Overview

The goal of this project is to create a conversation similar to one you would experience during a typical day working at Contrast Security as an engineer. You've made a pull request (PR) to a repository and are going through the code review process. It's the primary way we work from the infrastructure all the way through the front-end code.

The Project

- Create an account for our community edition by clicking "Create Free Account" and filling in the form at https://www.contrastsecurity.com/contrast-community-edition . Check your email to verify your account and complete the registration process. Any problems you encounter are candidates for creating a defect ticket in your repo!
- When you first login, it should present you with an onboarding flow and at some point, be asked if you want to view a sample/demo application or get to installing an agent. Choose the demo application as it will be pre-populated with data for you to explore.
- From here, explore the application and familiarize yourself with it. Our general docs can be found at https://docs.contrastsecurity.com and our API docs are https://api.contrastsecurity.com/

The Tasks

Finding Issues

Report one or more issues in writing (e.g., a "defect ticket") such that an engineer is likely to repair the problem. You can include these in your project as an additional text or markdown file.
Create one or more automated tests to show the issue.

It is important to note that time is a factor. We don't want you to spend a day finding a big juicy bug. Indeed, if you get to a reasonable time and you have not found a defect, create an artificial scenario that allows you to write the test automation you'd like to show off. Your code is far more interesting in this exercise than any defect you might find.

What is important is how you communicate the defect in writing. Things like reproducing the defect, specifying how it should work are ideal.

Automation

Besides the automated test showing your defect, write one or more tests that validate tagging a vulnerability, and then be able to filter by the newly added tag (see https://docs.contrastsecurity.com/en/tag-vulnerabilities.html for more info about tagging). It will be up to you how much you want to expand upon this.

You are welcome to write tests that directly exercise the UI, however we are specifically interested in your ability to work with our API. Please ensure you write one or more tests that exercise the API in a way that shows off the defect.

For these tasks, you have complete freedom to work the project as you see fit.

Completing the Assignment

When you've completed the tasks, create a repository on Github and push up your work. If it's private, add the users thekiiingbob and gibbsjosh, or if it's public just provide us the URL.

Important: we expect this project will take 4 to 6 hours. You are free to use more or less time.

Bonus Points

We work in a remote environment. High fidelity written communication is the key to maintaining velocity. Showing off your ability to write, provide screenshots, and expose your assumptions will go a long way.

Lastly, if you have questions please reach out to Bob Lubecker at robert.lubecker@contrastsecurity.com