**DAST-ZAP todoapp Implementation**

Briefly what the steps that was done to get the analysis for the todoapp using dast-zap tool:

* Run the app (todoapp) on our docker on our ubuntu VM, by building the container.
* Make sure that the docker’s java is using the same version as the apps’s maven and spring boot
* Go to GitHub and create a PAT here:  
  <https://github.com/settings/tokens/new>
* Give it a name (e.g., GHCR Access)
* Select the scope: **read:packages** (minimum)
* Generate the token and copy it.
* Log in to GHCR on your VM:
* Clone the open-source docker repo for dast-zap from github
* Download dast-zap docker container
* Run the image for dast-zap, “zap2docker-stable”
* Run the command: docker run -u zap -p 8090:8090 zap2docker-stable zap.sh -daemon -port 8090 -host 0.0.0.0 -config api.disablekey=true
* Add our machines local host to the container Tree of zap
* Then run these commands

# start spidering

* curl "http://localhost:8090/JSON/spider/action/scan/?url=http://localhost:8080&maxChildren=10"

# Check spider status

* curl "http://localhost:8090/JSON/spider/view/status/"
* # Start active scan (once spider is done)
* curl "http://localhost:8090/JSON/ascan/action/scan/?url=http://localhost:8080"

# Check active scan status

* curl "http://localhost:8090/JSON/ascan/view/status/"

# Get alerts (vulnerabilities found)

* curl <http://localhost:8090/JSON/core/view/alerts/?baseurl=http://localhost:8080>

afterwards download it as JSON on the machine or as html. Push it to gitlab.

You can read it as JSON or copy the html code into an online html compiler to see the results