ITRI615 Project

Lecturer: Prof. Lynette Drevin

Facilitator: Mr. Bernard Swanepoel

Date: 16 May

1 Introduction

This project is designed to be enjoyable and to learn a lot of new relevant industry knowledge, encouraging students to expand their knowledge beyond their current knowledge. The project consists of 70% practical work and 30% theoretical analysis.

NOTE!

- **Python is prohibited due to its ease of implementation, pushing students to learn other relevant languages and technologies.**
- **GitHub or any other source control should be used throughout this project.**

2 Practical part (70 marks)

2.1 Basic microservice (15 marks)

Students must create a basic microservice in any language except **Python**. This microservice can implement any functionality with any database and data (you could even implement an ML/DL model as a microservice if you wanted to). The initial microservice should be simple and account for only 15 marks of the project. The core objective of this project is to secure the microservice by implementing various security techniques. The project is broken down into the following components:

2.2 API gateway requirement (5 marks)

All access to microservices must go through an API Gateway.

2.3 Authentication (10 marks)

Implement authentication using JWT (JSON Web Token) or a similar method. Ensure that users can securely log in and interact with the microservice.

2.4 Authorization (10 marks)

Implement OAuth or a role-based authorization system to ensure that users have proper permissions and access control to the microservice.

2.5 Logging and monitoring (10 marks)

Implement basic logging and monitoring using tools such as Kibana and Prometheus or code-based solutions like Logger and LoggerFactory in Java Spring Boot or Winston in Express.js.

2.6 Input validation and limiting (10 marks)

Ensure all incoming requests are validated and sanitized to prevent security vulnerabilities such as SQL Injection and XSS. Additionally, implement request rate limiting to mitigate potential DDoS attacks.

2.7 Frontend implementation (6 marks)

Develop a simple web-based user interface using vanilla JavaScript or a JavaScript framework such as React, Angular, or Vue. No styling or CSS is required; the goal is to verify secure communication between the frontend and backend.

2.8 Application of security patterns (4 marks)

To conform to best practices, students should apply formal security patterns where applicable. Here is a link to some useful resources on security patterns.

2.9 Additional features (10 marks)

Bonus marks will be awarded for implementing additional security measures, deploying the microservice in the cloud, or using innovative technologies. However, marks cannot exceed 100% of the total project score.

3 Theory component (30 marks)

3.1 Microservice scenario (3 marks)

Give the scenario of your microservice. Why is it important that this microservice's data is kept secure?

3.2 Security analysis (7 marks)

Explain how your entire frontend and backend are secured. Discuss the security strategies implemented and their effectiveness.

3.3 Authentication and authorization methods (10 marks)

Provide an overview of different authentication and authorization methods. Justify your choice and explain why it is the most suitable approach for your project. Provide references where applicable.

3.4 Real-world security failures and analysis (10 marks)

Analyze real-world security failures where improper authentication, authorization, or security implementations led to vulnerabilities. Discuss what security measures should have been implemented to prevent such failures. Provide references where applicable.

1. Please add the theory part on the template that is provided on eFundi

2. The code will be uploaded on the Dropbox tool.

3. Please use GitHub - and make your setting **public** for the duration of the marking process. Provide the link/details.

bernardswanepoel1510@gmail.com

4 Summary of marks

Component	**Description**	**Marks**
Total Practical Marks: 70		
Basic microservice implementation	Develop a basic microservice in any language except Python	15
API gateway requirement	Ensure microservice is accessed via an API Gateway	5
Authentication	Implement authentication using JWT or a similar method	10
Authorization	Implement OAuth or role-based authorization	10
Logging and monitoring	Use tools like Kibana, Prometheus, or code-based logging solutions	10
Input validation and limiting	Validate and sanitize inputs, implement rate limiting	10
Frontend integration	Develop a basic web UI (React, Angular, Vue, or Vanilla JS)	6
Application of security patterns	Apply formal security patterns where applicable	4
Additional features	Bonus marks for advanced security or cloud deployment in AWS, Google Cloud or Azure	10
Total Theory Marks: 30		
Microservice scenario	Give microservice scenario and explain why it is relevant to be kept secure	3
Security analysis	Explain security measures in the frontend and backend	7
Auth methods comparison	Discuss different authentication and authorization methods, motivate your choice	10
Real-world failures	Analyze security failures and suggest improvements	10
Deductions (if applicable)		
Credentials exposed in GitHub	Deduction for exposing sensitive credentials in GitHub repositories	-10
Use of Python	Made use of Python as a "easy way out"	-10
Total for this project: 100		

Table 1: Project Mark Breakdown