Practice Report

Delivery 1

Student: Anna Melkumyan Canosa

Student: Cecília Maria Rodrigues Correia

Student: Irene Cerván Barriga

Student: Johanna Nuñez

Student: Oriol Ramos Puig

Escola Tècnica Superior
d'Enginyeria de Telecomunicació de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

telecos BCN
Escola Tècnica Superior
d'Enginyeria de Telecomunicació de Barcelona
UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona
FIB

Contents

Introduction

Our team will develop a secure web app inspired by Too Good To Go, where vendors register surplus food packages for sale and buyers can view and purchase them only with cash because card payment is not available in the web app. Vendors manage and publish the packages they want to sell while regular users access the platform and see all the packages available to buy.

Framework used

- **Backend:** django, to handle business logic, manage user authentication, and provide a secure connection between the database and the web interface.
- **Database:** postgresSQL, chosen as the database for its reliability, strong security features, and compatibility with Django's ORM.
- **Frontend:** django templates, for the frontend to dynamically render pages and ensure seamless integration with the backend logic.
- **Security:** *OWASP (ASVS)* 4.0.3, as our main security framework because it provides a structured set of controls and best practices for security development. It helps us verify authentication, data protection, and access control following industry standards.
- **Docker*:*** is listed as part of our framework setup because it provides isolated and reproducible environments. It improves security by separating services, managing dependences, and protecting configuration data through controlled containers.

Basic Functionalities

- User registration.
- Log In.
- Content registration and edition: all sellers can create and delete their packages and edit the descriptions, price, name, etc and upload images of their own packages.
- Customers can see all the packages offered by sellers to decide and book the one they want to buy.
- Package filter by some params (category, price, etc).
- User profile management (editing info like name, email, etc) and being able to delete their own account if they want.
- Admin functionalities: deactivate accounts, etc.

Security Features

- Create and install a server certificate. It can be a self-signed certificate. (using Let's Encrypt)
- Put some rules for the password (x length, number, special characters, etc)
- Password protection inside the database
- Access Control - Privacy rules: We will have only an user admin, sellers and buyers (the users will have access to edit their own content).
- File upload security: restrict files only to JPG and PNG, validate the size of the file,…

- Check software vulnerabilities during project development and when put into production: To ensure a secure lifecycle, our team will follow the recommendations of the OWASP ASVS 4.0.3 framework. We will use pip-audit to automatically scan known vulnerabilities. Then bandit for common security including checks of potential EEXML even if XML is not primarily used. The result will be documented for traceability.

Bibliography

List of reviewed webs that are not code repositories.
OWASP :
https://atenea.upc.edu/pluginfile.php/6584636/mod_resource/content/6/OWASP%20Application%20Security%20Verification%20Standard%204.0.3-es.pdf

Bandit: https://bandit.readthedocs.io/en/latest/

Pip-audit: https://pypi.org/project/pip-audit/,

Docker: https://www.youtube.com/watch?v=KiACzzCtz1s

Let's encrypt: https://letsencrypt.org/

Code repositories
https://github.com/pypa/pip-audit