Docker

## Summary

This lab introduces running basic Docker commands and running the OWASP Threat Dragon container.  To do you will run a interactive Ubuntu container.  Those new to Docker might find this resource helpful: <https://training.play-with-docker.com/beginner-linux/>

## Tools

Parrot: Docker

Websites:

<https://hub.docker.com/>

<https://labs.play-with-docker.com/>

## Tasks

* Configure docker.io as an apporved repo
* Configure the Threat Dragon environment
* Run Threat Dragon container
  + Show screenshot of docker command and timestamp
* Show Threat Dragon home page

## Submission

Upload to Canvas a pdf or docx with two screenshots

## Walkthrough

1. Configure environment

# Add following line to /etc/containers/registries.conf

unqualified-search-registries = ["docker.io"]

1. Not part of assignment but you run some practice docker commands for the class. And/or show: https://training.play-with-docker.com/beginner-linux/

docker container run --interactive --tty --rm ubuntu bash

docker container ls --all

docker ps -a

Pull, run, modify, list, kill HelloWorld from dockerhub.com (some exemples)

docker pull hello-world

docker run hello-world

docker container prune

1. Configure environment for Threat Dragon

cat <<EOF >.env

NODE\_ENV=development

ENCRYPTION\_KEYS='[{"isPrimary": true, "id": 0, "value": "deadbeef2233445566778899aabbccdd"}]'

ENCRYPTION\_JWT\_SIGNING\_KEY=deadbeef112233445566778899aabbcc

ENCRYPTION\_JWT\_REFRESH\_SIGNING\_KEY=deadbeef00112233445566778899aabb

SERVER\_API\_PROTOCOL='http'

EOF

1. Run Threat Dragon container

docker run -d -p 8080:3000 -v $(pwd)/.env:/app/.env owasp/threat-dragon:stable

docker container ls --all

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Clean up Docker containers

docker ps -a -q

docker stop $(docker ps -a -q)

docker rm $(docker ps -a -q)

- Submit screenshots of terminal and ThreatDragon main page.

## Additional Thoughts

How many containers can be associated with a single image?

How many images can be associated with a single container?

A good docker cheatsheet?

Docker swarm or Kubernetes?