

Automated Vulnerability Detection and Analysis with SASTER

Challenge Title: Automated Vulnerability Detection and Analysis with SASTER

Challenge Context and Purpose: SASTER is a cloud-based platform designed to automate the detection and analysis of security vulnerabilities in software projects. It offers a unified API and web interface that allows users to upload source code, run multiple static analysis modules and visualize results in a centralized dashboard. The purpose of this challenge is to demonstrate how SASTER can streamline secure software development by automatically identifying weaknesses in source code and assisting developers in improving their security posture. Participants will interact with SASTER in a guided environment to evaluate its ease of use, coverage and analysis report. This exercise aims to validate SASTER's role in helping software teams adopt proactive security testing practices as part of their DevSecOps workflow.

Goals of the Challenge: What do you aim to achieve by including this challenge?

- Assess the usability and effectiveness of SASTER's web interface.
- Evaluate the accuracy and clarity of vulnerability reports produced by the static analysis modules.
- Gather feedback on the integration potential of SASTER into CI/CD pipelines.
- Identify usability or visualization improvements that can enhance developer experience.

Key Tasks for Participants: List the main tasks participants will perform to test and evaluate the solution.

- Register and log in to the SASTER web interface.
- Create a new project and upload a provided codebase (e.g., sample C/C++/Python/Java application).
- Execute static vulnerability scans using available analysis modules.
- Submit short feedback on usability, report clarity and suggested improvements.

Feedback and Evaluation Requirements: Describe the feedback participants are expected to provide.

- An evaluation for SASTER's interface, clarity of results and ease of navigation.
- Observations on false positives, report readability and workflow efficiency.
- Suggestions for additional features or integrations (e.g., GitHub/GitLab support, report export formats).

Deliverables: Clearly state what participants are expected to submit.

- Completed evaluation form.
- Short notes or recommendations document on usability and functionality improvements.

Resources and Tools Provided:

- Access to SASTER via the CyberSuite platform.
- Sample project code for the vulnerability analysis.
- Predefined demo application for testing.

Estimated Time Requirement: ~ 20 minutes

Contact Point (Optional): Konstantinos Giapantzis (konstantinos.giapantzis@inqbit.io)

Instructions:

1. Register a new user

1. Open the SASTER site and click **Register**. (Website: <https://saster.inqbit.eu>)
2. Complete the registration form with the required information.
3. Click **Sign up** to create your account.

Tip: If you only want to test quickly, use one of the demo accounts below.

Demo accounts (for hackathon/testing)

Username: **Test** — Password: **testtest**

Username: **demo_user** — Password: **demodemo**

2. Log in

1. From the navigation bar, click **Login**.
2. Enter your username and password.
3. Click **Login**.

3. Create a new project

1. From the main page, click **Projects**.
2. Click **Create new project**.
3. Enter your Project Name in the appropriate field.
4. Click **Create** to add the project.

4. Upload code for analysis

Vulnerable example code (optional):

<https://drive.google.com/drive/folders/1wborl75swuHouL8wBtM8rldP6Q3ckG0a?usp=sharing>

1. Go to the Projects page — a list of your projects will be displayed.
2. Each project row has **four action** buttons. Click the **Upload** action for the project you want to analyze.
3. In the upload dialog, **drag-and-drop** your code or **browse** to select files.

(Optional) You can use the vulnerable example code from the Google Drive link above for testing.

5. Run analysis and review results

1. On the Projects page, find the project you uploaded code to and click the **Analyze** action.
2. The analysis runs in the background. Wait a few minutes for the scan to complete.
3. Return to the project list and click the **View Result** action to inspect the vulnerabilities discovered by SASTER.

Troubleshooting & Tips

- If analysis seems to hang for too long, refresh the Projects page after a minute or two to check status.
- Ensure uploaded files are within the allowed size and file-type limits.
- Use the demo accounts for quick access without registering.