National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Adarsh Sharma
Student ID: 20140207

School of Computing
National College of Ireland

Supervisor: Divyaa Elango

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Adarsh Sharma
Student ID:	20140207
Programme:	MSc in Cloud Computing **Year:** 2021
Module:	Research Project
Lecturer:	Divyaa Elango
Submission Due Date:	16- December- 2021
Project Title:	Eliminating Misconfiguration and Privilege Escalation in Docker Images
Word Count:	..... **Page Count:** .....

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: .....

Date: 16- December- 2021 .....

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Adarsh Sharma
20140207

1 Introduction

This setup instructions for the research have been divided into components. This manual includes detailed description of the whole implementations. The following subsections have been organized for ease of understanding are listed below:

- Setting up of Virtual Machine
- Installation and Configuration of Docker.
- Setup of AIA (Automatic Image Analyzer)
    - Setup of Anchore Engine tool
    - Configuration of module for result management.
- Direct Privilege Escalation Assault

2 Configuring Virtual Machine

I have carried both investigations in this study on a virtual computer. The primary purpose for utilizing a virtual instance is to minimize the security concerns associated with experimenting. VMware Workstation 16 Pro is utilized to meet the requirements (Figure 1). The VM's operating system is Linux Ubuntu 20.04.1 LTS and 4 GB of memory is allocated to the virtual machine with 100 GB of hard disk.

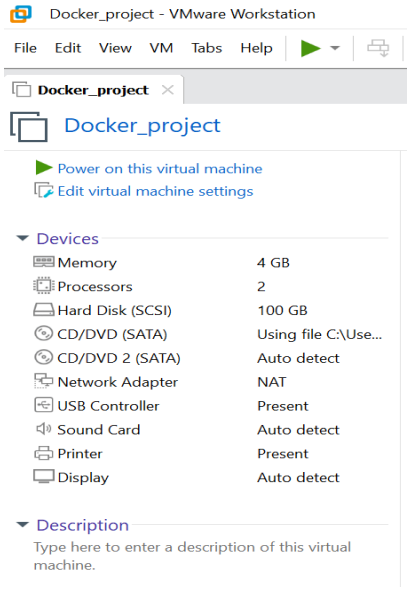


Figure 1- Virtual Machine

3 Configuration of Docker

The Docker Engine utilized for the configuration was configured with the default parameters. There are no additional plugins loaded for any value-added security objectives.

Please find the step below to install Docker in your local host

```
1-   $ sudo apt-get update
2-   $ sudo apt-get install \ ca-certificates \ curl \ gnupg \ lsb-release
3-   $ sudo apt-get -y install software-properties-common
4-   $ curl -fsSL https://download.docker.com/linux/ubuntu/gpg |
5-   $ sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

After completing the above-mentioned steps, the CLI for Docker is successfully installed. To verify the installation please check the given command below in your terminal example the latest docker version 20.10.11 is install the host VM for this project.

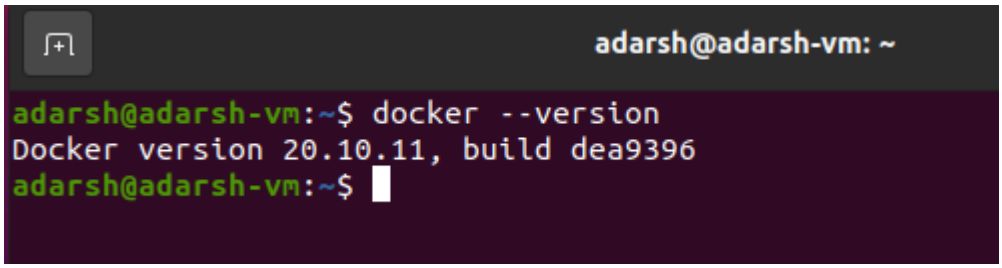


Figure 2 Installation of Docker

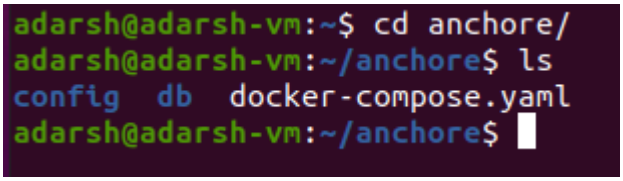
4 Setting up Automatic Image Analyzer

The AIA segment of the project is described in this section. All configuration records, as well as the stage process incorporation procedure, are discussed in the subcategories below. The whole module has being automated with Python; demonstrations of the code and results are provided in this part.

- ## To setup Anchore Engine,

There is a setup guide gicen on the website of Anchore Engine which is mentioned in the reference below.

1- mkdir anchore to create directory
2- In the Anchore Directory please make sub-directories of database and config
   cd anchore
   mkdir config
   mkdir db
3- Create a docker-compose.yaml



4- Refer the below mentioned for the recommended file.

```yaml
version: '2'
services:
  anchore-engine:
    image: docker.io/anchore/anchore-engine:v0.3.4
    #privileged: true
    depends_on:
      - anchore-db
    ports:
      - "8228:8228"
      - "8338:8338"
    volumes:
      - ./config:/config/:z
    logging:
      driver: "json-file"
      options:
        max-size: 100m
    environment:
      # NOTE: this should be set to the same name as this service (e.g. anchore-engine)
      - ANCHORE_HOST_ID=dockerhostid-anchore-engine
      - ANCHORE_ENDPOINT_HOSTNAME=anchore-engine
  anchore-db:
    image: "postgres:9"
    volumes:
      - ./db:/var/lib/postgresql/data/pgdata/:z
    environment:
      - POSTGRES_PASSWORD=mysecretpassword
      - PGDATA=/var/lib/postgresql/data/pgdata/
    logging:
      driver: "json-file"
      options:
        max-size: 100m
    #uncomment to expose a port to allow direct/external access to the DB, for debugging
    #ports:
    # - "2345:5432"
```

5- Now Create one more config.yaml file in the config folder.

6- Snipshot of config file is shown in the below Figure. The complete file is shared in zip folder.

```yaml
# otherwise anonymous access for feed sync is used

feeds:
  # If set to False, instruct anchore-engine to skip (all) feed sync operations
  sync_enabled: True
  ssl_verify: True
  selective_sync:
    # If enabled only sync specific feeds instead of all.
    enabled: True
    feeds:
      vulnerabilities: True
      # Warning: enabling the packages and nvd sync causes the service to require much
      #   more memory to do process the significant data volume. We recommend at least 4GB available for the container
      packages: False
      nvd: False
      # Enabling snyk syncs snyk vulnerability data from an on-premise anchore enterprise feeds service. Please contact
      # anchore support for finding out more about this service
      snyk: False
  anonymous_user_username: anon@ancho.re
  anonymous_user_password: pbiU2RYZ2XrnYQ
  url: 'https://ancho.re/v1/service/feeds'
  client_url: 'https://ancho.re/v1/account/users'
  token_url: 'https://ancho.re/oauth/token'
  connection_timeout_seconds: 3
  read_timeout_seconds: 60


# As of 0.3.0dev0 this section is used instead of the credentials.users section
# Can be omitted and will default to 'foobar' on db initialization
default_admin_password: 'foobar'

# Can be ommitted and will default to 'admin@myanchore'
default_admin_email: 'admin@myanchore'

credentials:
  users:
    admin:
      password: 'foobar'
      email: 'admin@myemail.com'
      external_service_auths:
      #  anchorelo:
      #    anchorecli:
      #      auth: 'myanchorelouser:myanchorelopass'
      #auto_policy_sync: True

  database:
    db_connect: 'postgresql+pg8000://postgres:mysecretpassword@anchore-db:5432/postgres'
```

7- Now the Anchore Engine configuration is completed.
8- Use (docker-compose up -d) to start the Anchore Engine
9- To Verify the state use (docker-compose ps).

```
adarsh@adarsh-vm:~/anchore$ ls
config  db  docker-compose.yaml
adarsh@adarsh-vm:~/anchore$ docker-compose up -d
Starting anchore_anchore-db_1 ...
Starting anchore_anchore-db_1 ... done
Starting anchore_anchore-engine_1 ...
Starting anchore_anchore-engine_1 ... done
adarsh@adarsh-vm:~/anchore$ docker-compose ps
         Name                    Command           State           Ports
-------------------------------------------------------------------------------
anchore_anchore-db_1     docker-entrypoint.sh     Up      5432/tcp
                         postgres
anchore_anchore-         /docker-entrypoint.sh    Up      0.0.0.0:8228->8228/tcp
engine_1                 anch ...                         ,:::8228->8228/tcp, 0.
                                                          0.0.0.0:8338->8338/tcp,:
                                                          ::8338->8338/tcp
adarsh@adarsh-vm:~/anchore$
```

10- Now you need to Install anchore-cli by apt install.

11- To start Anchore CLI need the below mentioned commands

export ANCHORE CLI URL
export ANCHORE CLI USER
export ANCHORE CLI PASS

12- When the Anchore Engine is fully operational, the feeds list (CVE database) on the local system must be updated. The screenshot is shown below:

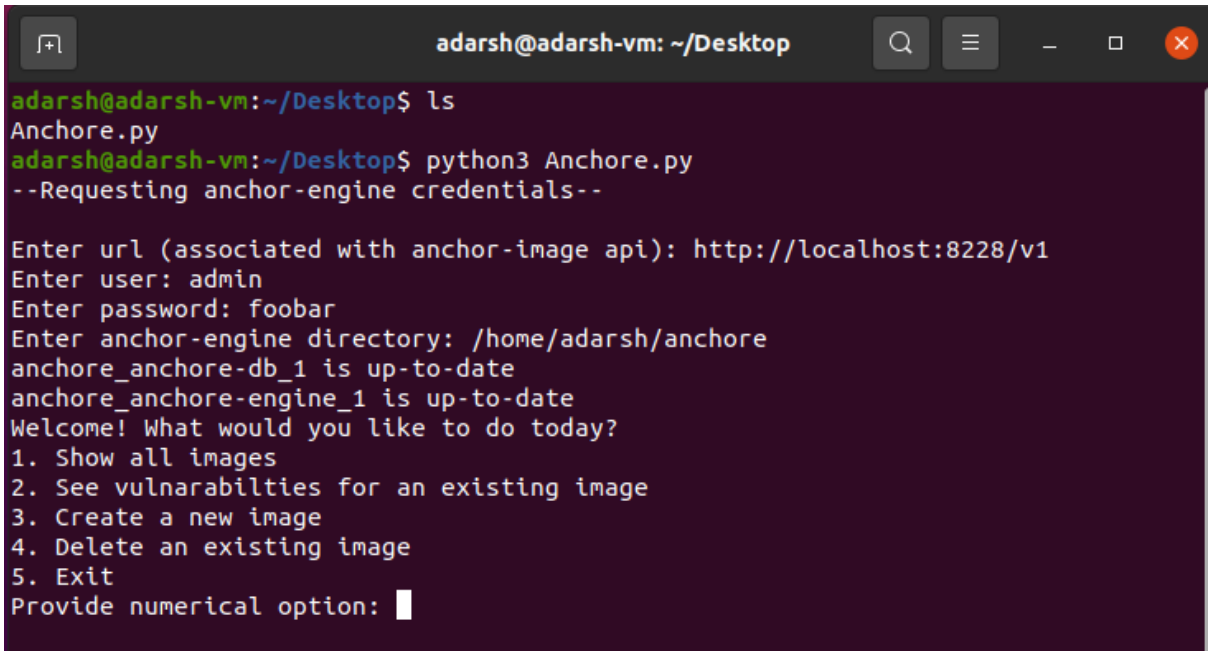
```
adarsh@adarsh-vm:~/anchore$ anchore-cli --url http://localhost:8228/v1 --u admin --p foobar system feeds list
Feed                Group                LastSync                        RecordCount
vulnerabilities     alpine:3.10          2021-12-16T12:37:08.090123      2329
vulnerabilities     alpine:3.11          2021-12-16T12:37:05.218541      2664
vulnerabilities     alpine:3.12          2021-12-16T12:37:05.837922      3025
vulnerabilities     alpine:3.13          2021-12-16T12:37:06.665675      3395
vulnerabilities     alpine:3.14          2021-12-16T12:36:29.989727      3792
vulnerabilities     alpine:3.15          2021-12-16T12:37:08.906162      3948
vulnerabilities     alpine:3.2           2021-12-16T12:36:30.593466      306
vulnerabilities     alpine:3.3           2021-12-16T12:36:31.634247      471
vulnerabilities     alpine:3.4           2021-12-16T12:36:32.068321      683
vulnerabilities     alpine:3.5           2021-12-16T12:36:32.887148      903
vulnerabilities     alpine:3.6           2021-12-16T12:36:33.461296      1077
vulnerabilities     alpine:3.7           2021-12-16T12:36:33.935423      1462
vulnerabilities     alpine:3.8           2021-12-16T12:36:34.901602      1675
vulnerabilities     alpine:3.9           2021-12-16T12:36:35.376898      1962
vulnerabilities     amzn:2               2021-12-16T12:36:36.250039      737
vulnerabilities     centos:5             2021-12-16T12:36:36.937150      1347
vulnerabilities     centos:6             2021-12-16T12:36:37.399590      1453
vulnerabilities     centos:7             2021-12-16T12:37:09.521243      1327
vulnerabilities     centos:8             2021-12-16T12:36:38.515099      782
vulnerabilities     debian:10            2021-12-16T12:36:39.585840      26640
vulnerabilities     debian:11            2021-12-16T12:36:40.605535      24098
vulnerabilities     debian:12            2021-12-16T12:36:41.355943      23157
vulnerabilities     debian:7             2021-12-16T12:36:41.792525      20455
vulnerabilities     debian:8             2021-12-16T12:36:42.286947      24058
vulnerabilities     debian:9             2021-12-16T12:36:43.369502      26516
vulnerabilities     debian:unstable      2021-12-16T12:36:44.755874      29046
vulnerabilities     ol:5                 2021-12-16T12:36:45.383647      1255
vulnerabilities     ol:6                 2021-12-16T12:36:45.966707      1644
vulnerabilities     ol:7                 2021-12-16T12:36:46.824218      1655
vulnerabilities     ol:8                 2021-12-16T12:36:47.404882      754
vulnerabilities     rhel:5               2021-12-16T12:37:10.144362      7764
vulnerabilities     rhel:6               2021-12-16T12:36:48.112199      7911
vulnerabilities     rhel:7               2021-12-16T12:36:48.726905      7272
vulnerabilities     rhel:8               2021-12-16T12:36:49.431171      3434
vulnerabilities     sles:11              2021-12-16T12:36:50.070717      594
vulnerabilities     sles:11.1            2021-12-16T12:36:50.680181      6008
vulnerabilities     sles:11.2            2021-12-16T12:36:51.289129      3291
vulnerabilities     sles:11.3            2021-12-16T12:36:51.912735      6897
vulnerabilities     sles:11.4            2021-12-16T12:36:52.326484      6437
vulnerabilities     sles:12              2021-12-16T12:36:52.934451      4244
vulnerabilities     sles:12.1            2021-12-16T12:36:53.548290      5203
```

13- Now you are ready to manually adding the Images and checking their vulnerability status.

For adding image: - anchore-cli image add <openjdk:8-jre-alpine> (openjdk:8-jre-alpine is an example of image). In the screenshot below the added images showing analysis status.

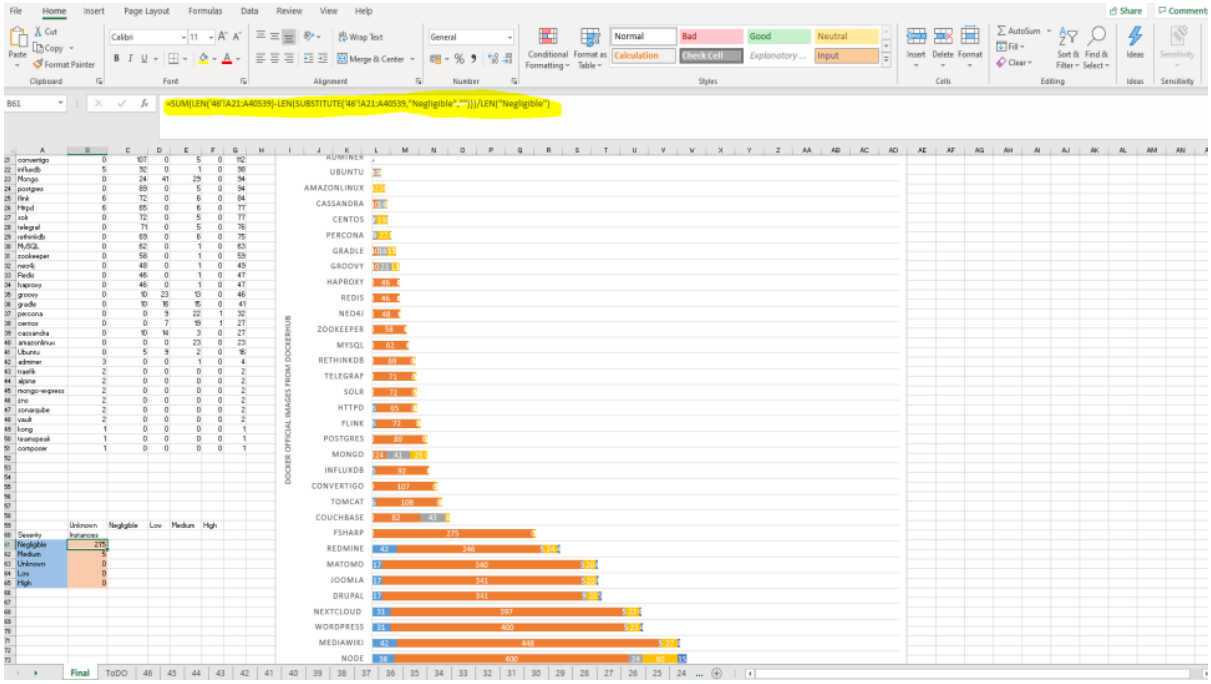
```
adarsh@adarsh-vm:~/anchore$ anchore-cli image list
Full Tag                                Image Digest
                                       Analysis Status
docker.io/couchbase:latest              sha256:f319e2e60eb1231197546238dfdfccaf7d
98540fb62d0b716c400ba507dce28a         analysis_failed
docker.io/library/debian:latest         sha256:2fa1cd247c9a06849534902b6b227efd21
dbb5c0fea11c75aeba1df18685f1b8         analyzed
docker.io/openjdk:10-jdk                sha256:923d074ef1f4f0dceef68d9bad8be19c91
8d9ca8180a26b037e00576f24c2cb4         analyzed
docker.io/openjdk:11-jdk                sha256:351cd469850de7ae1f820533ee18ef7ddb
cc0d1dcb6a3c8746e564e445476442         analyzed
docker.io/openjdk:8-jre-alpine          sha256:b2ad93b079b1495488cc01375de799c402
d45086015a120c105ea00e1be0fd52         analyzed
docker.io/vault:latest                  sha256:b7ef561ec7465fb621972b53a0066ccc04
7b3795d65288b128d5d171d69241c9         analyzed
adarsh@adarsh-vm:~/anchore$
```

14- Python language is used to automate the entire flow of Anchore Engine and using it user not need to go through with these much steps. The screenshot attached below is the User Interface.



- Management Module for Results

The AIA produces fairly long results. Figure below shows a sample result for one of the photos, which is around 12000 lines long. For analysis, the result management module incorporates macro scripts in Excel.



5 References

https://docs.docker.com/engine/install/ubuntu/

https://hub.docker.com/r/snyk/snyk-cli#:~:text=Snyk%20is%20a%20developer%2Dfirst,application%20code%20in%20real%20time