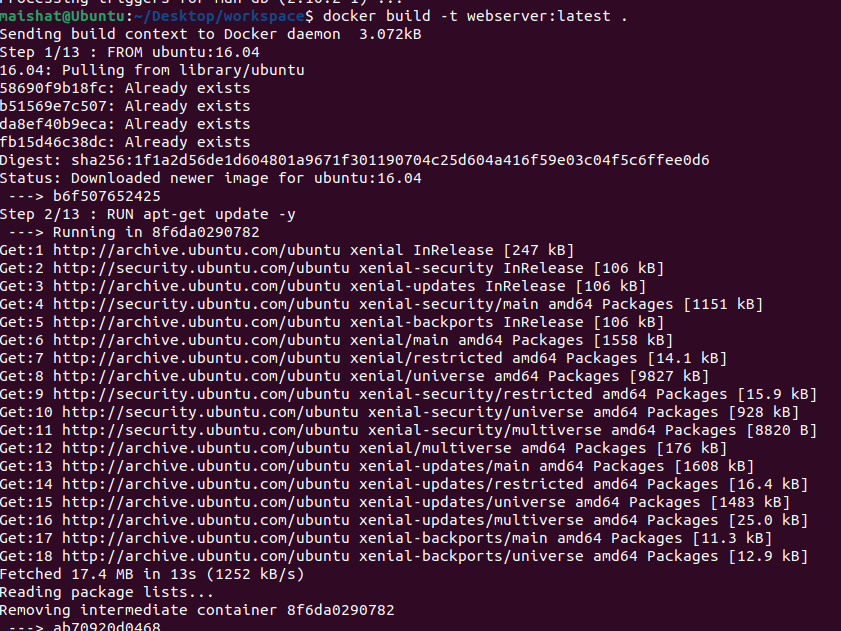
1. Here we are seeing the contents of the index.html and dockerfile. The index.html file will print the line in between header.

Dockerfile has base image of ubuntu16.04. We have all the commands in this file. We will add index.html to the container which is inside directory /var/www/html/. We will expose to the port 80. We will set the entry point which will start the apache server. The dockerfile will act as the configuration file to build the image.

A screenshot of a computer screen

Description automatically generated

1. We will build the image now.



A computer screen shot of white text

Description automatically generated

1. The list of all docker images. Ubuntu 16.04 is the base image and on top of it we will build webserver latest.

A screenshot of a computer screen

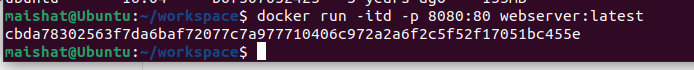
Description automatically generated

1. We have add new docker image which is alpine with latest tag.

A computer screen shot of a program

Description automatically generated

1. The 8080 is the port of the host and 80 is the port of the container. The port 8080 is mapped to the port 80 of the container. So we can access the website using the port which is mapped to the port of the host. From webserver image, the container will start. The output shows that the container has started.



1. The container has started and we can access the webpage we created.

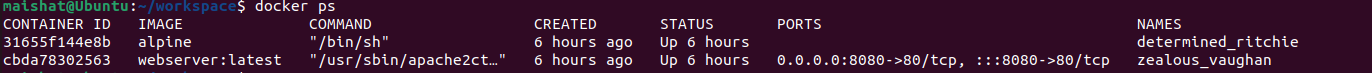
A screenshot of a computer

Description automatically generated

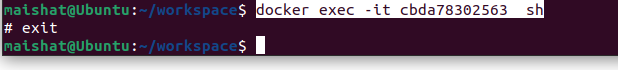
1. We can also start the container from alpine image.



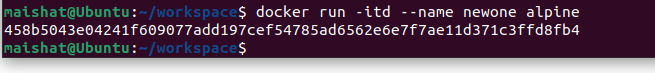
1. This command shows all the containers running in the system.



1. We can get an interactive shell in the container. Here we are running the command sh in the container id cbda78302563. Thus, getting the root shell.



1. We are starting a new container with name “newone” which will be interactive terminal and will run in backend.



1. Here, we are running the command sh in the container named “newone”.



1. We can run the command using 1st four digits of container id.

A computer screen with white text

Description automatically generated

1. We are stopping all docker images.

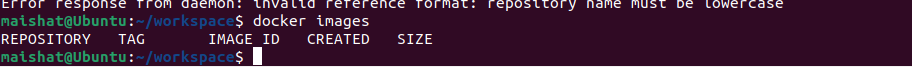
A computer screen with white text

Description automatically generated

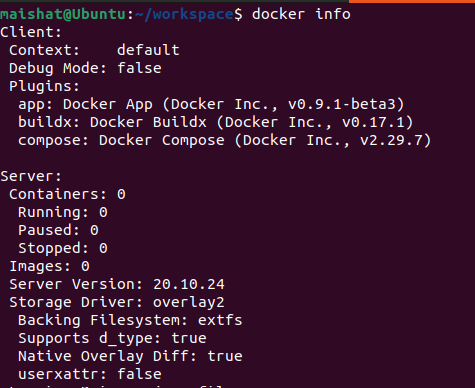
1. Now will remove all docker images.

A screenshot of a computer screen

Description automatically generated



1. This command gives all information of the docker.



1. Now if we go to the root directory and navigate through it, we get this output. And in the l directory, it does not contain anything for now.

A screenshot of a computer

Description automatically generated

1. We are again building the webserver image .

A computer screen shot of a program

Description automatically generated

A computer screen shot of a computer program

Description automatically generated

1. The root directory contains all the storage filesystem for Docker. Now if we again navigate through the l directory in overlay2 directory, we can see new layers are created when building the image.

A blue and white lines

Description automatically generated with medium confidence

1. It shows how cgroup limits the access of containers to system resources. If someone compromise the container, he can exhaust the resources of the host.

A computer screen with text

Description automatically generated

1. Here it shows how cgroup limit the pids to 6 from 12965 to prevent fork bombs.

A computer screen with white text

Description automatically generated

**Docker security and vulnerability:**

1. Trying to exploit vulnerable imageA computer screen shot of a computer error

   Description automatically generated
2. Getting into the reverse shell by listening to port 4444.

A screenshot of a computer

Description automatically generated

1. Privilege escalation by volume mounting. The elevated privilege of user to root level

A computer screen shot of a program

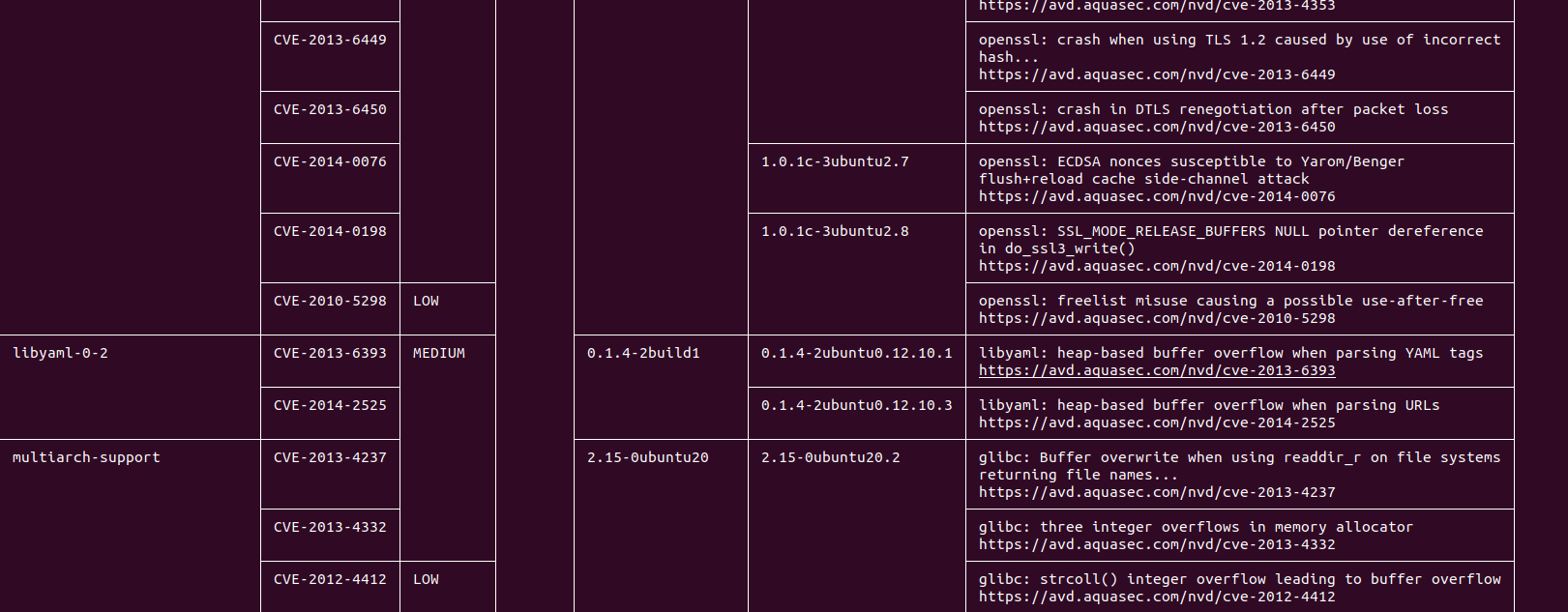
Description automatically generated

**Countermeasures for attack on docker image:**

1. Automated assessment -Trivy:

It does static analysis against vulnerable image.

docker run --rm -v $(pwd):/root/.cache/ aquasec/trivy image getcapsule8/shellshock:test



1. Starting the container vulnerable

A computer screen shot of a program

Description automatically generated

1. By running docker bench, the score is 18.

A screenshot of a computer program

Description automatically generated

1. If we made some modifications during starting the container.



1. After, we again run the docker bench, the score increases.

A screenshot of a computer program

Description automatically generated

1. If we start a new acontainer named vulnerable2 without any uid and gid, this gives us root shell access.A computer screen shot of a computer code

   Description automatically generated
2. Whereas vulnerable container with specified uid and gid, does not give us root shell access. Which prevents attacker from getting root access by exploiting vuknerability of images. And docker bench helps by warning this flags.

A computer screen shot of white text

Description automatically generated