Sprint 2 Review Report
USSF GRC Controls - Kubernetes

Sprint Dates:

October 14th, 2024 - October 18th,2024 :- Sprint 2
October 21st, 2024 - October 25th,2024 :- Sprint 2 Retrospective and Sprint 3 Planning

Links to Important Resources:

Heroku App: app link
GitHub Repository: USSF-CSCE606
GitHub Project: USSF

Team Roles:

- **Scrum Master** Medha Kaushika Podipireddi
- **Product Owner**: Sahil Fayaz
- **Developers:** Duy Vu, Maitreya Niranjan, Shravan Bhat, Vasudha Devarakonda, Aditya Gourishetty

Sprint Goal:

With the prototype approved by the client after Sprint 1, the goal of Sprint 2 was to focus on modifying the UI to enhance user experience. This included the display of inferences from the GRC reports to help users make decisions about deployment, visual consistency across all views, and adding pagination to the homepage for easier navigation. Additionally, the focus was also on doing security due diligence for the third-party library Trivy and providing functionality to rerun scans for images allowing users to stay updated on vulnerabilities.

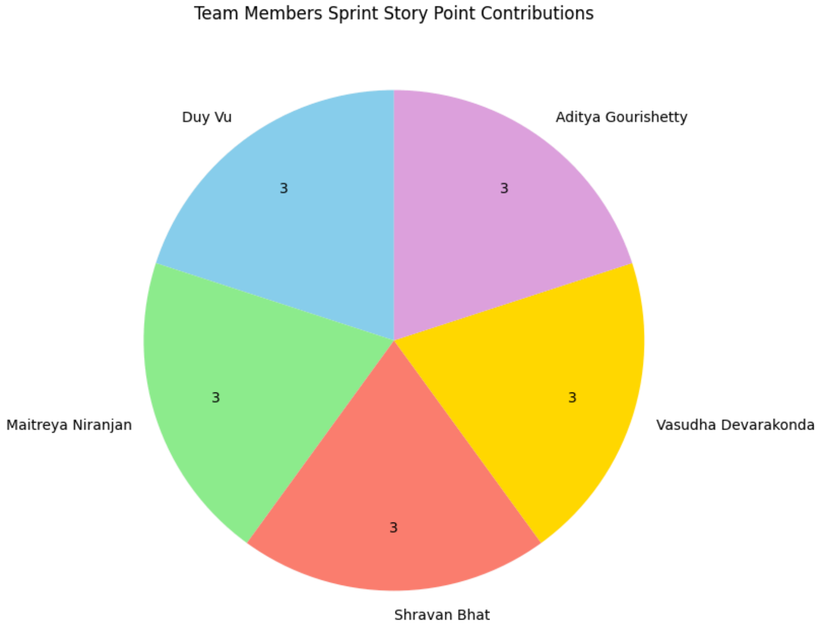
Sprint Summary and Achievements:

We completed all our goals and user stories for this sprint. We have a working application with all the core functionality embedded into it. We have enhanced the UI and navigation for better user experiences and also derived inferences from the generated output by the third-party security scanner, Trivy. Additionally, we also performed security due diligence on the Trivy to ensure the inputs that we have are securely getting scanned. We presented the same to the USSF clients and they have agreed upon the design and features to be in alignment with their idea of the solution to the problem statement. The application is currently hosted on Heroku. We have also achieved good code quality having used rubocop, rubycritic, and code coverage for rspec and cucumber test cases. All the committed stories for the sprint have been marked as done.

Story	Story Points	Sub Tasks	Sub Task Points	Assigned To	Status
Derive & Show Inferences from the GRC Report	6	Enhanced Vulnerability Report Page with Filters and Improved Layout	3	Aditya	Done ✅
		Vulnerability Analysis and Deployment Decision with Trivy-NIST Mapping	3	Shravan	Done ✅
Security Due Diligence of Trivy	2			Maitreya	Done ✅
Pagination in Homepage	3	Implement the pagination controls in the view. Show the paginated images.	2 1	Duy	Done ✅
Enhancing the UI of the application	3			Vasudha	Done ✅
Rerun scan for the image	1			Maitreya	Done ✅

See the Appendix for story descriptions.

Team member contributions:



Duy Vu: Implemented pagination controls in the home view and displayed the paginated images which allowed users to easily navigate through the multiple pages.

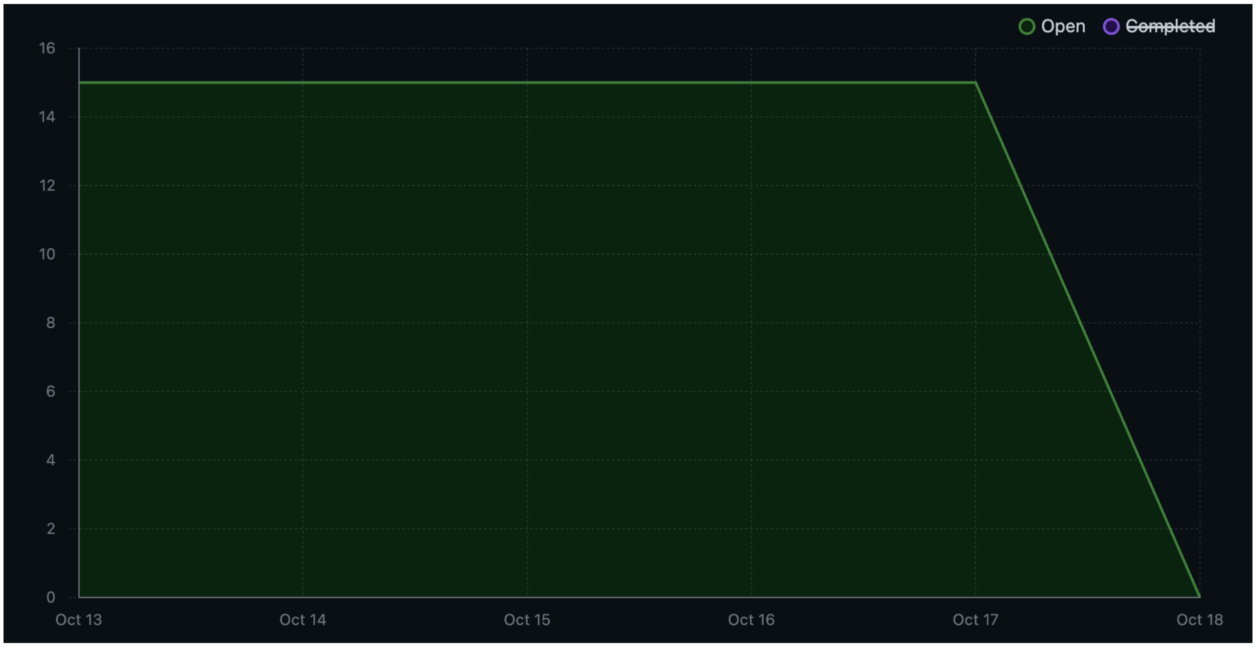
Maitreya Niranjan: Looked into security due diligence of the third party library, Trivy and created a detailed report of it. Also implemented the rerun scan button for existing images.

Vasudha Devarakonda: Worked on enhancing the application's UI by adding a consistent header across all pages and improving the CSS for better styling and responsiveness.

Shravan Bhat: Worked on mapping the vulnerabilities generated by Trivy to NIST SP 800-53 GRC framework. With this user can see more details of vulnerabilities in the CSF portal.

Aditya Gourishetty: Worked on improving the UI of the report page by adding filter options. Also presented a summarized view of the vulnerabilities identified.

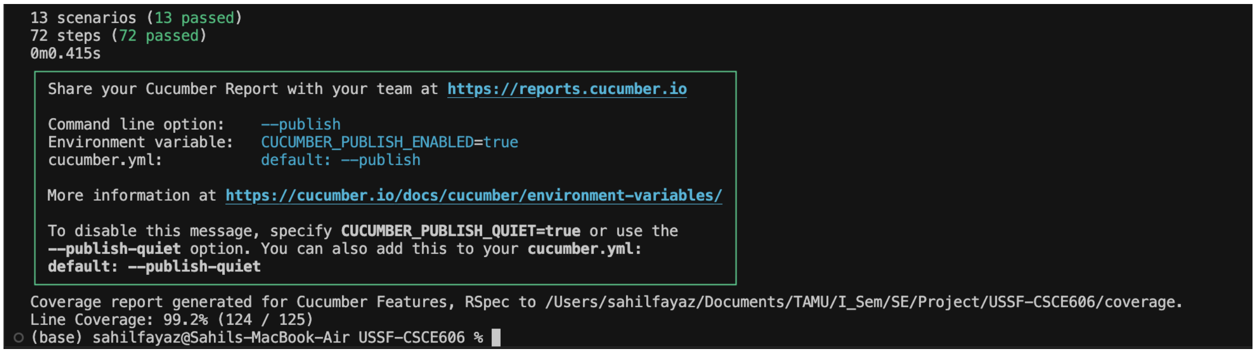
Burn Down Chart:



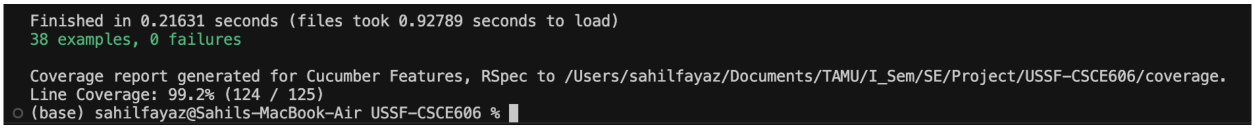
The y-axis represents the story points committed for the sprint.

Code Evaluations:

Rspec and Cucumber Coverage:

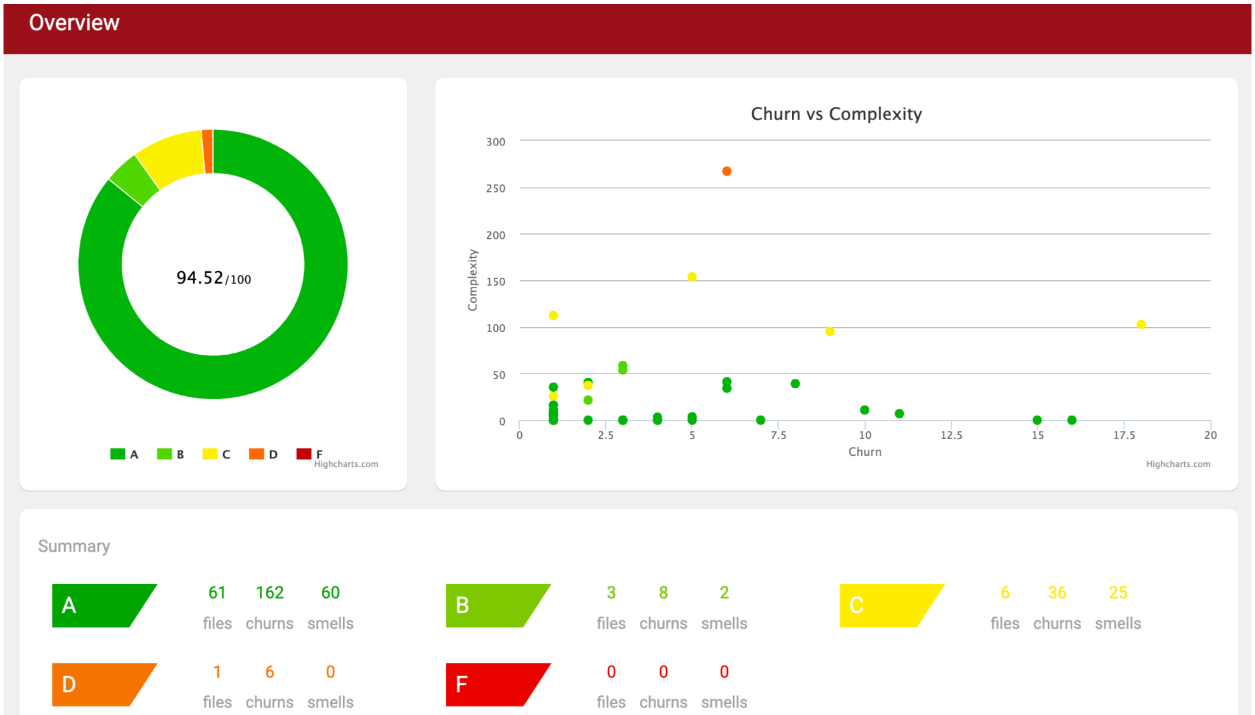


```
13 scenarios (13 passed)
72 steps (72 passed)
0m0.415s
 Share your Cucumber Report with your team at https://reports.cucumber.io

 Command line option:    --publish
 Environment variable:   CUCUMBER_PUBLISH_ENABLED=true
 cucumber.yml:           default: --publish

 More information at https://cucumber.io/docs/cucumber/environment-variables/

 To disable this message, specify CUCUMBER_PUBLISH_QUIET=true or use the
 --publish-quiet option. You can also add this to your cucumber.yml:
 default: --publish-quiet
Coverage report generated for Cucumber Features, RSpec to /Users/sahilfayaz/Documents/TAMU/I_Sem/SE/Project/USSF-CSCE606/coverage.
Line Coverage: 99.2% (124 / 125)
○ (base) sahilfayaz@Sahils-MacBook-Air USSF-CSCE606 %
```



```
Finished in 0.21631 seconds (files took 0.92789 seconds to load)
38 examples, 0 failures

Coverage report generated for Cucumber Features, RSpec to /Users/sahilfayaz/Documents/TAMU/I_Sem/SE/Project/USSF-CSCE606/coverage.
Line Coverage: 99.2% (124 / 125)
○ (base) sahilfayaz@Sahils-MacBook-Air USSF-CSCE606 %
```

RubyCritic:



Overview

Summary

A	61 files	162 churns	60 smells	B	3 files	8 churns	2 smells	C	6 files	36 churns	25 smells
D	1 files	6 churns	0 smells	F	0 files	0 churns	0 smells				

Client Sprint MVP meeting information:

Date: 10/24/2024
Time: 18:30
Location: Google Meet
Attendees: Aditya Gourishetty, Maitreya Niranjan, Duy Vu, Prof. Shreyas Kumar, Medha Kaushika Podipireddi, Sahil Fayaz, Shravan Bhat, Vasudha Devarakonda, Lt. Theresa Kopecky, Major Eric Griffin

- The team demonstrated the deployed application to the client including the UI and scanning workflow and the different pages of the application.
- Security due diligence of Trivy was also communicated to the client.
- The client gave the following feedback:
  - The USSF representatives agreed on the design and workflow stating that the solution we provided was in alignment with their idea.
  - Additional requirements:
    - One page overview about building a service like Trivy that includes details like estimated time of building such service, complexities involved in it, etc.,

        - How is our application different from any other automated vulnerability scanning applications available?
        - What is the scalability of the application and Trivy(third-party scanner)?
        - Can our application be integrated into the dev pipeline in its current stage or any other fine-tuning required for the purpose?

Recording Link :

Demo_1.mov

Appendix:

Story Descriptions:

Feature: Derive & Show Inferences from GRC Report

- **As a user** of the application
- **So that** I need to be able to view the security analysis of the run-time object on which the controls are getting evaluated and decide whether to go ahead with the deployment of the run-time object securely
- **I want** a page where I can see the details of the different vulnerabilities and their severity along with the details of the vulnerability.

Feature: Security Due Diligence of Trivy

- **As a stakeholder** of the application
- **So that** I can ensure the third-party library Trivy does not have any security vulnerabilities
- **I want** a thorough evaluation of Trivy including an assessment of potential vulnerabilities and risks

Feature: Pagination in Homepage

- **As a user** of the application
- **So that** I can easily navigate through large number of images in the homepage
- **I want** the homepage to include pagination allowing me to view images in smaller and manageable sections

Feature: Enhancing UI of the application

- **As a user** of the application
- **So that** I have a better experience while using the application
- **I want** all the pages/views to be visually consistent with all logos and stylings

Feature: Rerun scan for the image

- **As a user** of the application
- **So that** I can ensure the latest security vulnerabilities are detected for that particular image
- **I want** the option to rerun the scan for the selected image and view updated scan results.