**Juice Shop**

**SAST:**

Allow\_failure: true

Job is allowed to fail, since we only want to catch the vulnerabilities and don’t want to block the entire pipeline for it.

artifacts: always

it always generates artifacts even if the job is cancelled or failed

reports:

sast: semgrep-results.json

the file is declared as SAST job report so it can be stored and found in gitlab under the job called sast.

The file is formatted in .JSON so that we process it in our Vulnerability dashboard

expire\_in: 3 weeks

was done this way in case we don’t run our VD days or hours before the presentation it has enough time before the job artifacts expire. Since expired artifacts was not possible to pull from gitlab API

image: beshoynfarag/semgrep-juiceshop:latest

a costume docker image, uploaded on one of our team mates dockerhub. It includes the semgrep and the code for the juice shop ready for scanning.

semgrep --config p/ci --json -o semgrep-results.json juice-shop-master/ || true

semgrep: Runs the Semgrep tool.

--config p/ci: Tells Semgrep to use the predefined ruleset p/ci from the Semgrep Registry, optimized for CI pipelines (common security checks).

--json: Outputs the scan results in JSON format (machine-readable).

-o semgrep-results.json: Writes those results to a file called semgrep-results.json.

juice-shop-master/: Specifies the directory to scan — in this case, the source code of the Juice Shop app.

|| true: This ensures that even if Semgrep finds issues or returns a non-zero exit code, the job doesn't .

**Git-Leaks:**

gitleaks\_stage:

the job name and id in the pipeline

stage: sast\_stage:

the name of the stage that gitleaks is a part of

image: docker:24

services:

- docker:24-dind

The job uses docker image in the pipeline, uses docker image version 24

variables:

CI\_REGISTRY\_USER: beshoynfarag

CI\_REGISTRY\_PASSWORD: dckr\_pat\_H9qEvw8YgvWnAKPaAj6xTYCdUZk

DOCKER\_HOST: tcp://docker:2375

DOCKER\_TLS\_CERTDIR: ""

This declares variables that will be later on used for communication with the daemon and with dockerhub.

- docker pull zricethezav/gitleaks:latest

Pulls the latest git-leaks image from docker

- |docker run --rm -v "$CI\_PROJECT\_DIR":/src -w /src zricethezav/gitleaks:latestdetect --source=. --report-path=gitleaks-report.json || true

|| true at the end to make the job exists successfully no matter the scanning results

--report-path=gitleaks-report.json: Outputs results to a file.

artifacts: when: always

even if the job is canceled or failed always artifacts will be created.

expire\_in: 4 weeks

again for the same reason this one is stored for 4 weeks in case we don’t run the pipeline the artifacts does not expire, which will corrupt the .json and the vulnerability dashboard