

FINDING MY FRIEND WALKTHROUGH

Réaliser par :
Cheima TOUIR

Méthodologie utilisée

1. Reconnaissance

- Découverte de réseau :

```
FindingMyFriend [En fonction] - Oracle VM VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
Ubuntu 16.04.7 LTS findingmyfriend tty1
My IP address: 192.168.33.10

-----
| Created by Team :- VIEH GROUP |
|-----|
| Visit us :- www.viehgroup.com |
| Twitter :- @viehgroup |
|-----|
| Kshitiz Raj (@manitorpotterk) |
| Avinash Nagar (@_alpha_03) |
| Rohit Burke(@Buggrammers) |
|-----|
findingmyfriend login:
```

L'adresse IP de la cible est : 192.168.33.10

```
kali [En fonction] - Oracle VM VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
File Actions Edit View Help
(cheima㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.33.3 netmask 255.255.255.0 broadcast 192.168.33.255
        inet6 fe80::a00:27ff:fe8:76fa prefixlen 64 scopeid 0x20<link>
          ether 08:00:27:c8:76:fa txqueuelen 1000 (Ethernet)
            RX packets 72 bytes 8324 (8.1 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 200 bytes 30485 (29.7 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Local Loopback)
            RX packets 8 bytes 480 (480.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 8 bytes 480 (480.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(cheima㉿kali)-[~]
$ ping 192.168.33.10
PING 192.168.33.10 (192.168.33.10) 56(84) bytes of data.
64 bytes from 192.168.33.10: icmp_seq=1 ttl=64 time=1.08 ms
64 bytes from 192.168.33.10: icmp_seq=2 ttl=64 time=1.16 ms
64 bytes from 192.168.33.10: icmp_seq=3 ttl=64 time=0.811 ms
64 bytes from 192.168.33.10: icmp_seq=4 ttl=64 time=0.606 ms
^C
--- 192.168.33.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3020ms
rtt min/avg/max/mdev = 0.606/0.914/1.159/0.219 ms

(cheima㉿kali)-[~]
```

- La première commande exécutée est **ifconfig**, qui affiche les interfaces réseau de la machine.
- La commande **ping 192.168.33.10** est utilisée pour tester la connectivité réseau vers l'adresse IP 192.168.33.10.

Le test ping montre une bonne connectivité entre l'hôte et l'adresse 192.168.33.10, sans perte de paquets et avec des temps de réponse rapides. Cela indique un réseau local fonctionnel, sans problème de latence ou de connectivité significatif.

• Scanning des ports (nmap)

```
(cheima㉿kali)-[~]
$ sudo nmap -sS -sV -O -A 192.168.33.10
[sudo] password for cheima:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-29 09:57 CET
Stats: 0:00:02 elapsed; 0 hosts completed (0 up), 1 undergoing ARP Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Stats: 0:00:06 elapsed; 0 hosts completed (0 up), 1 undergoing ARP Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Stats: 0:00:10 elapsed; 0 hosts completed (0 up), 1 undergoing ARP Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Stats: 0:00:19 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 66.67% done; ETC: 09:57 (0:00:03 remaining)
Nmap scan report for 192.168.33.10
Host is up (0.0010s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 3.0.3
22/tcp    open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   2048 ce:19:b7:da:b3:c5:10:73:a7:43:3c:7e:93:50:74:3d (RSA)
|   256 35:25:f6:bb:df:1d:b6:fd:cd:0b:df:4b:30:14:3d:3b (ECDSA)
|_ 256 ac:c6:71:53:6b:b5:4a:0a:3a:85:ae:67:32:5d:e2:04 (ED25519)
80/tcp    open  http   Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
MAC Address: 08:00:27:82:AC:A7 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1  1.04 ms  192.168.33.10

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 22.20 seconds
```

Commande exécutée :

sudo nmap -sS -sV -O -A 192.168.33.10

Cette commande effectue un scan Nmap avec les options suivantes :

- **-sS** : Scan SYN (scan furtif) pour détecter les ports ouverts.
- **-sV** : Détection de la version des services sur les ports ouverts.
- **-O** : Détection du système d'exploitation.
- **-A** : Activation de l'analyse avancée, incluant la détection du système d'exploitation, la version des services et le traceroute.

Résultats du scan :

- Adresse cible : 192.168.33.10
- Ports ouverts et services détectés :
 - 21/tcp (FTP) : Le port 21 est ouvert et utilise le service ftp avec le logiciel vsftpd 3.0.3.
 - 22/tcp (SSH) : Le port 22 est ouvert avec OpenSSH 7.2p2, fonctionnant sous Ubuntu 4ubuntu2.10.
 - 80/tcp (HTTP) : Le port 80 est ouvert avec un serveur web Apache 2.4.18 fonctionnant également sur Ubuntu.
- Le titre de la page web est : "Site doesn't have a title (text/html)", indiquant qu'aucun titre spécifique n'est défini.
- Informations sur les clés SSH : Des informations sur les clés SSH pour les méthodes RSA, ECDSA et ED25519 sont affichées, avec des empreintes spécifiques pour chaque type.
- Adresse MAC : 08:00:27:AC:AA:7A, indiquant que l'hôte est probablement une machine virtuelle VirtualBox.

- Détails du système d'exploitation : Le scan identifie l'OS comme étant basé sur Linux, version du noyau 3.x à 4.x, avec un CPE (Common Platform Enumeration) pour Linux Kernel.

Traceroute :

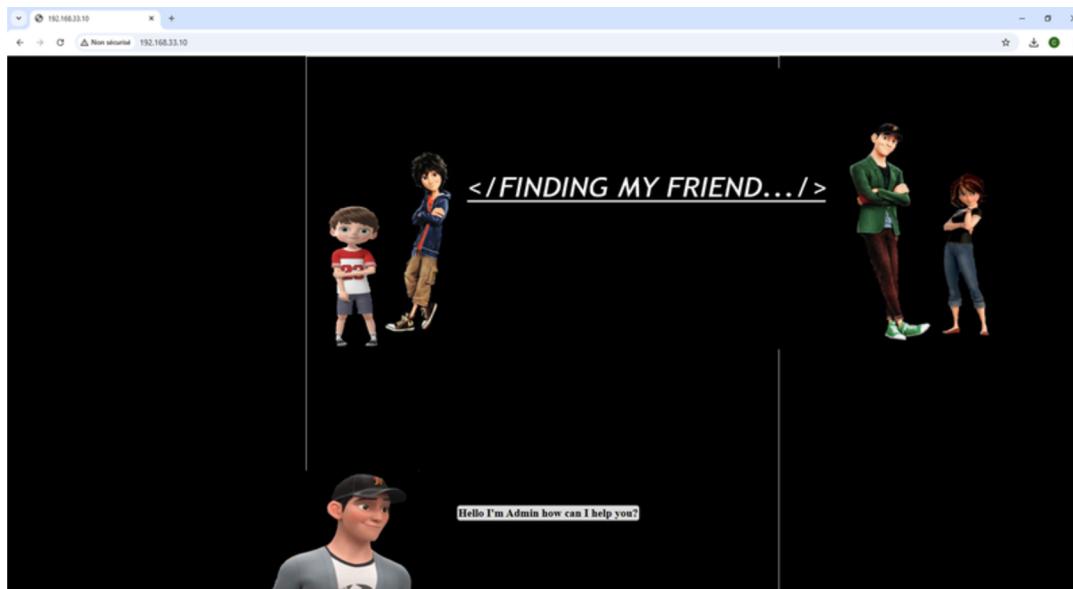
- Le traceroute indique une distance réseau de 1 saut avec une RTT (Round Trip Time) de 1.04 ms pour atteindre l'adresse cible 192.168.33.10, ce qui confirme que l'hôte est sur le même réseau local.

Résultats du scan :

- Le scan a détecté un total de 3 ports ouverts (21, 22, 80), tous associés à des services bien connus (FTP, SSH, HTTP).
- Le système d'exploitation est détecté comme étant basé sur Linux, probablement sur une machine virtuelle VirtualBox sous Ubuntu.

2. Scanning

- Sur le port 80 de service HTTP**



Visiter la page web de l'adresse IP de la cible <http://192.168.33.10>

On a utilisé l'outil Dirb sous Kali Linux pour effectuer un scan de découverte de répertoires sur le site web hébergé à l'adresse <http://192.168.33.10>

Commande exécutée :

```
dirb http://192.168.33.10
```

Cette commande lance Dirb, un outil de bruteforce qui cherche des répertoires et fichiers sur un serveur web en utilisant une liste de mots ([wordlist](#)). Ici, Dirb utilise la liste de mots par défaut située dans [/usr/share/dirb/wordlists/common.txt](#).



```
(cheima㉿kali)-[~]
$ dirb http://192.168.33.10

DIRB v2.22
By The Dark Raver

START_TIME: Tue Oct 29 10:04:18 2024
URL_BASE: http://192.168.33.10/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

— Scanning URL: http://192.168.33.10/
⇒ DIRECTORY: http://192.168.33.10/friend/
⇒ DIRECTORY: http://192.168.33.10/images/
+ http://192.168.33.10/index.html (CODE:200|SIZE:2275)
+ http://192.168.33.10/server-status (CODE:403|SIZE:278)

— Entering directory: http://192.168.33.10/friend/
+ http://192.168.33.10/friend/index.html (CODE:200|SIZE:2391)

— Entering directory: http://192.168.33.10/images/
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

END_TIME: Tue Oct 29 10:04:25 2024
DOWNLOADED: 9224 - FOUND: 3

(cheima㉿kali)-[~]
$
```

Résultats générés :

DIRECTORY :

- **http://192.168.33.10/friend/ :**

Dirb a trouvé ce répertoire sur le serveur. Lorsqu'il entre dans ce répertoire, il trouve un fichier index.html (Code HTTP 200), de taille 2391 octets.

- **http://192.168.33.10/images/ :**

Ce répertoire est également trouvé, et Dirb indique que le répertoire est listable, ce qui signifie que l'indexation des fichiers est activée et que les fichiers contenus dans ce dossier peuvent être consultés directement sans besoin de bruteforce.

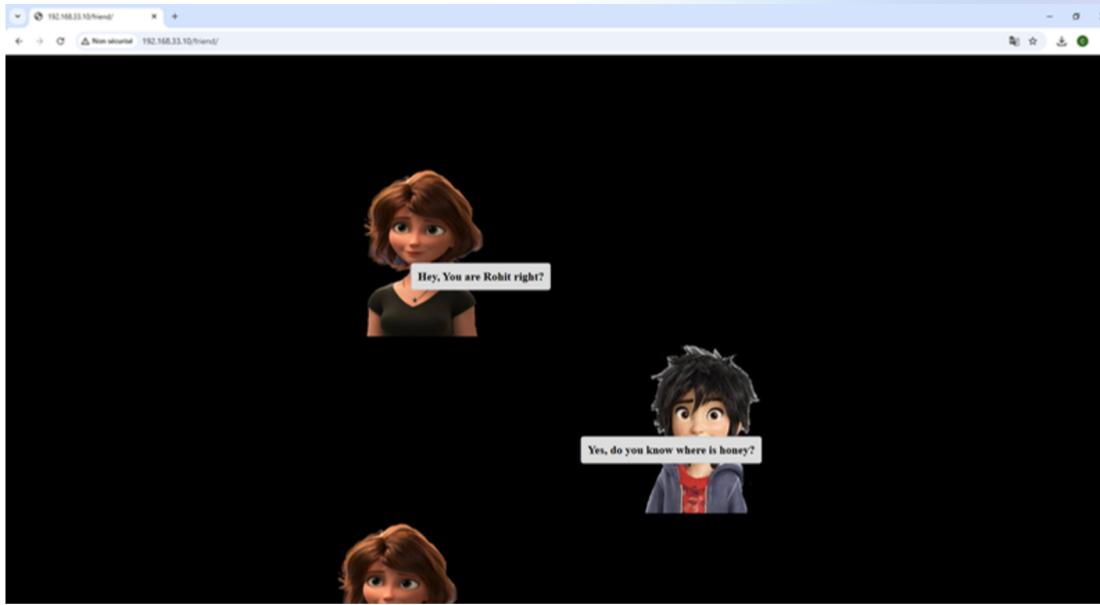
Fichiers et autres réponses HTTP :

- **http://192.168.33.10/index.html :**

Page d'index avec le code HTTP 200 (OK), indiquant qu'elle est accessible, de taille 2275 octets.

- **http://192.168.33.10/server-status :**

Cette ressource retourne un code HTTP 403 (Forbidden), ce qui signifie que l'accès est interdit.



```

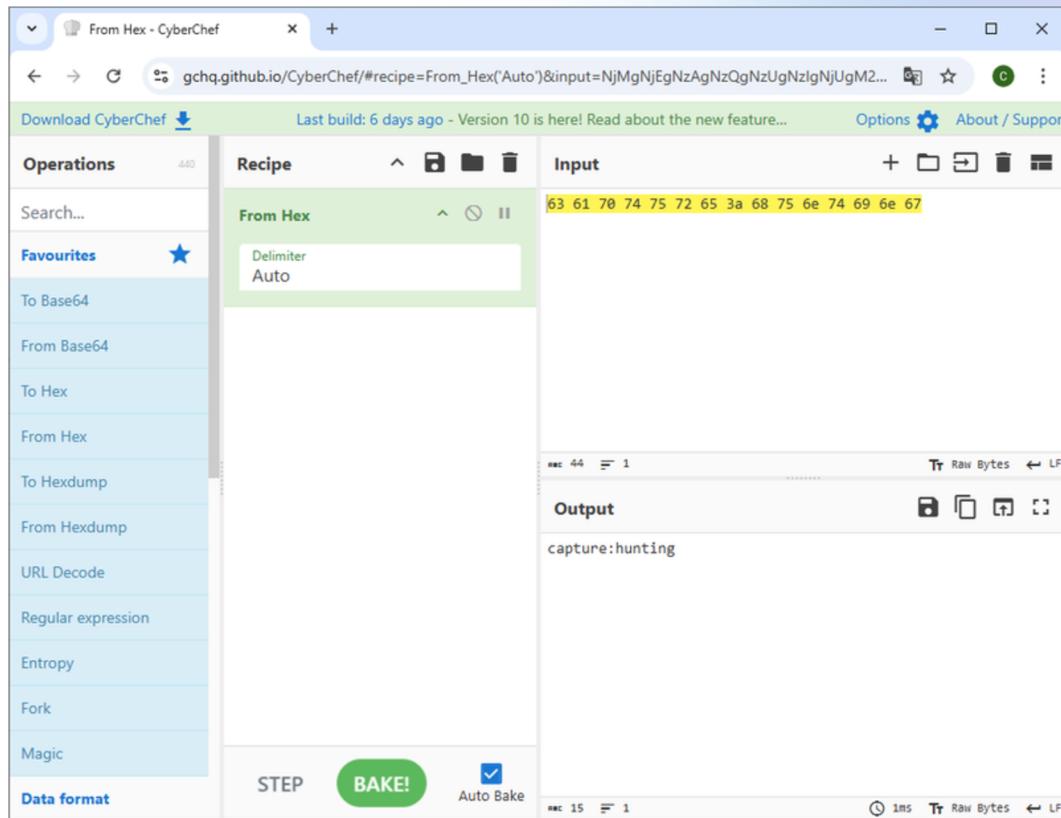
87 
88 <div class="container darker" style = "position:absolute; font-size:20px; right:250px; top:160px;">
89   <b>I can't read that. But it was like this</b> <!-- NjMgNjEgNzAgNzQgNzUgNzIgNjUgM2EgNjggNzUgNmUgNzQgNjkgNmUgNjc= -->
90   </div>
91
92 </body>
93 </html>
94
95

```

- Ce bloc de texte, avec la classe CSS "container darker" contient le texte "I can't read that. But it was like this", ce qui suggère qu'un contenu difficile à lire ou encodé est présent
- Un commentaire HTML contenant une chaîne de caractères encodée en Base64. Cette chaîne pourrait contenir des informations cachées ou un indice.

The screenshot shows the CyberChef interface with the following details:

- Operations:** A sidebar on the left listing various encoding/decoding tools: To Base64, From Base64, To Hex, From Hex, To Hexdump, From Hexdump, URL Decode, Regular expression, Entropy, Fork, Magic.
- Recipe:** The main panel shows a "From Base64" recipe with the following settings:
 - Alphabet: A-Za-z0-9%2B%3D
 - Remove non-alphabet chars: checked
 - Strict mode: unchecked
- Input:** The input text is: NjMgNjEgNzAgNzQgNzUgNzIgNjUgM2EgNjggNzUgNmUgNzQgNjkgNmUgNjc=
- Output:** The output bytes are: 63 61 70 74 75 72 65 3a 68 75 6e 74 69 6e 67



Étapes détaillées :

- Première capture d'écran : Décodage de Base64
 - La chaîne encodée en Base64 :
NjM5NzA2NGQzNzU2NjUgMzE0Njg0NmU2YjU4NjkNmU2Njc= est entrée dans CyberChef avec l'opération From Base64.
 - Le résultat est une séquence en Hexadécimal :
63 61 70 74 75 72 65 3a 68 75 6e 74 69 6e 67.
- Deuxième capture d'écran : Décodage de Hexadécimal
 - La séquence Hexadécimale :
63 61 70 74 75 72 65 3a 68 75 6e 74 69 6e 67 est ensuite décodée avec l'opération from Hex.
 - Le texte décodé est :
capture:hunting



- **Exploitation port 21 de service FTP**

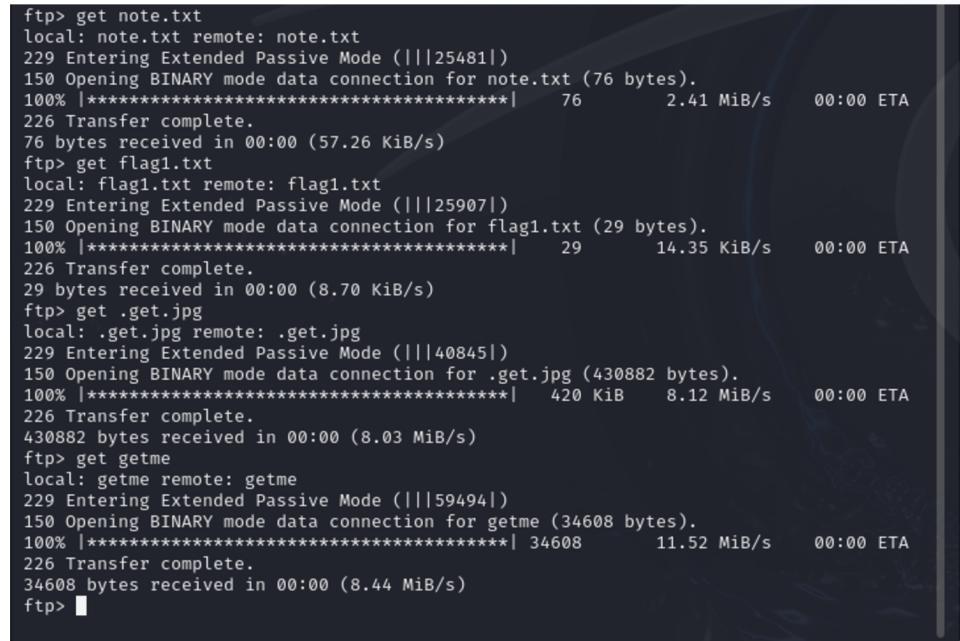
```
(cheima㉿kali)-[~]
└─$ ftp 192.168.33.10
Connected to 192.168.33.10.
220 (vsFTPD 3.0.3)
Name (192.168.33.10:cheima): capture
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

- La commande `ls -l` est utilisée pour lister les fichiers dans le répertoire courant.
- La commande `ls -al` montre la liste des fichiers, incluant les fichiers cachés et les permissions complètes

```
(cheima㉿kali)-[~]
└─$ ftp 192.168.33.10
Connected to 192.168.33.10.
220 (vsFTPD 3.0.3)
Name (192.168.33.10:cheima): capture
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -l
229 Entering Extended Passive Mode (|||53083|)
150 Here comes the directory listing.
-rwxr-x-- 1 1002 1002 29 Jan 06 2021 flag1.txt
-rwxr-x-- 1 1002 1002 34608 Jan 06 2021 getme
-rwxr-x-- 1 1002 1002 76 Jan 06 2021 note.txt
226 Directory send OK.
ftp> ls -al
229 Entering Extended Passive Mode (|||16030|)
150 Here comes the directory listing.
drwxr-x-- 2 1002 1002 4096 Jan 06 2021 .
drwxr-x-- 2 1002 1002 4096 Jan 06 2021 ..
-rwxr-x-- 1 1002 1002 430882 Jan 06 2021 .get.jpg
-rwxr-x-- 1 1002 1002 29 Jan 06 2021 flag1.txt
-rwxr-x-- 1 1002 1002 34608 Jan 06 2021 getme
-rwxr-x-- 1 1002 1002 76 Jan 06 2021 note.txt
226 Directory send OK.
ftp> 
```

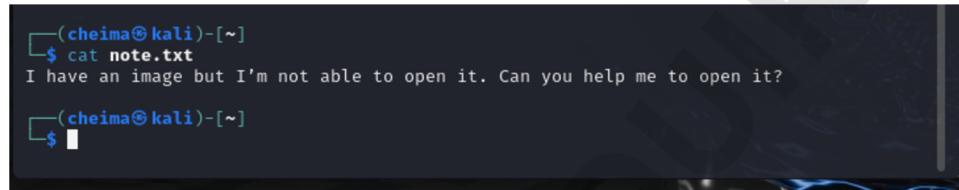
Téléchargement de fichiers :

La commande get note.txt est utilisée pour télécharger le fichier note.txt.



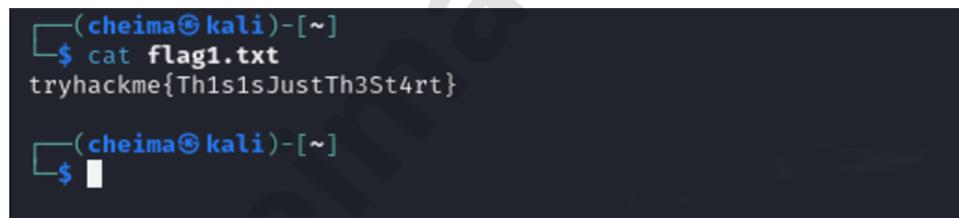
```
ftp> get note.txt
local: note.txt remote: note.txt
229 Entering Extended Passive Mode (|||25481|)
150 Opening BINARY mode data connection for note.txt (76 bytes).
100% |*****| 76      2.41 MiB/s  00:00 ETA
226 Transfer complete.
76 bytes received in 00:00 (57.26 KiB/s)
ftp> get flag1.txt
local: flag1.txt remote: flag1.txt
229 Entering Extended Passive Mode (|||25907|)
150 Opening BINARY mode data connection for flag1.txt (29 bytes).
100% |*****| 29      14.35 KiB/s  00:00 ETA
226 Transfer complete.
29 bytes received in 00:00 (8.70 KiB/s)
ftp> get .get.jpg
local: .get.jpg remote: .get.jpg
229 Entering Extended Passive Mode (|||40845|)
150 Opening BINARY mode data connection for .get.jpg (430882 bytes).
100% |*****| 420 KiB   8.12 MiB/s  00:00 ETA
226 Transfer complete.
430882 bytes received in 00:00 (8.03 MiB/s)
ftp> get getme
local: getme remote: getme
229 Entering Extended Passive Mode (|||59494|)
150 Opening BINARY mode data connection for getme (34608 bytes).
100% |*****| 34608    11.52 MiB/s  00:00 ETA
226 Transfer complete.
34608 bytes received in 00:00 (8.44 MiB/s)
ftp> 
```

La commande get note.txt , get flag1.txt ,get .get.jpg et get getme sont utilisées pour télécharger les différents fichiers trouvés.



```
(cheima㉿kali)-[~]
$ cat note.txt
I have an image but I'm not able to open it. Can you help me to open it?
(cheima㉿kali)-[~]
```

Dans cette capture d'écran, le contenu du fichier note.txt est affiché en utilisant la commande cat note.txt



```
(cheima㉿kali)-[~]
$ cat flag1.txt
tryhackme{Th1s1sJustTh3St4rt}

(cheima㉿kali)-[~]
```

Dans cette capture d'écran, le contenu du fichier flag1.txt est affiché en utilisant la commande cat flag1.txt.

Le fichier contient le texte suivant : tryhackme{Th1s1sJustTh3St4rt}

- Savoir les types des fichiers getme et .get.jpg

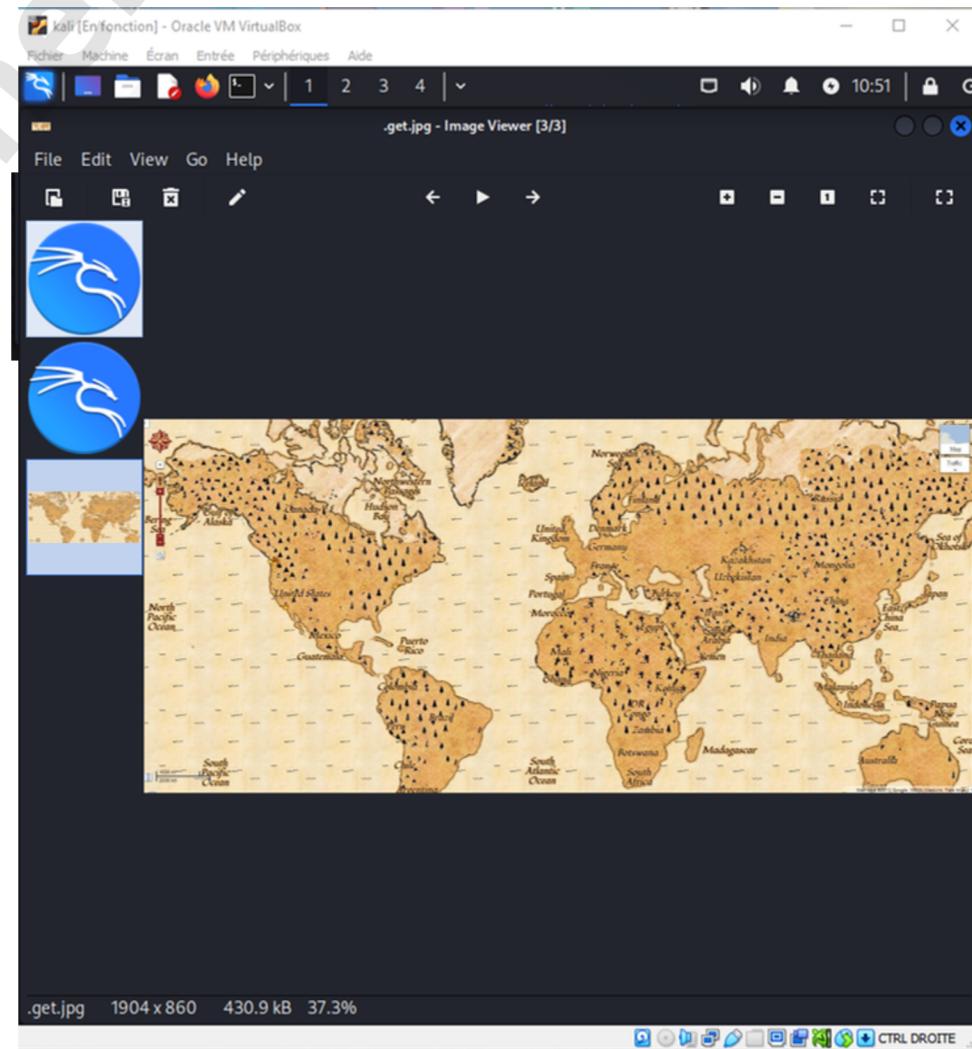
```
(cheima㉿kali)-[~]
└─$ file getme
getme: data

(cheima㉿kali)-[~]
└─$
```

```
(cheima㉿kali)-[~]
└─$ file .get.jpg
.get.jpg: JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16, baseline, precision 8, 1904x860, components 3

(cheima㉿kali)-[~]
└─$
```

- Voir le contenu d'image .get.jpg



- Récupérer des données cachées dans l'image

```
└── [cheima㉿kali)-[~]
$ stegcracker -get.jpg /usr/share/wordlists/rockyou.txt.gz
StegCracker 2.1.0 - (https://github.com/Paradoxis/StegCracker)
Copyright (c) 2024 - Luke Paris (Paradoxis)

StegCracker has been retired following the release of StegSeek, which
will blast through the rockyou.txt wordlist within 1.9 second as opposed
to StegCracker which takes ~5 hours.

StegSeek can be found at: https://github.com/RickdeJager/stegseek

Error: It appears you're using a gzipped variant of a wordlist, instead of the actual wordlist itself. You can decompress the gzipped using the following command: gzip -d /usr/share/wordlists/rockyou.txt.gz

└── [cheima㉿kali)-[~]
$ gunzip /usr/share/wordlists/rockyou.txt.gz
gzip: /usr/share/wordlists/rockyou.txt: Permission denied

└── [cheima㉿kali)-[~]
$ sudo gunzip /usr/share/wordlists/rockyou.txt.gz

└── [cheima㉿kali)-[~]
$ stegcracker -get.jpg /usr/share/wordlists/rockyou.txt
StegCracker 2.1.0 - (https://github.com/Paradoxis/StegCracker)
Copyright (c) 2024 - Luke Paris (Paradoxis)

StegCracker has been retired following the release of StegSeek, which
will blast through the rockyou.txt wordlist within 1.9 second as opposed
to StegCracker which takes ~5 hours.

StegSeek can be found at: https://github.com/RickdeJager/stegseek

Counting lines in wordlist..
Attacking file '.get.jpg' with wordlist '/usr/share/wordlists/rockyou.txt'..
Successfully cracked file with password: pollito
Tried 1167 passwords
Your file has been written to: .get.jpg.out
pollito
```

Dans cette capture d'écran, l'utilisateur utilise StegCracker pour tenter de récupérer des données cachées dans le fichier .get.jpg en utilisant une attaque par dictionnaire avec la wordlist rockyou.txt.

Explication des étapes :

- Décompression de rockyou.txt.gz
 - sudo gunzip /usr/share/wordlists/rockyou.txt.gz
 - Exécution de StegCracker avec rockyou.txt
 - stegcracker .get.jpg /usr/share/wordlists/rockyou.txt
 - Résultats
 - StegCracker réussit à "cracker" le fichier et identifie le mot de passe "polulito".

Le mot de passe pour accéder aux données cachées dans .get.jpg est "polulto". Ce mot de passe a permis à StegCracker d'extraire les informations cachées, qui sont maintenant disponibles dans le fichier **.get.jpg.out**. L'utilisateur peut maintenant ouvrir ce fichier pour examiner le contenu caché.

• Extraction de données avec steghide

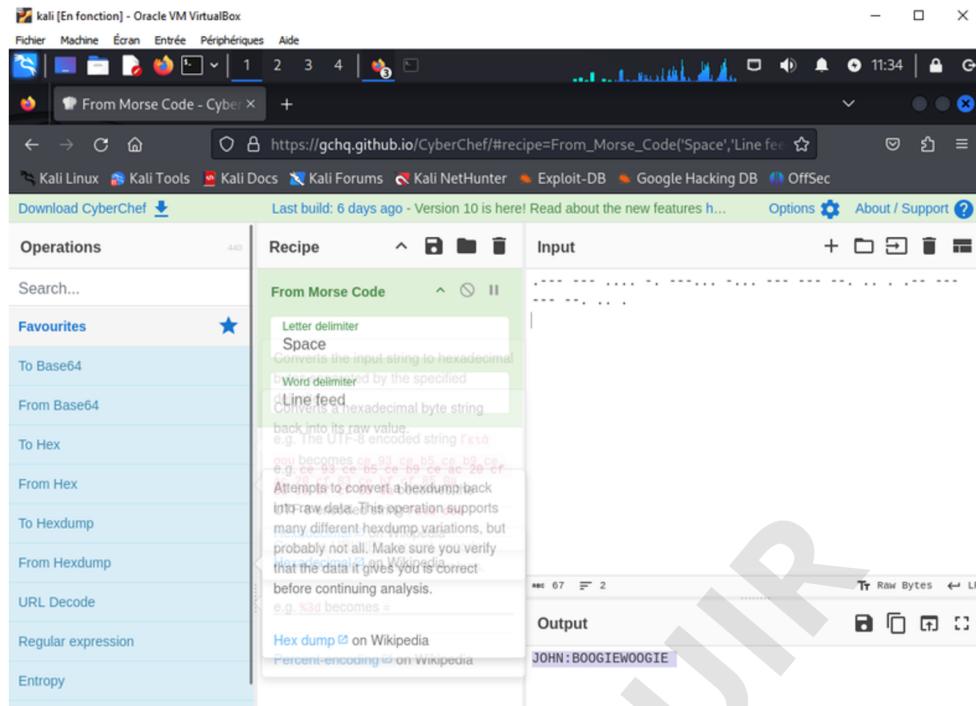
`sudo steghide extract -sf .get.jpg`

Steghide demande une passphrase, qui est probablement "polulito" (le mot de passe trouvé précédemment avec StegCracker).

Les données extraites sont enregistrées dans un fichier nommé abcd.txt.

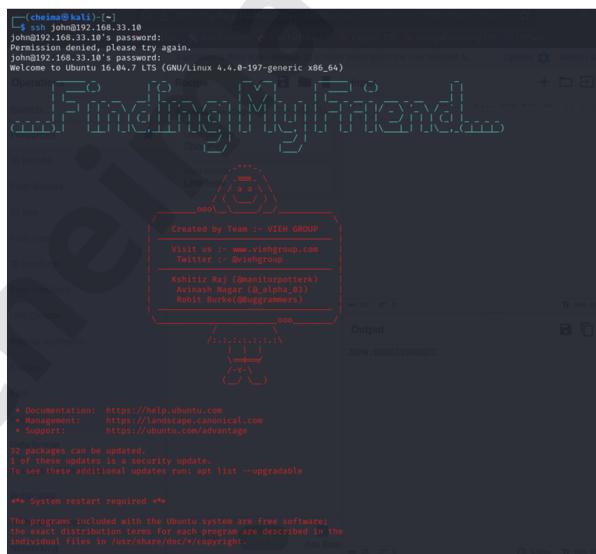
En ouvrant abcd.txt avec la commande cat, on voit une série de points et de tirets, ce qui ressemble à du code Morse.

- **Décodage du code Morse avec CyberChef**



Cela ressemble à un identifiant et un mot de passe (probablement pour un accès SSH).

- **Connexion SSH avec les informations d'identification**



Utiliser ces identifiants pour se connecter avec succès au serveur cible via SSH.

Accéder au serveur en tant qu'utilisateur john avec le mot de passe BOOGIEWOOGIE.