Adaptive IAM

-Manik Singla(s222092193)

1. Introduction

The Adaptive IAM (Identity and Access Management) application is a web-based system designed to manage user authentication and monitor login attempts. It uses Flask as the web framework and integrates various libraries for enhanced functionality.

2. Features

2.1 User Authentication

- The application allows users to log in using predefined credentials stored in the application.

- It provides a simple login form where users can enter their username and password.

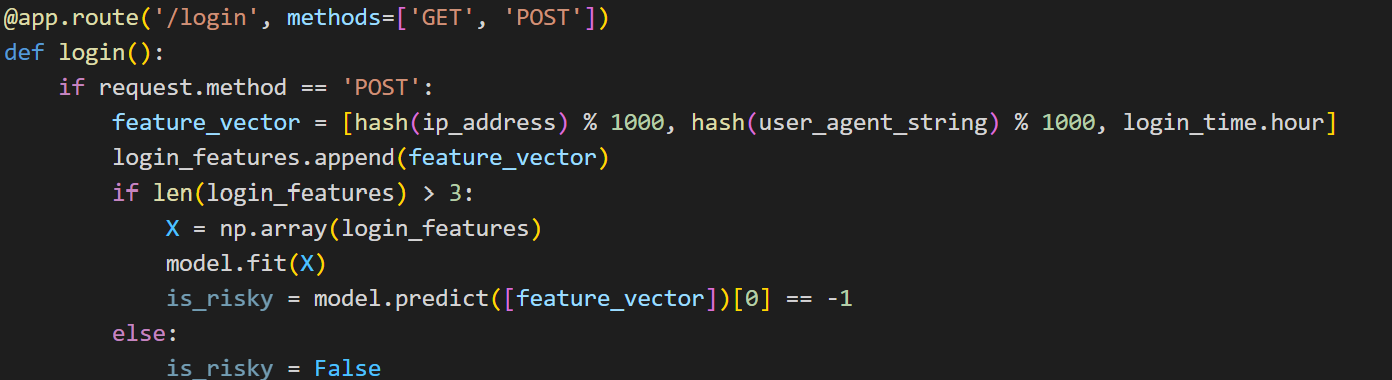
- Successful and failed login attempts are tracked and recorded.

2.2 Risk Assessment

- Utilizes machine learning models, specifically the Isolation Forest algorithm, to detect risky login attempts.

- Risk assessment is based on features such as IP address, user agent, and login time.

- The model is trained on login attempt data to identify anomalies that may indicate a security threat.

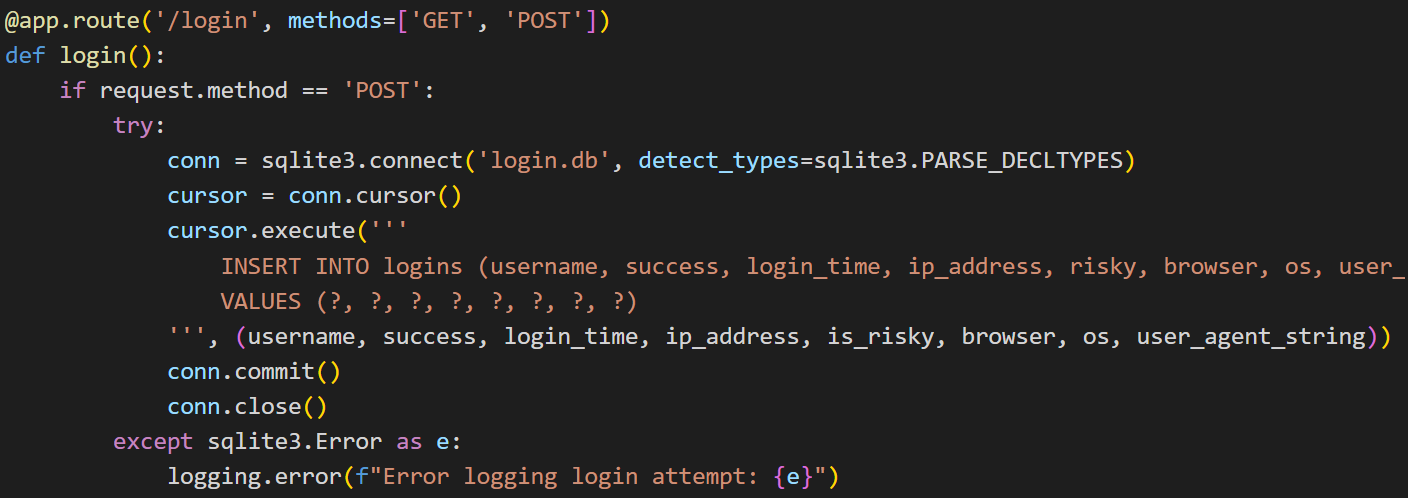


2.3 Logging

- Records login attempts in an SQLite database.

- Each record includes details such as username, success status, timestamp, IP address, browser, OS, and user agent.

- The database schema is designed to store and retrieve login attempt data efficiently.



2.4 Web Scraping

- Provides a feature to scrape web pages and extract information like title, description, and headings.

- Uses BeautifulSoup to parse HTML content and extract relevant information.

- The scraped data is displayed to the user in a structured format.

A screen shot of a computer code

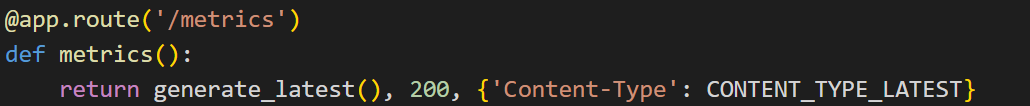
Description automatically generated

2.5 Metrics

- Exposes Prometheus metrics for monitoring various aspects of the application.

- Metrics include the total number of login attempts, successful logins, risky logins, and failed logins.

- These metrics can be used to monitor the application's performance and detect potential security issues.



3. Implementation Details

3.1 Flask Application

- The main application is built using Flask, a lightweight web framework for Python.

- Routes are defined for login, scraping, and metrics.

- The application uses Flask's templating engine to render HTML pages.

3.2 Machine Learning

- Uses the Isolation Forest algorithm from scikit-learn to identify anomalous login attempts.

- The model is trained on a feature vector that includes hashed IP address, hashed user agent, and login time.

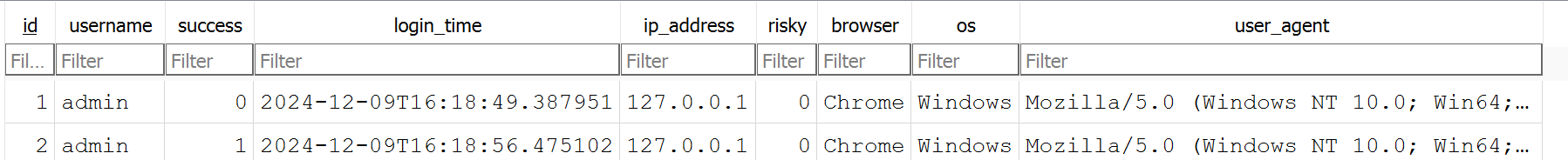
- The contamination parameter is set to 0.1, indicating that 10% of the data is expected to be anomalies.

3.3 Database

- SQLite is used to store login attempt records.

- The database schema includes fields for username, success status, login time, IP address, risky status, browser, OS, and user agent.

- Datetime fields are properly handled using adapters and converters to ensure accurate storage and retrieval.



3.4 Web Scraping

- BeautifulSoup is used to parse HTML content and extract relevant information.

- The application sends HTTP requests to the specified URL and processes the response to extract the title, description, and headings.

- The extracted information is displayed to the user in a structured format.

3.5 Prometheus Integration

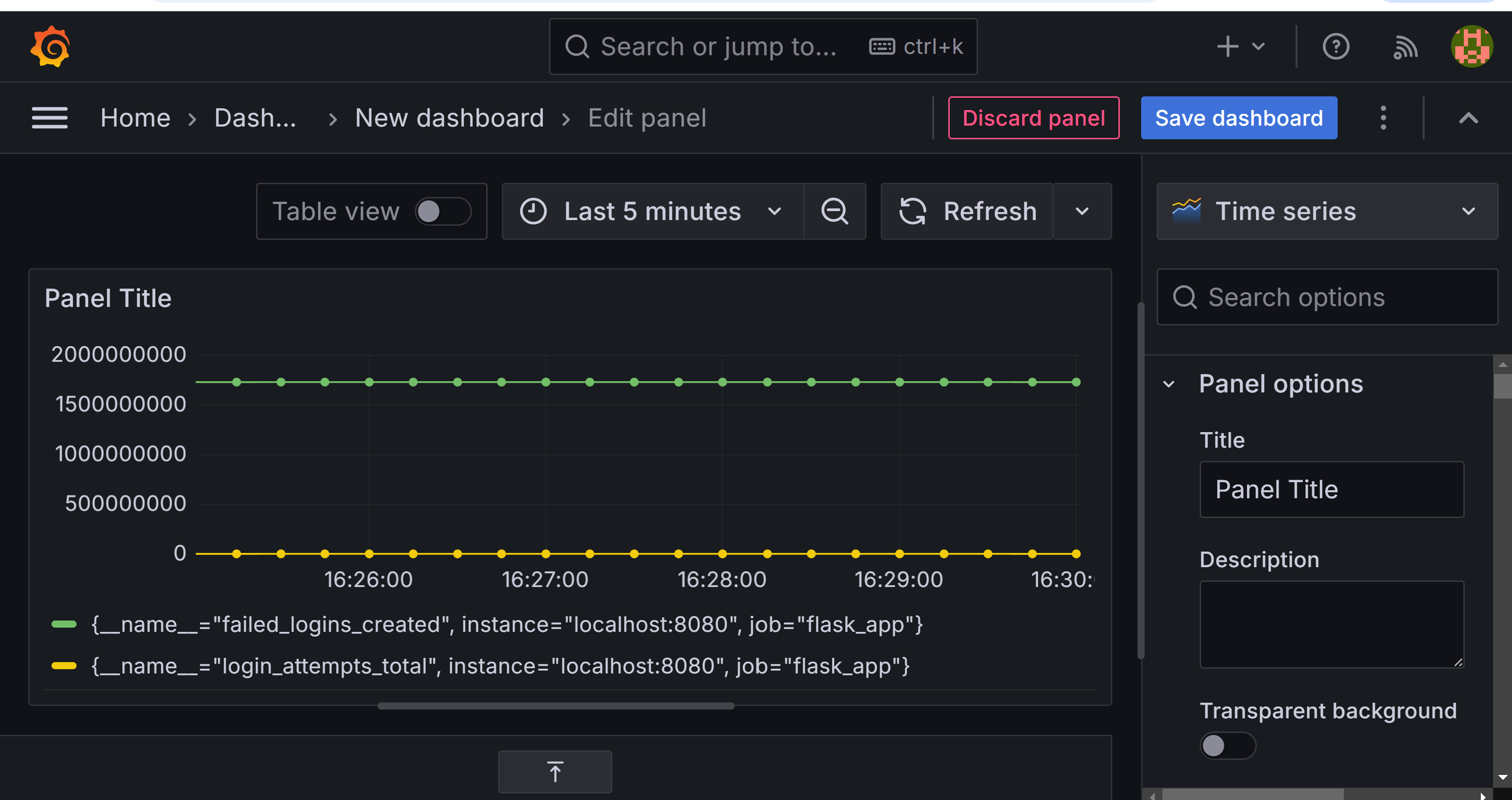
- Metrics are exposed using the Prometheus client library.

- The application starts an HTTP server on port 5000 to serve the metrics endpoint.

- Metrics can be scraped by Prometheus and visualized using Grafana or other monitoring tools.

A screenshot of a computer

Description automatically generated



4. Conclusion

The Adaptive IAM application provides a robust solution for managing user authentication and monitoring login attempts. Its integration with machine learning models and Prometheus metrics enhances its capability to detect and report risky logins, ensuring better security and monitoring. The application's features, including user authentication, risk assessment, logging, web scraping, and metrics, make it a comprehensive tool for identity and access management.