <id> is the unique id of the photo in the database. Deleting a photo removes its entry from the database and its file from the filesystem.

Part 1: Security Analysis of the Website (50 marks)

For this part of the project, you will perform a basic security analysis of the existing website. You need to complete the following tasks:

Task 1: Security requirements (10 marks)

Identify at least 10 security requirements that you would include in either the functional or the technical specification. For each security requirement you identify, explain what vulnerabilities may occur without this security feature. You must refer to the OWASP Top 10 or other security principles studied in lectures.

Task 2: Threat Modelling (15 marks)

This task requires you to incorporate threat modelling into the specifications. As with Assignment 1, you may use Threat Dragon or any software/tool of your choice.

a. Construct a dataflow diagram for the website based on the implementation provided in the code. Your diagram should include entities such as the web browser, web application and database (and any others used in the implementation of the website).
b. Draw the trust boundaries that occur between elements of your diagram. For each boundary, include 1-2 sentences explaining the threats that could cause a security breach on that boundary.
c. In your dataflow diagram, identify at least 5 security threats to the website. You should indicate the type of threat, the entity at which the threat occurs, the vulnerability it exposes, and a mitigation strategy.

Task 3: Code analysis (20 marks)

Perform a code analysis of the website code. You should use a combination of automated tools (eg. Codeql) and manual code review. Your analysis should include the following:

a. Identify any security vulnerabilities or insecure coding practices that you find in the code. You should refer to the OWASP Top 10 or other security principles. You must identify the line(s) of code that are vulnerable.
b. Assess whether the code is vulnerable to attacks, paying particular attention to those studied in lectures such as XSS, CSRF, SSRF and denial of service attacks.
c. For any vulnerabilities that you find, suggest mitigation strategies and explain why these would be effective.
d. Suggest tests that you would incorporate into the build cycle to ensure that the code remains protected against the vulnerabilities and attacks that you identified. Your tests should incorporate security checks even where you found that the code is not vulnerable to attacks (why?)

Task 4: Summary and recommendations (5 marks)

Provide a summary of your findings and give an overall assessment of the security of the web application. If you would recommend changing some of the specifications or redesigning aspects of the code, provide reasons.

Part 2: Authentication (30 marks)

The website owner would like to add a new feature to the website: to allow users to manage their photos, and to allow the website administrator to manage all of the photos. The website owner provides the following functional specification:

- Users must be able to manage their own uploaded photos, including edit and delete
- Users must not be able to edit or delete photos which they did not upload
- Administrators must be able to edit or delete any photo

Given the above specification, complete the following tasks:

Task 5: Technical specification (15 marks)

Write a technical specification that incorporates the above functional requirements. You can use the technical specification given in the Project Description as a guide.

a. You are expected to include an authentication mechanism as part of your technical specification, which distinguishes between the 3 types of users (non-logged-in users, logged-in users and administrators). You may include more than 3 types of users, and you may choose any authentication mechanism that you think is appropriate to this task.
b. Your specification should indicate any new endpoints that will be included and which users can access that endpoint. You may use a table as was done in lectures.
c. Your specification should also indicate behaviours that should occur in the case of errors. For example, you may choose to redirect users to the login page if they land on a page that is designed for logged-in users only.
d. Your specification must include security features by design. Identify which security features you have incorporated, referring to appropriate security principles.

Task 6: Threat modelling (5 marks)

Update the threat model that you developed in Task 2, incorporating the new features you created in Task 5. You should identify new trust boundaries and new security threats resulting from the above specifications.

Task 7: Implementation (10 marks)

Implement some aspects of your authentication mechanism. You will not lose marks for failing to get the authentication mechanism working. You will be graded on the following:

a. Identify in the code (eg. in comments) where you have applied secure coding principles and/or where secure coding principles should be applied.
b. Identify tests that you would incorporate into the build process to ensure that security vulnerabilities are not present (or not introduced) in the code. You should include stubs for the tests (if not implemented) with appropriate documentation/comments so that these can be graded.

Part 3: Additional Features (20 marks)

For this part of the project, your group should choose 2 features that you would like to incorporate into the website to improve its functionality. You may choose 1 feature from the following list and invent 1 feature of your own.

1. Users can leave comments on photo pages.
2. Photos are searchable using keywords from the metadata.
3. Users can tag their photos using predefined categories, which can then be used in a “Browse by Category” feature.

4. Users can label their photos as “private”, which means only users with the URL of the photo can access it. Private photos should not appear when browsing or searching, including in Google searches.

For each of the 2 features that you choose, you should complete the following tasks:

Task 8: Technical specification and threat modelling (5 marks)

Write a technical specification incorporating secure design principles, as you did for Task 5.
   a. Your specification should include newly defined endpoints and indicate new threats or vulnerabilities that may arise as a result of including this feature.
   b. You should identify which security features you have incorporated, referring to appropriate security principles.
   c. You should include an updated threat model that includes any new trust boundaries and identifies new security threats.

Task 9: Implementation and testing (5 marks)

Provide an implementation of your feature. (You will not lose marks if your implementation is incomplete or not functional). As with Task 7, you will be graded on the following:
   a. Identify in the code (eg. in comments) where you have applied secure coding principles and/or where secure coding principles should be applied.
   b. Identify tests that you would incorporate into the build process to ensure that security vulnerabilities are not present (or not introduced) in the code. You should include stubs for the tests (if not implemented) with appropriate documentation/comments so that these can be graded.

What to Submit

Your group is required to submit the following for marking:

1. A report (PDF format) detailing your answers to the project tasks listed above. Your report should be written in English in a formal style (no bullet points). The following information must be included:
   a. A title page indicating the names and student IDs of the members in your group. It should also include a link to your code repository and the snapshot tag that you created for the final submission.
   b. Write-ups for each of the project tasks in clearly marked sections (You may use the Task number to indicate which task you are writing about).
   c. You should use drawings, images and tables where appropriate (eg. For threat models / dataflow diagrams or listing new web endpoints as part of a technical specification).
   d. A clear indication of which team members worked on which aspects of the project.
   e. Citations for any websites or other resources that you use for assistance. If ChatGPT or other AI is used, you must cite this where it is used in your report.

2. You will create a tag (“snapshot”) of your code repository which constitutes your final code submission and will be included in the report as indicated above. Your code should include documentation or comments which clearly indicate which changes were made to the code, and for which tasks these changes were made. The documentation and the code supplied in the repository snapshot will be used to grade the tasks identified as “implementation” tasks in the project specification above.

3. Your presentation slides for the presentation to be held in the Week 12 workshops. Your slides will need to be submitted on the Monday of Week 12. More information will be provided on iLearn.

You should check iLearn for instructions on when and where to do your submission.

Rubric Guide

Your report will be graded according to how well you answered the questions for each Task. This includes the quality of your writing and the clarity of your explanations.

Your code will be used to grade the tasks described as “Implementation”. You will **not** lose marks if your code does not run; the tasks have been designed so that you can demonstrate your understanding of secure coding practices. Your code should include sufficient documentation to demonstrate your understanding even when you have not been able to implement the required feature(s).