Practice Lab: Security Vulnerability Scan and Fix

Estimated time needed: 30 minutes

In this lab, you will learn how to check code on GitHub for vulnerabilities in order of severity and fix the vulnerabilities.

Learning Objectives

After completing this lab, you will be able to:

- Perform a vulnerability scan on your code
- Internalize best practices for reducing the risk of vulnerability
- Fix the vulnerabilities in the code

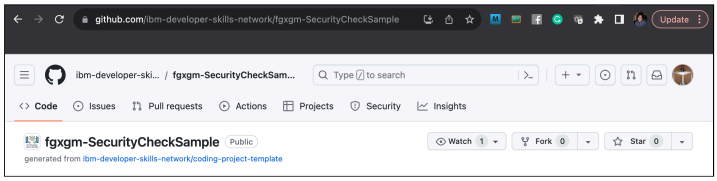
Prerequisites:

You must have a GitHub account as well as a Snyk account that is authenticated to use your GitHub account. Please ensure you have completed this lab, before you proceed.

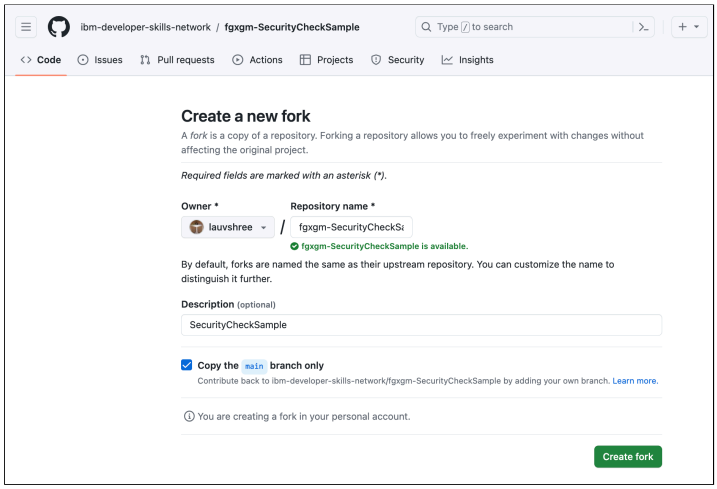
Task 1: Get a repo copy

1. Go to https://github.com/ibm-developer-skills-network/fgxgm-SecurityCheckSample.git.

2. Fork the repository (or repo) to get your own copy of the repository. Remember that you can only scan your own repositories or public repositories. This repository has a simple server side application that is to be dockerized and deployed on the cloud.



3. You are asked to confirm about creating the fork. Read the details and confirm the action by clicking `Create Fork`.

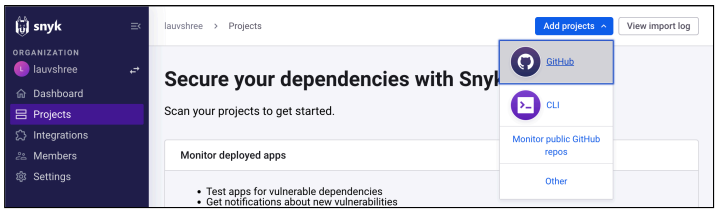


Now you have your own copy of the repository, and any changes you make to this will not change the source you copied from.

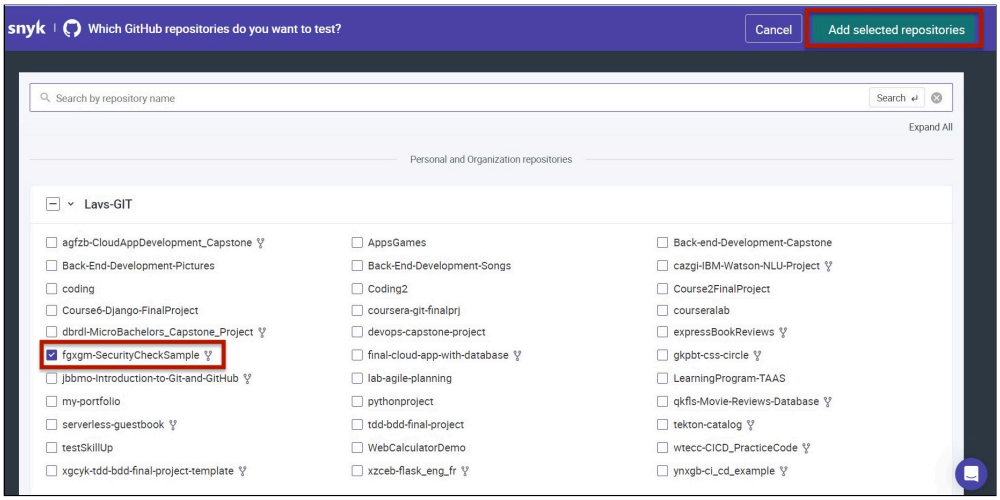
Task 2: Scan the repo

1. Now that you have your own copy of the repository, go to https://app.snyk.io/login and log in with your GitHub credentials.

2. On Snyk, click `Add Projects` and choose `GitHub` as the source of the project to add the repository that you have forked.



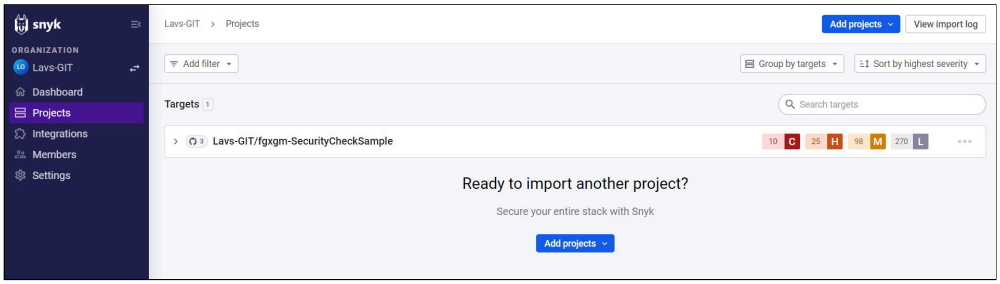
3. Select the repository your forked and click `Add Selected Repositories`.



4. Once the project is added, it is imported and scanned by Snyk.

This may take a few seconds. Once the scan is done, you will see a report of the scan.

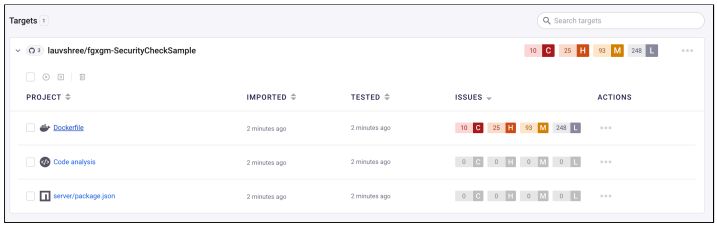
5. Once the scan is completed, you will see an issue report as below.



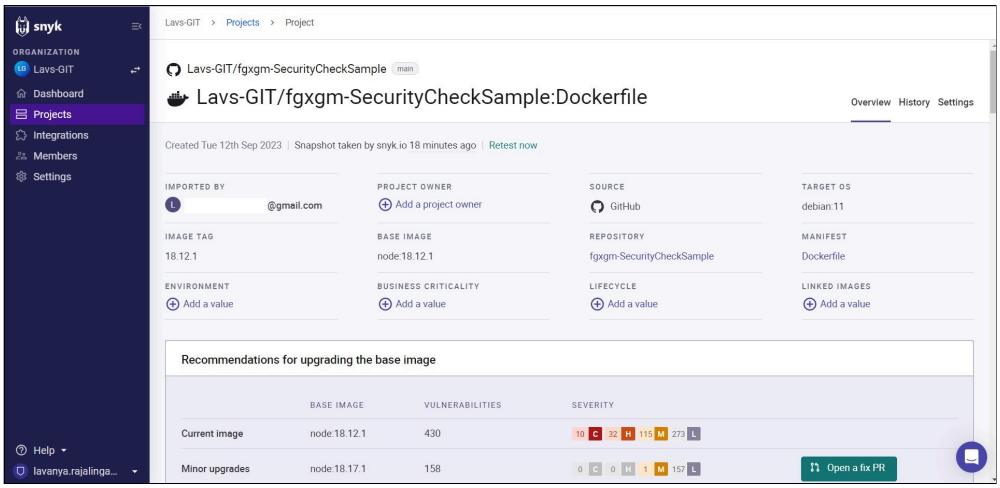
Task 3: View scan report

1. Click the arrow before the repo name to see a detailed report which tells where the vulnerabilities are and the severity of the vulnerabilities. There are 4 different vulnerabilities shown for each file:

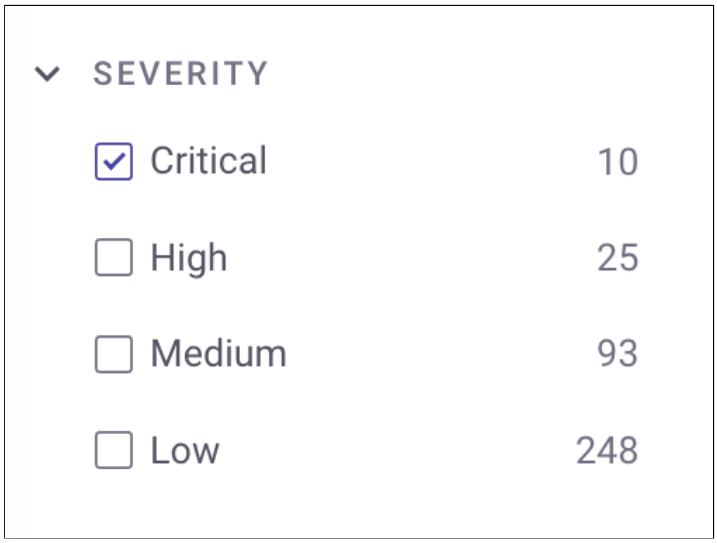
- Critical
- High
- Moderate
- Low



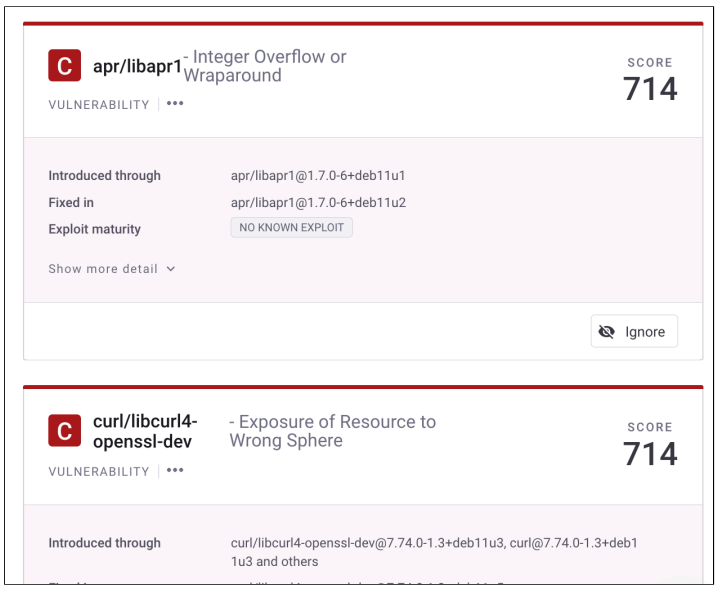
2. Click the `Dockerfile` to see the vulnerabilities that have been scanned.



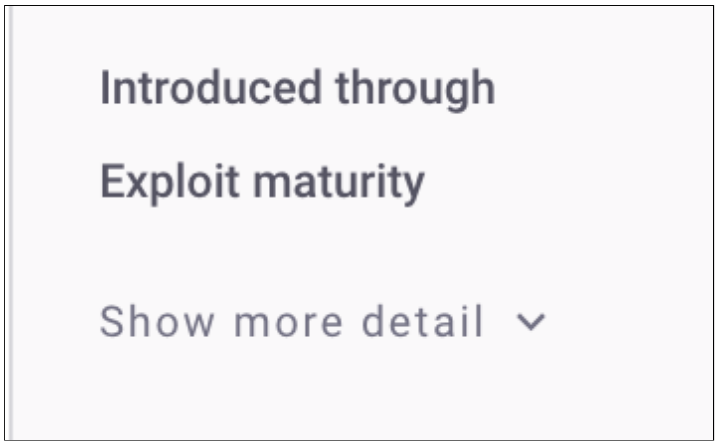
3. The result can be filtered on the basis of the severity to get a detailed view of a particular kind of vulnerability.



4. You can look at what errors are listed and what kind of vulnerability or security risk it can be.



5. For each error, you can see a detailed report which advises on the repercussions of the error and also suggests how the error can be rectified.



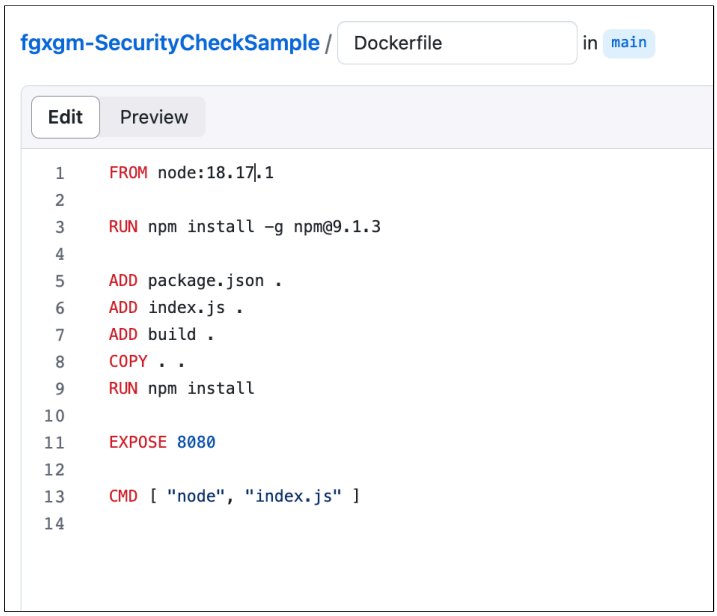
6. If you scroll up the page, there will mostly be a big recommendation that you can follow to get rid of most errors. This will be on the basis of a recent version where most issues are fixed.

7. As seen in the above example, it suggests upgrading the node version from 18.12.1 to 18.17.1.

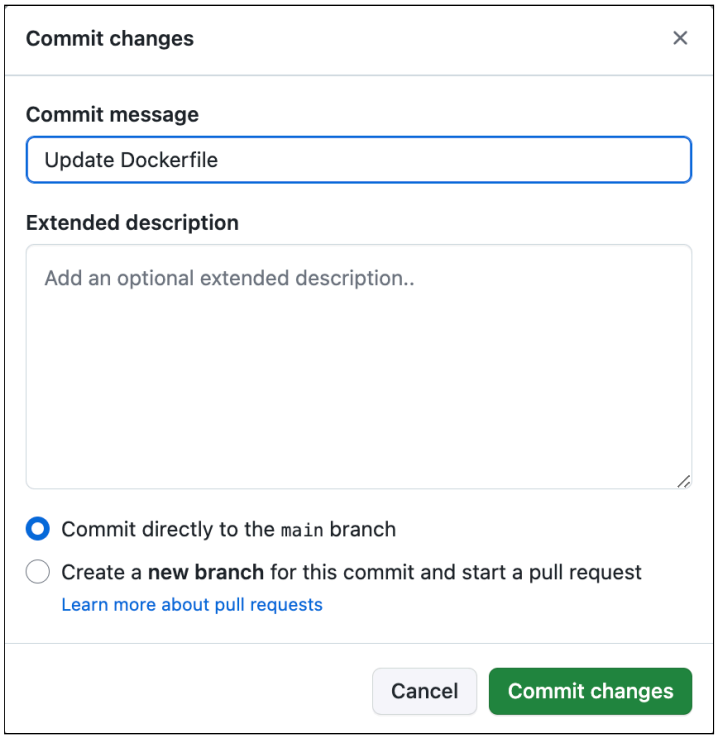
Task 4: Fix vulnerabilities

1. Go back to the repository on GitHub. Click `Dockerfile` to open and view it.

2. Click the pencil icon on the top right to edit it and change the node version to `18.17.1` as recommended.

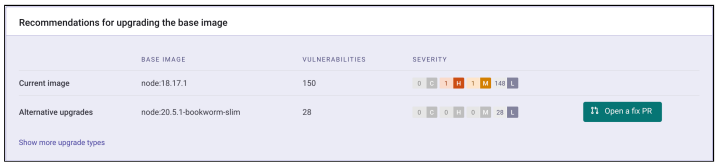


3. Commit the changes to make them permanent in your copy of the repository.



Task 5: Verify vulnerabilities fix

1. Go back to the browser tab where you did the Snyk scan. If you have closed the tab, feel free to open a new one.
2. The scan will re-run, and an updated report will be available.
3. Check for new recommendations, if any.



4. Make changes on Git and commit.
5. Iteratively, keep making recommended changes and check the project report again. When there are no C, H, or M issues, it is considered good to go.

Task 6 (Optional): Fix low-priority issues

1. Check if, based on the recommendation, you can fix the low-priority issues that are marked L.

Congratulations! In this lab, you learned how to scan a git code repository for vulnerabilities and fix the vulnerabilities.

Author(s)

Lavanya T S