Team So Far, So Good: PRDv2

Project Title: GitHub Auditor

Company: Xenon

Team

- **Lead:** Chris Yang
- **Scribe:** Thomas Zhang
- **Other members:** Nico Wong, Edward Thai, Kyle Stubbs

Introduction

Background

Xenon Partners manages a portfolio of many companies and provides them services critical to improving their success. They provide capital and insight to the companies to help them grow and reach the next level. This has resulted in the majority of these clients strategically exiting and making it to the next step.

Problem

Each of the companies that Xenon Partners manages is in charge of a GitHub organization. These companies do not have the tools necessary to thoroughly dive into the repositories and check for potential security issues. Sensitive information that could cause harm within the organization could go undetected if it is not picked up when being pushed into the repository. Xenon's stakeholders would massively benefit from a tool that could automatically detect such issues and have an easy way of accessing and understanding what is going on.

Existing Solution

GitHub already has a built-in dashboard designed to track activity within an organization. This provides great insight into the commits that are occurring within the repositories. However, it does not provide any insight into the security of the organization, which requires manual checking for information such as two-factor authentication or Dependabot alerts. This is effective for looking at the activity that is occurring within the organization, but the lack of sufficient security information makes it inefficient to check for each possible concern.

Goals/objectives

The goal of our project is to implement a GitHub auditor that contains insights into the repositories contained within a company's organization. This will consist of:

- GitHub API server integration to make requests multiple times per day.
- UI Dashboard in order to display information to the permitted users.
- GitHub authentication in order to verify that the users are allowed to access the necessary dashboard.
- Alerts set up to notify users in case of any potential security concerns.

In addition to the dashboard being set up to display all of the information in a user-friendly way, it would be very beneficial to notify the users of any possible security issues, so that way they are notified without having to look at the dashboard. These alerts would report on issues pertaining to:

- Two-factor authentication requirement
- Outside contributors
- Dependabot alerts
- Pull requests lacking an assigned viewer
- Sensitive information such as API keys or passwords in the code

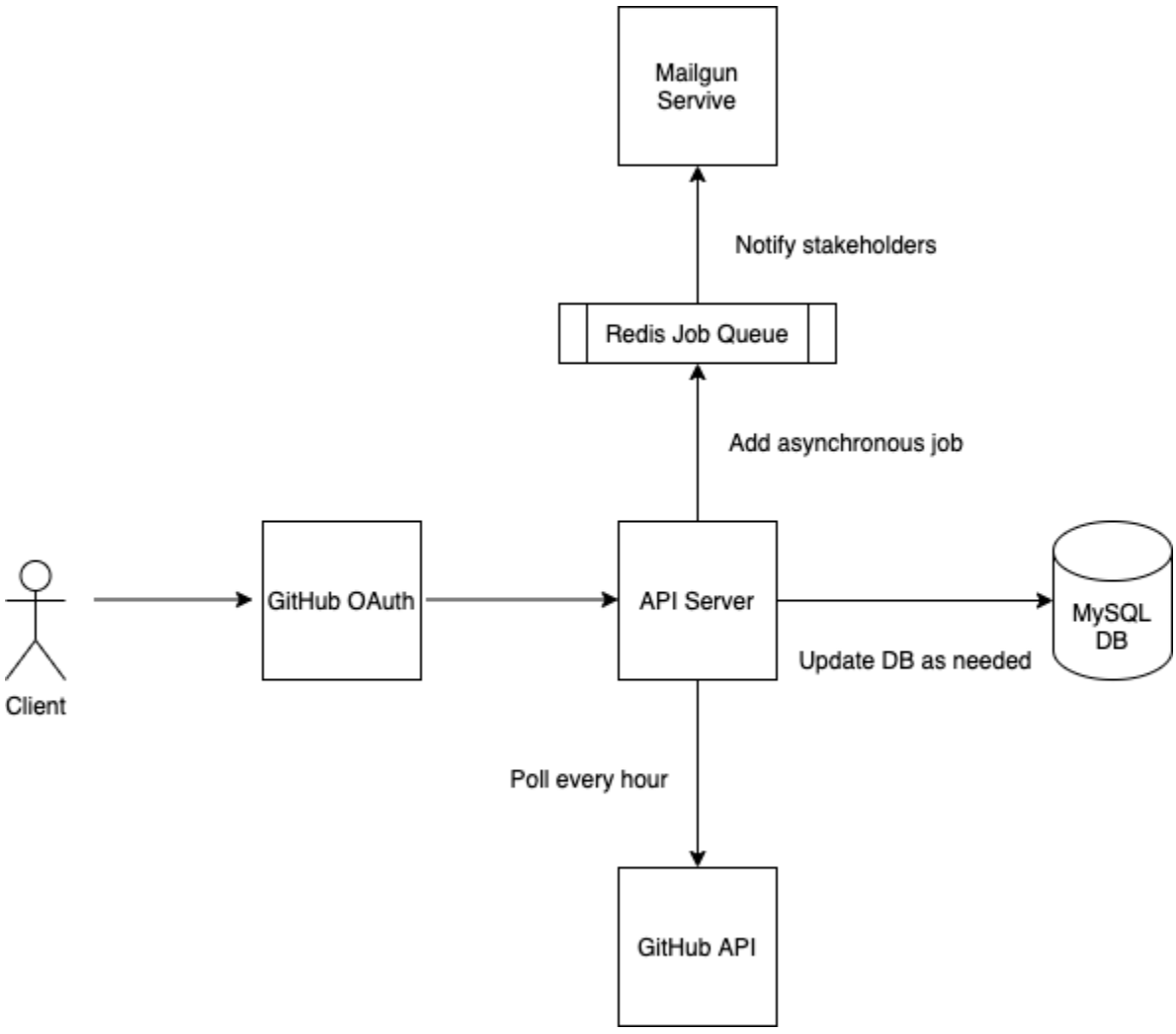
These security concerns will either be detected from the API or use a text-processing tool to detect certain risks in the code. The clients will receive an email containing all of the information needed to alert them, and then the data will be easily accessible when they go to take a look at the dashboard.

Assumptions

- Xenon Partner's clients own a GitHub organization containing multiple repositories.
- Security concerns will happen and need to be addressed
- The preferred method of receiving alerts is by email rather than another form of alert
- The dashboard will be easy to interpret and understand by clearly organizing and labelling everything

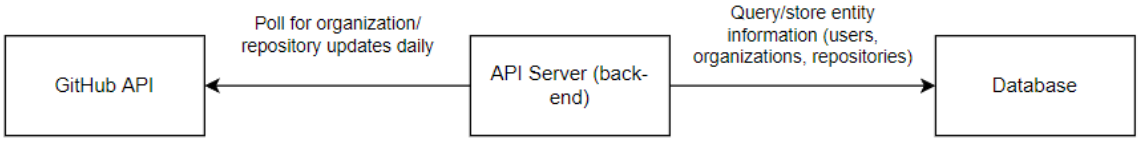
System Architecture Overview

High level Diagram:

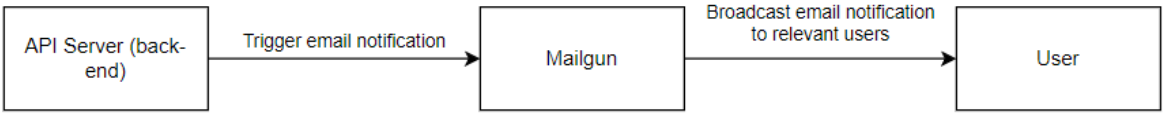


1. User interaction and design

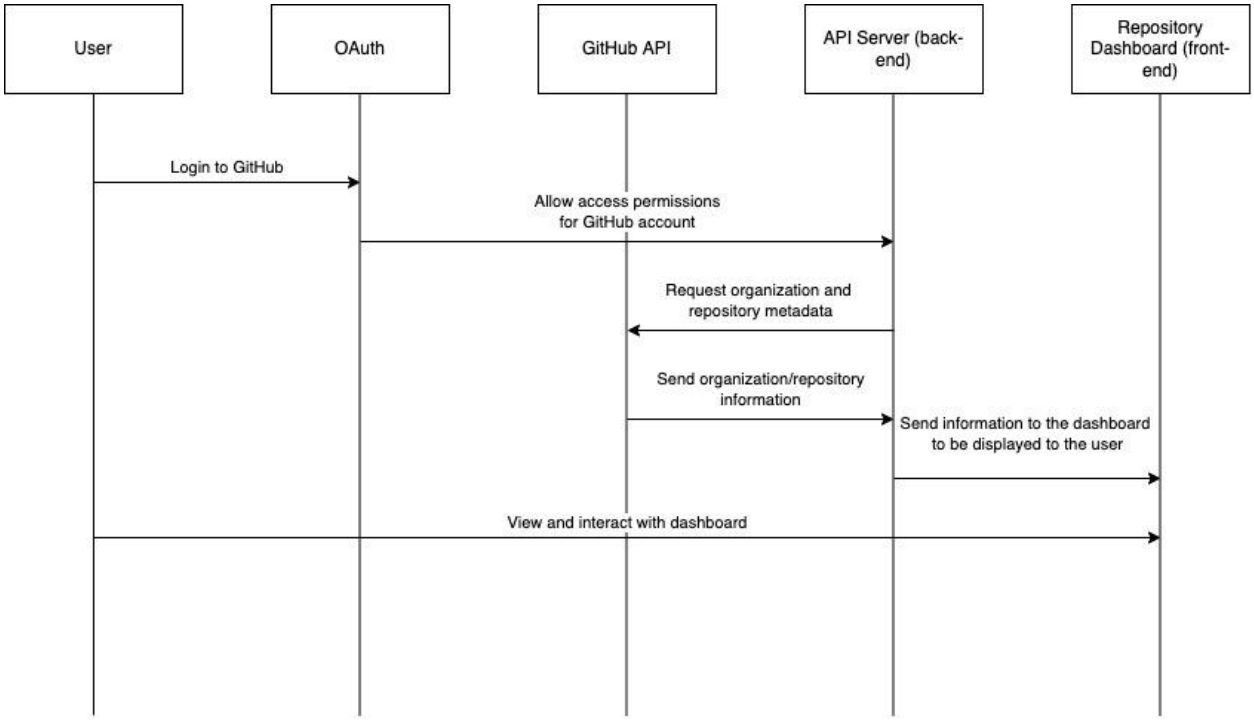
Repository/Organization Data Polling Mechanism



Security Risk Email Notification Mechanism

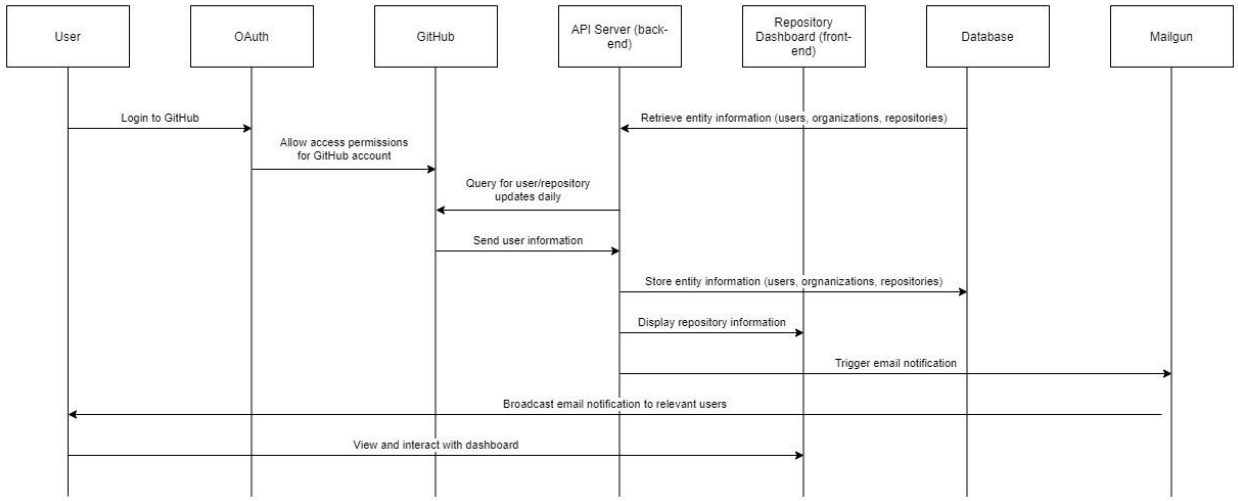


User Login OAuth



System Diagrams

Class Interactions



Requirements

User stories

- As a new user, I am directed to a login page that displays a login button.
    - Trello card
    - **Scenario 1**: The user visits the login page directly.
        - It displays the login button.
    - **Scenario 2**: The user attempts to visit other pages.
        - They are redirected to the login page with a login button.
- As a new user, when I click the login button I can log in through my GitHub account.
    - Trello card
    - **Scenario 1**: The user is logged in to GitHub already.

        - Their credentials are retrieved from the browser and sent to the GitHub OAuth portal, which asks them to give access to GitHub Auditor.
    - **Scenario 2**: The user is not logged in to GitHub.
        - They are redirected to the GitHub OAuth portal, which prompts them to enter their credentials. Upon logging in, they are asked by the GitHub OAuth portal to give access to GitHub Auditor.
- As a logged in user, I am directed to a welcome page.
    - Trello card
    - **Scenario 1**: The user has just logged in.
        - They are redirected to the welcome page.
    - **Scenario 2**: The user has been logged in and visits the Home/Login page.
        - They are redirected to the welcome page.
- As a user, I can navigate among pages using a nav bar.
    - Trello card
    - **Scenario 1**: The user is logged in.
        - They can access Home, Organizations, Settings, and Logout in that order.
    - **Scenario 2**: The user is not logged in.
        - They can access Home/Login. (When not logged in, home is an alias for Login)
- As a logged in user, I can view my organizations on a dedicated page.
    - Trello card
    - **Scenario 1**: User clicks Organizations on the nav bar.
        - They are redirected to the Organizations page which displays their organizations.
    - **Scenario 2**: User directly enters the Organizations page's URL.
        - They can see the Organizations page which displays their organizations.

- As a logged in user, I can choose which organizations of mine to track.
    - Trello card
    - **Scenario 1**: User is at the Organizations page and selects add/remove organizations.
        - They are redirected to the GitHub page where they can manage permissions for the GitHub Auditor application.
    - **Scenario 2**: User is at the Settings page and selects manage organizations.
        - They are redirected to the GitHub page where they can manage permissions for the GitHub Auditor application.
- As a logged in user, I can click on an organization to bring up more detailed information about its security flaws.
    - Trello card
    - **Scenario 1**: User clicks on an organization from the Organizations page.
        - They are redirected to a separate page dedicated to that organization.
    - **Scenario 2**: User directly enters the URL for the organization page they want.
        - They are directed to the page for that organization.
- As a logged in user, I can click on an organization to see information about its individual repositories.
    - Trello card
    - **Scenario 1**: User clicks on an organization from the Organizations page.
        - On the organization's page, its repositories are listed.
    - **Scenario 2**: User directly enters the URL for the organization page they want.
        - They are directed to the page for that organization.
- As a logged in user, I can clearly see if my organization is missing the two-factor authentication requirement.
    - Trello card

    - **Scenario 1**: User is at the Organizations page.
        - Organizations that do not have the two-factor authentication requirement are highlighted with an indicator.
    - **Scenario 2**: User is at the page for a specific organization.
        - The two-factor authentication requirement is clearly listed as enabled or disabled.
- As a logged in user, I can clearly see if my organization has outside contributors.
    - Trello card
    - **Scenario 1**: User is at the Organizations page.
        - Organizations that have outside contributors are highlighted with an indicator.
    - **Scenario 2**: User is at the page for a specific organization.
        - Outside contributors is clearly visible as an attribute of the organization.
- As a logged in user, I can clearly see if my organization has Dependabot alerts.
    - Trello card
    - **Scenario 1**: User is at the Organizations page.
        - Organizations that have Dependabot alerts are highlighted with an indicator.
    - **Scenario 2**: User is at the page for a specific organization.
        - Dependabot alerts is clearly visible as an attribute of the organization.
- As a logged in user, I can clearly see if my organization has pull requests lacking an assigned reviewer.
    - Trello card
    - **Scenario 1**: User is at the Organizations page.
        - Organizations that have pull requests lacking an assigned reviewer are highlighted with an indicator.

    - **Scenario 2**: User is at the page for a specific organization.
        - The existence of pull requests lacking an assigned contributor is clearly visible as an attribute of the organization.
- As a logged in user, I can clearly see if my organization has sensitive information such as API keys or passwords in the code.
    - Trello card
    - **Scenario 1**: User is at the Organizations page.
        - Organizations that have API keys or passwords in the code are highlighted with an indicator.
    - **Scenario 2**: User is at the page for a specific organization.
        - The existence of API keys or passwords in the code is clearly visible as an attribute of the organization.
- As a user, I can choose to be notified about changes in any of the above security flaws via email.
    - Trello card
    - **Scenario 1**: User is at the Settings page.
        - There is a submenu the user can navigate to via a sidebar that allows the user to toggle email notifications for each of their organizations.
    - **Scenario 2**: User is at the page for a specific organization.
        - There is an option to enable or disable email notifications for the given organization.
- As a returning user, I can return to the web application without having to reauthenticate.
    - Trello card
    - **Scenario 1**: User has not logged out.
        - They are not prompted to log in.
    - **Scenario 2**: User has logged out and logs in.

        - They are not redirected to GitHub OAuth or prompted to authorize GitHub Auditor to access their GitHub account.
- As a user, I want this security information to be relatively up-to-date so I have a realistic overview of my organizations' security status.
    - Trello card
    - **Scenario 1**: User does not log in for a while.
        - The application passively polls the GitHub API at intermittent intervals (at least daily) to detect any changes, so that the user does not have to directly interface with the application to receive email notifications regarding changes.
    - **Scenario 2**: User logs in to the web application.
        - Upon the user logging in, the application calls the GitHub API to instantly refresh its data.
- As an admin, I can see all organizations tracked by GitHub Auditor.
    - Trello card
    - **Scenario 1**: A user is logged in as an admin.
        - They have an additional item in the navbar, All Organizations, where they can view all the organizations tracked by GitHub Auditor.
    - **Scenario 2**: A user is not logged in as an admin.
        - They are only able to see their own organizations.
- As an admin, I can see all users tracked by GitHub Auditor.
    - Trello card
    - **Scenario 1**: A user is logged in as an admin.
        - They have an additional item in the navbar, All Users, where they can view all the users tracked by GitHub Auditor.
    - **Scenario 2**: A user is not logged in as an admin.

        - They are only able to see themself as a user.
- As an admin, I can see which users belong to which organizations.
    - Trello card
    - **Scenario 1**: A user is logged in as an admin and visits the All Organizations page.
        - Upon selecting an organization, they can see all the users that belong to that organization and are being tracked by GitHub Auditor.
    - **Scenario 2**: A user is not logged in as an admin.
        - They are only able to see the organizations they belong to.
- As an admin, I can see which organizations belong to which users.
    - Trello card
    - **Scenario 1**: A user is logged in as an admin and visits the All Users page.
        - Upon selecting a user, they can see all the organizations that belong to that user and are being tracked by GitHub Auditor.
    - **Scenario 2**: A user is not logged in as an admin.
        - They are only able to see their own organizations.

Workflow

- GitHub: Feature Branch workflow
    - Feature development takes place in a dedicated branch outside of the main branch.
    - The main branch never contains broken code.
- Local testing
    - ngrok for local testing of remote integrations
- Heroku deployment for staging/production
    - Separate environments for staging and production

Appendix

Technologies employed

Front End: React.js

Back End: Ruby on Rails

Database: PostgreSQL

Redis, GitHub OAuth/API, Mailgun