University of Passau

Professor: Prof. Dr. Joachim Posegga
Assistants: Talaya Farasat & Aleena Elsa George

Security Insider Lab II- SS 2024

Project Abstract

Sayed Alisina Qaderi (112092)
Dusan Dordevic (115556)
Atiqullah Ahmadzai (112518)

April 2024

Project Overview

The Vulnerability Scanner project aims to develop a tool that identifies security vulnerabilities in software projects. It provides users with the ability to log in, create and manage projects, and scan code repositories or Python source files for various vulnerabilities. The system employs machine learning models to detect vulnerabilities and generates comprehensive reports. Users can rescan their code after making fixes to ensure security improvements.

Functional Requirements

1. **User Authentication**
   - **Function:** Login with username and password
   - **Description:** Users must authenticate themselves using a username and password to access the system. This ensures that only authorized users can utilize the scanner.
2. **Project Creation**
   - **Function:** Create project
   - **Description:** Users can create a new project by providing a name and selecting either a code repository (e.g., GitHub, Bitbucket) or uploading a Python source code file.
3. **Repository Management**
   - **Function:** Choose repository or Python source code file
   - **Description:** Users can specify the source of the code to be scanned. This can be a link to a repository or a direct upload of a Python source code file.
4. **Code Retrieval**
   - **Function:** System will pull the repository
   - **Description:** The system will fetch the code from the specified repository or process the uploaded Python source code file.
5. **Data Conversion**
   - **Function:** System will convert it to w2v
   - **Description:** The retrieved code will be converted to word2vec (w2v) format, preparing it for vulnerability analysis.
6. **Vulnerability Detection**
   - **Function:** Find the vulnerability with one of the trained models (LSTM, CNN, MLP)

- **Description:** The system will analyze the word2vec formatted code using one of the pre-trained machine learning models: LSTM (Long Short-Term Memory), CNN (Convolutional Neural Network), or MLP (Multi-Layer Perceptron) to identify vulnerabilities.

7. **Report Generation**
   - **Function:** System will generate report
   - **Description:** After scanning, the system generates a detailed report outlining the detected vulnerabilities, their severity, and recommendations for fixing them.
8. **Rescan Capability**
   - **Function:** User can rescan after fixes
   - **Description:** Users can rescan their code after making the recommended fixes to ensure that vulnerabilities have been addressed.
9. **Supported Vulnerabilities**
   - **Function:** System support the below vulnerabilities
   - **Description:** The system specifically scans for the following types of vulnerabilities:
     - XSS (Cross-Site Scripting)
     - Path Disclosure
     - Remote Code Execution
     - Command Injection

Technical Architecture

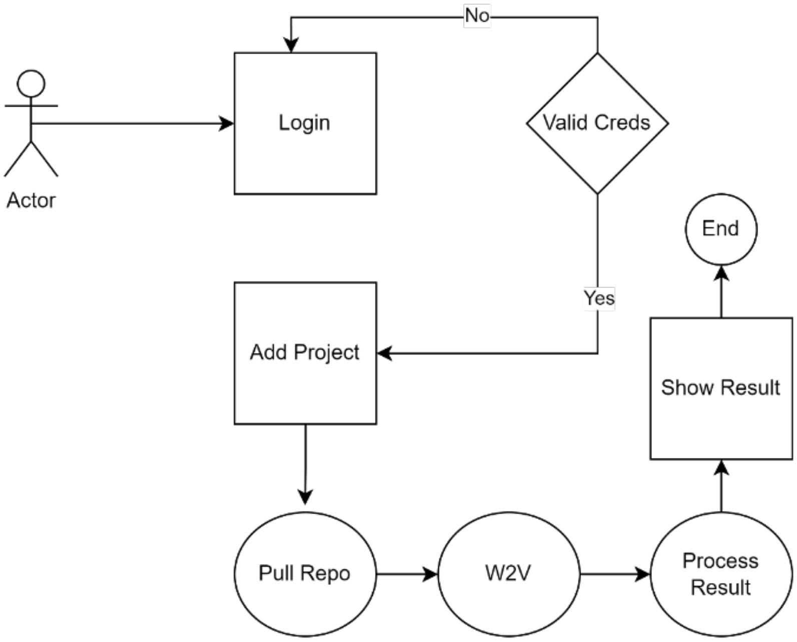
System Components

1. **User Interface (UI):**
   - Login Page
   - Dashboard for project management
   - Report viewing and rescan options
2. **Backend:**
   - Authentication Service
   - Project Management Service
   - Code Retrieval Service
   - Vulnerability Detection Service
   - Report Generation Service

3. **Database:**
    - User credentials
    - Project details
    - Scanning results and reports
4. **Machine Learning Models:**
    - Pre-trained LSTM, CNN, and MLP models for vulnerability detection

Workflow

1. Login: User logs in using credentials.
2. Project Creation: User creates a new project and selects the source of the code.
3. Code Retrieval: System fetches the code from the repository or processes the uploaded file.
4. Data Conversion: Code is converted to w2v format.
5. Vulnerability Detection: Selected ML model analyzes the code for vulnerabilities.
6. Report Generation: System generates and displays a report.
7. Rescan: User can rescan the code after making fixes.



User Roles

Administrator

- Manage users
- Oversee project activities
- View all generated reports

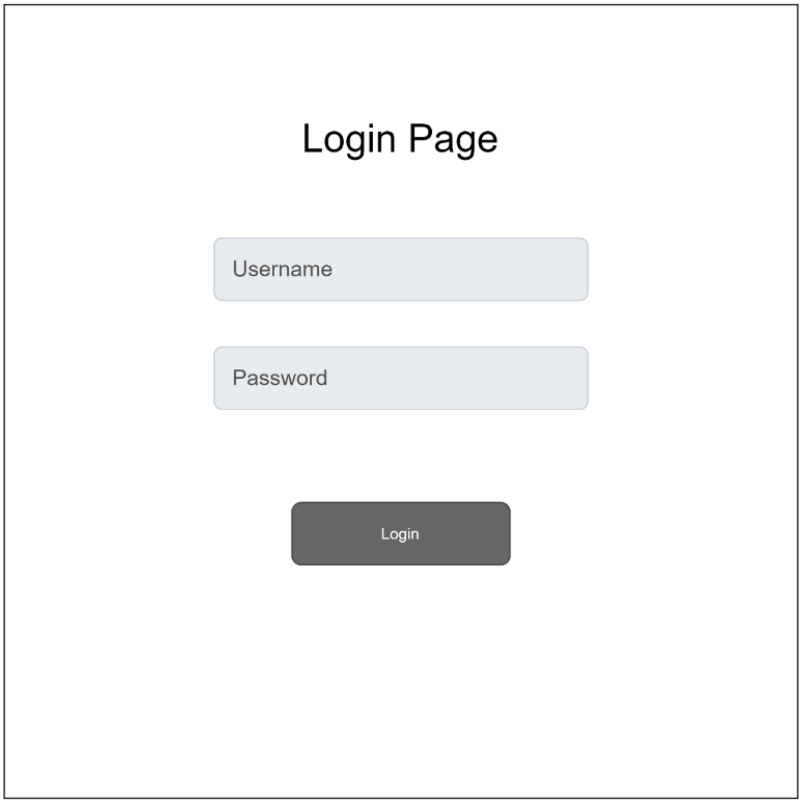
Regular User

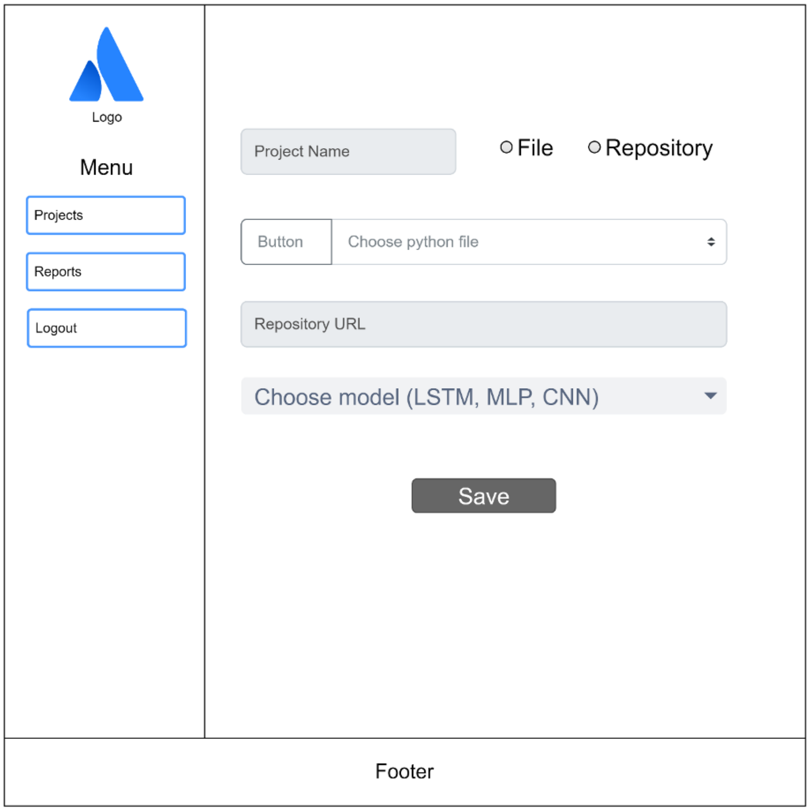
- Create and manage their own projects
- Perform scans and view reports
- Rescan projects after fixes

Non-Functional Requirements

- **Security:** Ensure secure handling of user credentials and project data.
- **Performance:** Efficiently handle code retrieval and scanning for large repositories.

- **Scalability:** Support multiple concurrent users and scans.
- **Usability:** Provide an intuitive user interface for easy navigation and use.





Conclusion

The Vulnerability Scanner project aims to provide a robust tool for identifying and fixing security vulnerabilities in software projects. By leveraging machine learning models, it ensures accurate and efficient vulnerability detection, aiding developers in maintaining secure codebases.