**JuiceShop DAST Implementation**

**Stage: dast\_juice\_shop\_stage – Dynamic Application Security Testing (DAST) for OWASP Juice Shop**

**YAML Breakdown and Justification**

dast\_juice\_shop\_stage:

stage: dast\_stage

* Declares the job name and assigns it to the dast\_stage phase. This categorizes all runtime security scanning jobs together in the pipeline.

**Image and Docker Service Setup**

image: docker:latest

services:

- docker:dind

* **docker:latest**: The job runs inside a Docker environment with the CLI available.
* **docker:dind**: Stands for Docker-in-Docker, allowing the container to execute and manage other Docker containers. Required here to run the ZAP container.

**Environment Variables**

variables:

DOCKER\_HOST: tcp://docker:2375

DOCKER\_TLS\_CERTDIR: ""

DOCKERHUB\_USERNAME: beshoynfarag

DOCKERHUB\_TOKEN: dckr\_pat\_H9qEvw8YgvWnAKPaAj6xTYCdUZk

* **DOCKER\_HOST**: Connects the container to the internal Docker daemon.
* **DOCKER\_TLS\_CERTDIR**: Disabled to simplify Docker communication in non-production environments.
* **DOCKERHUB\_USERNAME / TOKEN**: Used for authenticating to Docker Hub in case the image is private or to avoid rate limits when pulling images.

**Artifacts**

artifacts:

paths:

- juice-dast-report.html

* Saves the DAST report as a CI artifact, allowing developers and security teams to download and review it via the GitLab UI.

**Script: Step-by-Step Breakdown**

script:

- echo "$DOCKERHUB\_TOKEN" | docker login -u "$DOCKERHUB\_USERNAME" --password-stdin

* Securely logs in to Docker Hub using an access token. This is necessary if you're pulling from a private repo or pushing images.

- docker pull beshoynfarag/zap2docker-stable:latest

* Pulls a custom or mirrored image of the official OWASP ZAP tool (assumed to be customized or cached by the user for local improvements or faster pulls).

- |

docker run -v $(pwd):/zap/wrk/:rw beshoynfarag/zap2docker-stable \

zap-baseline.py -t http://10.97.7.109:3000 -r juice-dast-report.html || true

* Runs the OWASP ZAP baseline scan in a container:
  + -v $(pwd):/zap/wrk/:rw: Mounts the working directory into the container, allowing ZAP to write the report (juice-dast-report.html) back to the host.
  + zap-baseline.py: A ZAP script that performs a quick passive scan (non-intrusive), ideal for CI pipelines.
  + -t http://10.97.7.109:3000: Target application URL. This must be a **live and reachable** instance of Juice Shop.
  + -r juice-dast-report.html: Output report name.
  + || true: Ensures that the pipeline does not fail if vulnerabilities are found. Instead, it logs them for review.